# STaRFlow: A SpatioTemporal Recurrent Cell for Lightweight Multi-Frame Optical Flow Estimation

Pierre Godet, Alexandre Boulch, Aurélien Plyer, Guy Le Besnerais

# STaRFlow: A SpatioTemporal Recurrent Cell for Lightweight Multi-Frame Optical Flow Estimation

Pierre Godet*, Alexandre Boulch†, Aurélien Plyer* and Guy Le Besnerais*
*DTIS, ONERA, Université Paris-Saclay, FR-91123 Palaiseau, France
Email: {pierre.godet, aurelien.plyer, guy.le_besnerais}@onera.fr
†valeo.ai, Paris, France
Email: alexandre.boulch@valeo.com

***Abstract*—We present a new lightweight CNN-based algorithm for multi-frame optical flow estimation. Our solution introduces a double recurrence over spatial scale and time through repeated use of a generic "STaR" (SpatioTemporal Recurrent) cell. It includes (i) a temporal recurrence based on conveying learned features rather than optical flow estimates; (ii) an occlusion detection process which is coupled with optical flow estimation and therefore uses a very limited number of extra parameters. The resulting STaRFlow algorithm gives state-of-the-art performances on MPI Sintel and Kitti2015 and involves significantly less parameters than all other methods with comparable results.**

## I. Introduction

Optical Flow (OF) is the apparent displacement of objects between two frames of a video sequence. It expresses the direction and the magnitude of the motion of each object at pixel level. The OF is a key component for several computer vision tasks, such as action recognition [1], autonomous navigation [2], tracking [3], or image registration for multi-view applications like video inpainting [4], super-resolution [5], [6], [7] or structure from motion [8]. OF estimation must be fast, accurate even at subpixel level for some applications like super-resolution, and reliable even at sharp motion boundaries despite occlusion effects. Particularly, it must deal with challenging contexts such as fast motions, motion blur, illumination effects, uniformly colored objects, etc.

Starting from the seminal work of Horn and Schunck [9], OF estimation has been the subject of numerous works. Recently, a breakthrough came with deep neural networks. Convolutional neural network-based (CNNs) methods [10], [11], [12], [13] reached the state of the art on mostly all large OF estimation benchmarks, e.g., MPI Sintel [14] and Kitti [15], while running much faster than previous variational methods.

In order to increase the efficiency and the robustness of these methods, the focus has then been put on occlusion detection [16], [17], [18], temporal dependency [17] or memory efficiency [19], [18]. Building on these concerns, our work follows two main orientations. First, when processing a video sequence, most object motions are continuous across frame pairs. Thus, most of the uncertainties arising from two-frame OF estimation can be solved using a number of frames greater than two. This calls for a multi-frame estimation process able to exploit temporal redundancy of the OF. Second, we believe that related operations can be performed by identical models with shared weights. We apply this principle to temporal recurrence, as in [17], to scale recurrence, as in [18], but also to occlusion detection, which is strongly correlated with OF estimation. Based on these considerations, we propose a "doubly recurrent" network over spatial scales and time instants. It takes explicitly into account the information from previous frames and the redundancy of the estimation at each network scale within a unique processing cell, denoted *STaR* cell, for *S*patio*T*empor*a*l *R*ecurrent cell. Given information from the past and from a lower scale, the STaR cell outputs the OF and occlusion map at current image scale and time instant. This cell is repeatedly invoked over scales in a coarse-to-fine scheme and over sets of $N$ successive frames, leading to the STaRFlow model. Thanks to this doubly recurrent structure, and by sharing the weights between processes dedicated to flow estimation and to occlusion detection, we obtain a lightweight model: STaRFlow is indeed slightly lighter than LiteFlowNet [19], while producing jointly multi-frame OF estimation and occlusion detection.

Let us now outline the organization of the paper while listing our main contributions. We first discuss related work in Section II, then Section III is devoted to the description of our main contribution, the STaRFlow model for multi-frame OF estimation. Experiments are presented in Section IV, with results on MPI Sintel [14] and Kitti [15]: examples of results of STaRFlow on these two datasets are presented in Figure 1. We conduct in particular an ablation study that addresses three important subjects: temporal recurrence, occlusions and scale recurrence. First, as regards temporal recurrence, we show that passing learned features between instants compares favourably to passing previously estimated OF as in ContinualFlow [17]. Our approach also makes a higher benefit from larger number of frames than [17]. Secondly, our occlusion handling appears as efficient as previously published approaches, but is much simpler and involves a significantly lower number of extra parameters. Thirdly, the study of scale recurrence highlights the compactness of our model. Finally, concluding remarks and perspectives are given in Section V. Our implementation of STaRFlow, with training code and trained model parameters, is available on https://github.com/pgodet/star_flow.

Fig. 1. Qualitative results of the proposed STaRFlow model, on MPI Sintel final pass (top line) and KITTI 2015 (bottom line) test sets. STaRFlow allows accurate motion estimation on partially occluded objects (right knee of character in upper leftmost example) and on thin objects (fingers and posts in the rightmost examples).

## II. Related Work

### A. Optical Flow (OF) Estimation With CNN

Dosovitskiy *et al.* [10] were the first to publish a deep learning approach for OF estimation. They proposed a synthetic training dataset, FlyingChairs, and two CNN architectures FlowNetS and FlowNetC. They have shown fairly good results, though not state-of-the-art, on benchmarks data which are very different from their simple 2D synthetic training dataset. By using a more complex training dataset, FlyingThings3D [11], and a bigger architecture involving several FlowNet blocks, Ilg *et al.* [12] proposed the first state-of-the-art CNN-based method for OF estimation. Moreover, their learning strategy (FlyingChairs $\rightarrow$ FlyingThings) was then used by several supervised learning approaches.

Some of the works that followed [20], [19], [13] sought to leverage well-known classical practices in OF estimation, like warping-based multi-scale estimation, within a deep learning framework, leading to state-of-the-art algorithms [19], [13]. In particular, PWC-Net of [13] has then been used as a baseline for several top-performing methods [17], [21], [22], [18], [23]. Very recently, Hur and Roth [18] got even closer to classical iterative OF estimation processes with an "iterative residual refinement" (IRR) version of PWC-Net. IRR mainly consists in using the same learned parameters for every stage of the decoder, so as to obtain a lighter and better-performing method. We exploit the same idea but extend it to scale *and temporal* iterations in a multi-frame setting.

### B. Multi-Frame Optical Flow Estimation

Exploiting temporal coherence as been proven to improve estimation quality. Wang *et al.* [24] use multiple frames in a Lucas-Kanade [25] estimation process and show better results when increasing the number of frames, *i.e.* a less noisy estimation and a reduced number of ambiguous matching points. Volz *et al.* [26] also improve their estimate, in particular in untextured regions, by modeling temporal coherence with an adaptive trajectory regularization in a variational method. Kennedy and Taylor [27] shown improved results on the MPI Sintel benchmark [14] by using additional frames, more significantly in unmatched regions.

Additional frames are useful to cope with occlusions, as, for instance, pixels visible at time $t$ and occluded at time $t+1$ may have been visible at time $t-1$. Hence, the OF is ill-defined from $t$ to $t+1$ but can be filled in with the estimation at the previous time step. Ren *et al.* [21] propose a multi-frame fusion process to fuse the current OF estimate with the estimate at the previous time step. Maurer and Bruhn [28] propose to learn, with a CNN, how to infer the forward flow from the backward flow, and fuse it with the actual estimated forward flow. Note that in these references, the multi-frame estimation stems from the fusion of two OF estimates provided by classical two-frame processes launched between different frame pairs. In contrast, in the ContinualFlow model of [17], a temporal connection is introduced to pass the OF estimate at time $t-1$ to the estimation process at time $t$, making the estimation recurrent in time. Let us also mention that, in an unsupervised learning framework, [29], [22] and [30] also show improved results, more significantly in occluded areas, by using multiple frames. These methods use 3 frames and estimate jointly the OF from $t$ to $t-1$ and from $t$ to $t+1$.

Our work is closer to [17], as we propose to use a recurrent temporal connection, but is based on passing learned features from one instant to the next rather than OF estimates. According to our experiments, this approach is more efficient and allows to exploit a larger time range than ContinualFlow [17].

### C. Occlusion Handling

As OF is ill-defined at occluded pixels, occlusions have to be accounted for during estimation. Classical methods either treat occlusion as outliers within a robust estimation setting [31], or conduct explicit occlusion detection, often using a forward-backward consistency check [32]. In a deep learning framework, several methods estimate jointly OF and occlusion maps. In doing so, most authors (eg. [17], [18]) observe a significant improvement on the OF estimation — an exception being [16]. Unsupervised methods also estimate occlusion maps, as they need to ignore occluded pixels in their photometric loss. [33] estimates occlusion maps by forward-backward check, [29], [22] learn occlusion detection in an unsupervised manner. Very recently, [34] proposes a self-supervised method to learn an occlusion map and uses it to filter the features warping so as to avoid ambiguity due to occlusions.

Here, we propose a very simple and lightweight way of dealing with occlusions by processing occlusion maps almost

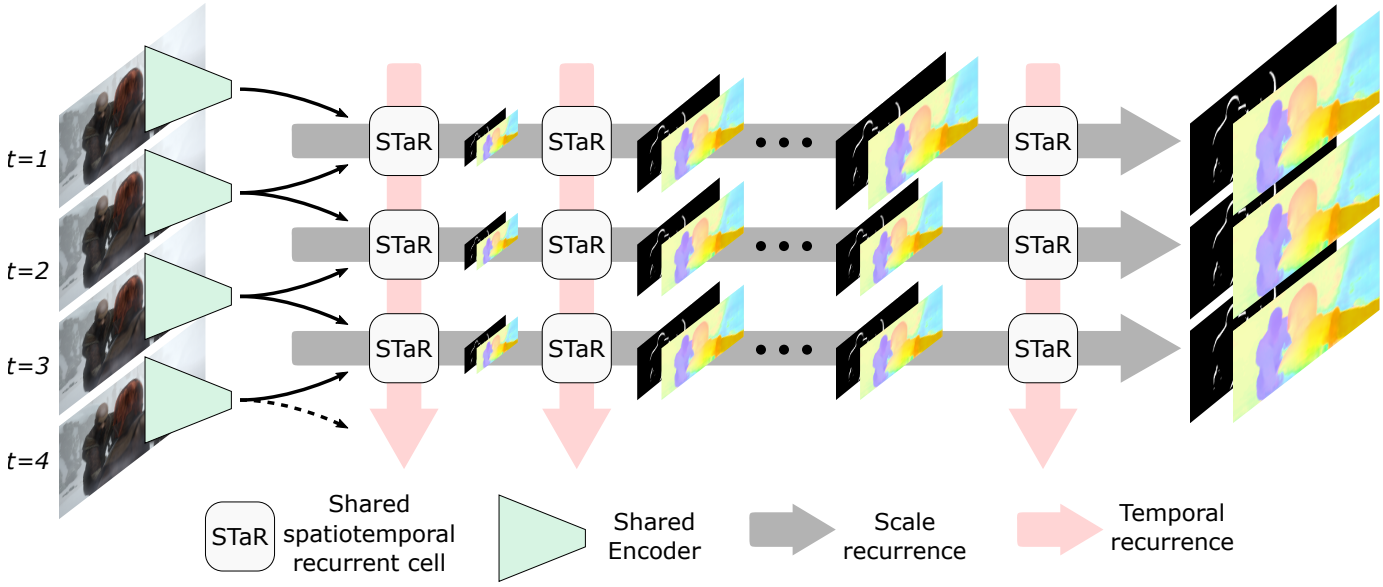


Fig. 2. Unrolled view of the proposed SpatioTemporal Recurrent architecture for multi-frame OF estimation (STaRFlow).

in the same way as OF estimates and observe a significant gain on OF accuracy in accordance with [17], [18].

## III. Proposed Approach

We propose a doubly recurrent algorithm for optical flow (OF) estimation. It is mainly the repeated application of the same *Spatio**T**empor**a**l* *R*ecurrent (STaR) cell recursively with respect to time and spatial scale on features extracted from each image of the sequence. Fig. 2 presents an unrolled representation of this recurrent "STaRFlow" model. Feature extraction uses a shared encoder (green block) which architecture comes from [13]. The scale recurrence, represented as horizontal gray arrows in Fig. 2, consists in feeding the STaR cell at each scale with the features extracted from the current frame and with the OF and occlusions coming from previous scale. The data flow related to the temporal recurrence carries learned features from one time instant to the next; it is depicted as vertical pink arrows.

The rest of this Section aims at a complete description of STaRFlow. The internal structure of the STaR cell is presented in section III-A. Then section III-B focuses on the temporal recurrence, section III-C is dedicated to occlusions handling, and section III-D presents the spatial recurrence. Finally, in Section III-E, we discuss the compound loss used for multi-frame optical estimation and the optimization process.

### A. STaR Cell

As several other recent OF estimation approaches, the proposed method builds upon PWC-Net [13], which has been designed to use well-known good practices from energy minimization methods: multi-scale pyramid, warping, cost-volume computation by correlation. These three elements are found in the architecture of the STaR cell presented in Fig. 3. It is fed by features from a siamese pyramid encoder applied to both frames. Similarly to PWC-Net, the core trainable block is a CNN dedicated to OF (blocks *CNN optical flow estimator* and *Context network* in Fig. 3). Finally, to avoid blurry results near motion discontinuities, we use the lightweight bilateral refinement of [18].

In addition to the inputs already appearing in PWC-Net (features from reference image, cost-volume from correlation of features and the upsampled flow from the previous scale), two supplementary input/output data flows are involved in the STaR cell. The first one implements the temporal recurrence leading to a multi-frame estimation. It conveys features from the highest layers of the CNN OF estimator which are fed into the CNN OF estimator at the next time step, see Sec. III-B. The second concerns the occlusion map, which undergoes essentially the same pipeline as the OF — further details on occlusions handling are given in Sec. III-C.

### B. Temporal Recurrence for Multi-Frame Estimation

The temporal connection passes features from time $t-1$ to time $t$ (Figure 3). These features are the outputs of the penultimate layer of the CNN OF estimator at $t-1$, which are compressed by a $1 \times 1$ convolution to keep the number of input channels constant from one time step to the next. They are then warped into the current first image geometry, using the previous time-step backward flow *i.e.*, the optical flow from $t$ to $t-1$. This flow is not directly accessible at inference as our network predicts the forward flow, *i.e.*, from $t-1$ to $t$. Thus, we apply our network on two frames with reversed time (from $t$ to $t-1$) to estimate the backward flow (the temporal connection being set to zero).

### C. Joint Estimation of Occlusions

As already mentioned, previous works such as [17], [18] considered the idea of estimating jointly OF and occlusion

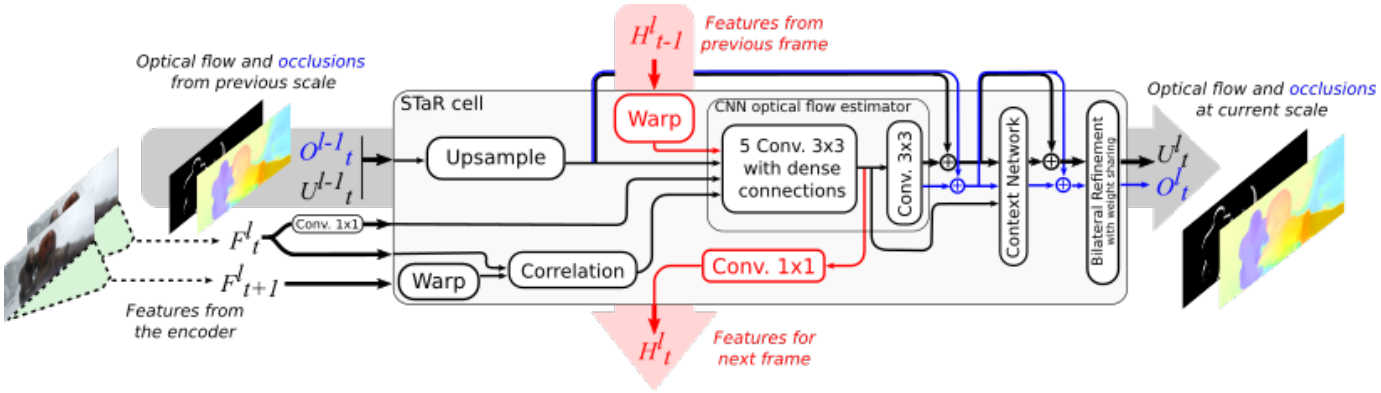


Fig. 3. Structure of the proposed SpatioTemporal Recurrent cell (STaR cell).

maps, with the purpose of improving OF estimation. In [17] occlusion maps are estimated using an extra CNN module and used as an input of the OF estimator, while [18] processes occlusion map and OF in parallel by adding an occlusion CNN estimator with the same architecture as the OF CNN estimator, but ending with a one-channel sigmoid layer. These methods, especially [18], lead to a significant increase in the number of parameters of the model.

In the STaR cell, joint estimation of OF and occlusions is done simply by adding a channel to the last convolutional layer of the CNN OF estimator (which, hence, becomes a "OF+occlusion" estimator). After a sigmoid layer, this supplementary channel gives an occlusion probability map with value between 0 (non-occluded) and 1 (occluded). Compared to [18], [17], this leads to a negligible number of extra parameters, while achieving competitive results, according to the experiments conducted in Sec. IV-C.

### D. Spatial Recurrence over Scales

We iterate on the same weights on each scale, according to the IRR approach of [18] — but unlike them we apply this coarse-to-fine process to a concatenation of the OF and the occlusion map. This allows a significant decrease in the number of parameters, while keeping estimation results almost unchanged, as shown in Sec. IV-D3.

### E. Multi-Frame Training Loss

We use $N$-frame training sequences and train our network to estimate the OFs for each pair of consecutive images. From the second image pair of the sequence, information from previous estimations is transmitted through the temporal connection. At the end of the sequence, we update the weights so as to decrease:

$$\mathcal{L} = \frac{1}{N} \sum_{t=1}^{N} \mathcal{L}_t \tag{1}$$

where $\mathcal{L}_t$ is a multi-scale and multi-task loss for image pair $(I_t, I_{t+1})$:

$$\mathcal{L}_t = \sum_{l=1}^{L} \alpha_l \left( \mathcal{L}_{\text{flow}}^{t,l} + \lambda \mathcal{L}_{\text{occ}}^{t,l} \right) \tag{2}$$

coefficients $\alpha_l$ being chosen as in [13]. The supervision of OF $u_t^l(x)$ at each time step $t$ and each scale $l$ is done as in [13] using the $L_2$ norm summed over all pixel positions:

$$\mathcal{L}_{\text{flow}}^{t,l} = \sum \left\| u_t^l - u_{t,\text{GT}}^l \right\|_2 \tag{3}$$

For the occlusion map $o_t^l$, the loss is a weighted binary cross-entropy:

$$\mathcal{L}_{\text{occ}}^{t,l} = -\frac{1}{2} \sum \left( w_t^l o_t^l \log o_{t,\text{GT}}^l + \bar{w}_t^l (1 - o_t^l) \log(1 - o_{t,\text{GT}}^l) \right) \tag{4}$$

where summation is done over all pixel positions and denoting $w_t^l = \frac{H^l \cdot W^l}{\sum o_t^l + \sum o_{t,\text{GT}}^l}$ and $\bar{w}_t^l = \frac{H^l \cdot W^l}{\sum (1-o_t^l) + \sum (1-o_{t,\text{GT}}^l)}$, $H^l$ and $W^l$ being the image size at scale $l$. As in [18] we update at each iteration the weight $\lambda$ that balances the flow loss and the occlusion loss.

## IV. Experiments

### A. Implementation Details

As proposed in [12], all models are first trained on FlyingChairs [10] and then on FlyingThings3D [11]. We then finetune on either Kitti or MPI Sintel. We use photometric and geometric data augmentations as in [18] except that for the geometric augmentations we do not apply relative transformations.

*1) Pretraining on Image Pairs on FlyingChairs:* Following [17], we first train our multi-frame architecture, except from the temporal connection, on 2D two-frame data. To supervise both OF and occlusion estimation, we use the FlyingChairsOcc dataset [18]. We train with a batch size of 8 for 600k iterations, with an initial learning rate of $10^{-4}$ which is divided by 2 every 100k iterations after the first 300k iterations.

*2) Multi-Frame Training on FlyingThings3D:* Then we train the STaRFlow model on sequences of $N = 4$ images from FlyingThings3D, the temporal data stream being initialized to zero — note that longer sequences could be exploited, at the cost of an increase in the memory space required for training. As it is the first training for the temporal connection, we start with a higher learning rate of $10^{-4}$ compared to two-frame training (as suggested by [17]) and train for 400k

TABLE I
RESULTS ON MPI SINTEL AND KITTI 2015 BENCHMARKS (TEST SETS). ENDPOINT ERROR [PX] ON SINTEL, PERCENTAGE OF OUTLIERS ON KITTI.

| Method | MPI Sintel clean | MPI Sintel final | KITTI 2015 Fl-all | Number of parameters |
|---|---|---|---|---|
| ARFlow-mv* [30] | 4.49 | 5.67 | 11.79 % | **2.37M** |
| LiteFlowNet [19] | 4.54 | 5.38 | 9.38 % | 5.37M |
| PWC-Net [13] | 4.39 | 5.04 | 9.60 % | 8.75M |
| LiteFlowNet2 [35] | 3.48 | 4.69 | 7.62 % | 6.42M |
| PWC-Net+ [36] | 3.45 | 4.60 | 7.72 % | 8.75M† |
| IRR-PWC [18] | 3.84 | 4.58 | 7.65 % | 6.36M |
| MFF* [21] | 3.42 | 4.57 | 7.17 % | 9.95M |
| ContinualFlow_ROB* [17] | 3.34 | 4.53 | 10.03 % | 14.6M† |
| SelFlow* [22] | 3.74 | 4.26 | 8.42 % | 4.79M‡ |
| MaskFlowNet [34] | **2.52** | 4.17 | **6.11 %** | N/A |
| ScopeFlow [23] | 3.59 | *4.10* | *6.82 %* | 6.36M |
| STaRFlow-ft* (*ours*) | *2.72* | **3.71** | 7.65 % | *4.77M* |

Best results are in bold characters, second ones in italic. Multi-frame methods are marked with *. †: value given in [18], ‡: value given in [30].

iterations, dividing the learning rate by 2 every 100k iterations after the first 150k iterations. We use a batch size of 4. For the ablation study, this is the final step of our training.

*3) Finetuning on MPI Sintel or Kitti:* We use the same finetuning protocol as [18] but extended to our multi-frame ($N = 4$) estimation process. For Sintel, we can supervise every time step. In KITTI, only one time step is annotated, hence we only supervise the last time-step estimation. This finetuning step is only used for benchmark submissions.

*4) Running Time:* On Sintel images ($1024 \times 436$) the inference time of STaRFlow is of 0.22 second per image pair, on a mid-range NVIDIA GTX 1070 GPU.

### B. Optical Flow Results on Benchmarks

Results of STaRFlow on benchmarks MPI Sintel and KITTI 2015 are given in Tab. I, and compared to top-leading methods and/or methods closely related to our approach. STaRFlow reaches the best EPE score on the final pass of Sintel, is second on the clean pass, and is on par with IRR-PWC on Kitti2015. Kitti2015 is characterized by very large movements of foreground objects, which generally disadvantages multi-frame methods: among them, STaRFlow still ranks second behind MFF. Regarding the number of parameters, STaRFlow ranks second behind ARFlow but outperforms it (as well as other light methods such as LiteFLowNet and SelFlow) in terms of OF precision. It is also interesting to compare STaRFlow with the related methods [17] and [18]. STaRFlow significantly outperforms ContinualFlow [17] on all benchmarks while being three times lighter. Compared to IRR-PWC [18], the benefit of the multi-frame estimation of STaRFlow clearly appears on MPI Sintel.

### C. Occlusion Estimation

Our main purpose here is to compare our solution for occlusion estimation, which shares almost all its weights with the OF estimator, to the dedicated decoder used in IRR-PWC. To do this comparison as fairly as possible, we have trained a two-frame version of STaRFlow (by removing the red connections and operators on Fig. 3), which then essentially differs from IRR-PWC by the occlusion detection process. In Tab. II, we compare F1-scores of our occlusion estimator and various methods (including IRR-PWC) on occlusion maps estimated from MPI Sintel data. Our occlusion estimation is on par with IRR-PWC while being much lighter. We also report scores of SelFlow and ScopeFlow for comparison to other state-of-the-art methods.

TABLE II
OCCLUSION MAP ESTIMATION RESULTS (F1-SCORE) ON MPI SINTEL.

| Method | Clean | Final | Parameters |
|---|---|---|---|
| ContinualFlow [17] | - | 0.48 | 14.6M |
| SelFlow [22] | 0.59 | 0.52 | 4.79M |
| IRR-PWC [18] | 0.71 | 0.67 | 6.36M |
| ScopeFlow [23] | **0.74** | **0.71** | 6.36M |
| Our occlusion estimator | 0.70 | 0.66 | **4.09M** |

Best results are in bold characters.

### D. Ablation Study

In this section, we consider the contributions of the following components of the STaRFlow model to the OF estimation: temporal recurrence and number of used frames, joint occlusion estimation and spatial recurrence. For all the experiments, our backbone is the two-frame PWC-Net architecture [13][1] that we trained as described in [18]. As this backbone does not include a bilateral refinement module, we do not include this module in the following tests. The models are trained on FlyingChairs and FlyingThings3D, without any further finetuning, and tested on the training sets of MPI Sintel and KITTI2015. All comparisons are made with the main performance metrics proposed in the benchmark websites — note that we use the revised occlusion maps provided by [18] to compute occ/noc scores on MPI Sintel.

*1) Temporal Recurrence:* Two different temporal recurrences are evaluated, with and without occlusion handling in Tab. III, and compared to the two-frame backbone. The first one, termed "TRFlow", is inspired from [17], and passes the estimated OF at time $t-1$ to the CNN OF estimator at $t$. In the second approach, denoted by "TRFeat", the temporal connection conveys learned features. "TRFeat" is the method implemented in STaRFlow and described in Sec. III-B.

According to Tab. III, using learned features in the temporal connection yields better results than passing estimated OFs, with higher EPE gains on degraded images (Sintel Final vs. Sintel Clean) and especially on the real images of KITTI2015 training dataset. Results are consistent whether an occlusion module is used or not.

The qualitative results displayed in Fig. 4–6 aim to better understand the gains brought by our temporal connection and occlusions handling. As could be expected, multi-frame estimation improves robustness to degraded image quality. This is shown in Fig. 4 which compares results on Sintel Clean and Final (blurry) images.

[1]Implementation from https://github.com/visinf/irr

TABLE III
INFLUENCE OF TEMPORAL CONNECTION AND OCCLUSION MODULES ON PERFORMANCES (MPI SINTEL AND KITTI 2015 TRAINING SETS).

| Method | Cat. | Sintel Clean [px] | | | Sintel Final [px] | | | KITTI 2015 | | Parameters | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | all | noc | occ | all | noc | occ | epe-all | Fl-all | number | relative |
| *Without joint occlusion estimation.* | | | | | | | | | | | |
| Backbone (PWC-Net) | 2F | 2.74 | 1.46 | 16.48 | 4.18 | 2.56 | 21.70 | 11.75 | 33.20 % | 8.64M | 0 % |
| Backbone + TRFlow | MF | 2.47 | **1.41** | 13.97 | 4.01 | 2.52 | 20.00 | 11.27 | 33.77 % | 8.68M | +0.5 % |
| Backbone + TRFeat | MF | **2.45** | 1.44 | **13.36** | **3.76** | **2.46** | **17.82** | **9.94** | **32.12 %** | 12.31M | +42.5 % |
| *With joint occlusion estimation.* | | | | | | | | | | | |
| Backbone | 2F | 2.46 | 1.32 | 14.82 | 3.96 | 2.47 | 20.06 | 10.58 | 31.28 % | 8.68M | +0.5 % |
| Backbone + TRFlow | MF | 2.17 | 1.23 | 12.33 | 3.90 | 2.50 | 19.11 | 10.82 | 32.51 % | 8.73M | +1.0 % |
| Backbone + TRFeat | MF | **2.09** | **1.21** | **11.63** | **3.43** | **2.24** | **16.24** | **8.79** | **28.18 %** | 12.38M | +43.3 % |
| *With joint occlusion estimation and spatial recurrence.* | | | | | | | | | | | |
| Backbone | 2F | 2.29 | **1.20** | 14.03 | 3.72 | **2.32** | 18.77 | 10.74 | 31.35 % | 3.37M | −61.0 % |
| Backbone + TRFlow | MF | 2.20 | 1.25 | 12.40 | 3.98 | 2.56 | 19.38 | 11.00 | 35.23 % | 3.38M | −60.9 % |
| Backbone + TRFeat | MF | **2.10** | 1.22 | **11.67** | **3.49** | **2.32** | **16.15** | **9.26** | **30.75** % | 4.37M | −49.4 % |

Best results are in bold characters. Fl-all, on KITTI, is the percentage of outliers (epe $> 3$ px).
2F (resp. MF) refers to two-frame (resp. multi-frame) methods. *TR* stands for *temporal recurrence*.

Multi-frame estimation also allows temporal inpainting: for a region occluded at time $t+1$ but visible at $t$ and previous time steps, the previously estimated motion can be used to predict the motion between $t$ and $t+1$. This could be observed on the Sintel example shown in the upper left part of Fig. 1: the right knee of the central character, although occluded in the next frame, is correctly estimated by STaRFlow. Fig. 5 displays an example extracted from KITTI2015 training set where temporal connection and occlusions estimation are both required to correctly estimate motion of the roadsign on the lower right part of the image, which is occluded in the next frame. Finally, Fig. 6 shows that our temporal connection with learned features yields increased sensitivity to small object motion compared to the backbone and also to TRFlow.

*2) Occlusion Handling:* Comparison of methods with and without occlusion estimation in Tab. III shows that adding the task of detecting occlusions consistently helps OF estimation. This is true for two-frame and multi-frame methods.

*3) Spatial Recurrence:* The lower part of Tab. III is devoted to the spatial recurrence, *i.e.* the iterations on the same weights over scales in the coarse-to-fine multi-level estimation [18]. While OF precision is only marginally affected by this implementation, large gains in terms of number of parameters are obtained with respect to the PWC-Net backbone (see last column).

*4) Impact of the Number of Frames at Test Time:* Recall that $N = 4$ frames are used for training multi-frame models (TRFlow and TRFeat). It means that, at training time, the temporal connection is reinitialized to zero every 4 frames, essentially to avoid an increased memory cost, beyond the capacity of the hardware. However, *at test time*, the temporal connection can be exploited over a different time horizon. This is the object of Tab. IV, which compares temporal connections TRFlow and TRFeat when increasing the number of frames $N'$ used at test time. Each line of the Table presents scores computed for the OF estimated between time instants $N'-1$ and $N'$.

According to Tab. IV, performance improves more for TRFeat than for TRFlow when $N'$ increases. This is particularly true for degraded (Sintel Final) or real images (KITTI), or in occluded regions. Furthermore, we observe that TRFeat still improves using $N' = 5$ frames. TRFeat, by propagating learned features in the temporal connection instead of OF, exploits more efficiently long term memory than TRFlow and appears even able to learn a temporal continuity beyond the number of frames used for training.

This can also be seen on the qualitative results presented on Fig. 7. Estimations using $N' = 3$ and $N' = 4$ (columns 2 and 3) are presented for TRFlow and TRFeat. The fact that the object is very close to the image border makes the problem difficult. For the two methods, using 3 frames is not enough to correctly estimate the object's contour. TRFeat manage to resolve the contour with a 4th frame, while TRFlow still fails to do it.

TABLE IV
IMPACT OF THE NUMBER OF FRAMES $N'$ USED AT TEST TIME.

Backbone + occ + TRFlow + SR

| $N'$ | Sintel Clean | | | Sintel Final | | | Kitti15 | |
|---|---|---|---|---|---|---|---|---|
| | all | noc | occ | all | noc | occ | epe-all | Fl-all |
| 2 | 2.36 | 1.27 | 14.17 | 4.05 | 2.57 | 20.06 | 12.53 | 35.95 % |
| 3 | **2.17** | **1.24** | **12.29** | **3.95** | **2.56** | **19.03** | 11.26 | 35.35 % |
| 4 | 2.20 | 1.25 | 12.40 | 3.98 | **2.56** | 19.38 | 11.01 | 35.27 % |
| 5 | 2.20 | 1.26 | 12.37 | 3.98 | **2.56** | 19.30 | **10.94** | **35.17 %** |
| 6 | 2.20 | 1.26 | 12.33 | 3.98 | 2.58 | 19.11 | **10.94** | 35.19 % |

Backbone + occ + TRFeat + SR

| $N'$ | Sintel Clean | | | Sintel Final | | | Kitti15 | |
|---|---|---|---|---|---|---|---|---|
| | all | noc | occ | all | noc | occ | epe-all | Fl-all |
| 2 | 2.40 | 1.30 | 14.34 | 4.04 | 2.55 | 20.12 | 12.01 | 34.22 % |
| 3 | 2.10 | 1.23 | 11.60 | 3.58 | 2.35 | 16.90 | 9.95 | 31.49 % |
| 4 | 2.10 | **1.22** | 11.67 | 3.49 | 2.32 | 16.15 | 9.26 | 30.78 % |
| 5 | **2.08** | **1.22** | **11.36** | **3.43** | **2.27** | **15.99** | 9.17 | **30.66 %** |
| 6 | 2.09 | **1.22** | 11.52 | 3.50 | 2.32 | 16.25 | **9.14** | 30.69 % |

Best results are in bold characters.

## V. CONCLUSION

We have presented STaRFlow, a new lightweight CNN method for multi-frame OF estimation with occlusion handling. It involves a unique computing cell which recursively


Fig. 4. Multi-frame estimation provides robustness to degraded image quality: results on Sintel clean (upper row) and Sintel Final pass (lower row).


Fig. 5. Both the occlusion and temporal coherence modules are needed here to resolve the motion of the lower right roadsign.

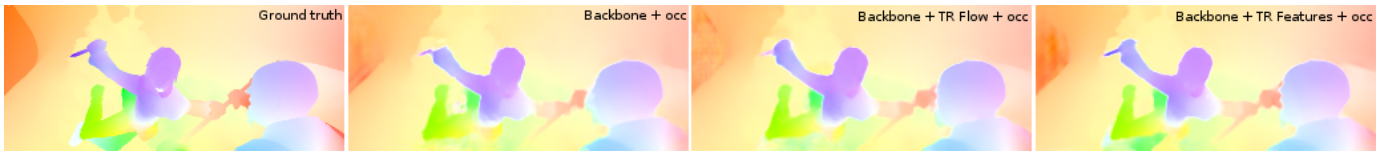


Fig. 6. Our temporal recurrent cell improves optical flow estimation of small objects.

processes both a spatial data flow in a coarse-to-fine multi-scale scheme and a temporal flow which conveys learned features. Using learned features in the temporal recurrence allows better exploitation of temporal information than propagating OF estimates as proposed in [17]. STaRFlow builds upon approaches such as [37], [18] based on the repeated use of the same weights over a scale recurrence but extends this idea to a double time-scale recurrence. Moreover, we have also shown that occlusion estimation can be done with a minimal number of extra parameters, simply by adding a dedicated layer to the output tensor of the CNN OF estimator. STaRFlow gives state-of-the-art results on the two benchmarks MPI Sintel and Kitti2015, even outperforming, at the time of writing, all previously published methods on Sintel final pass. Moreover, STaRFlow is lighter than all other two-frame or multi-frame methods with comparable performance.

Quantitative and qualitative evaluations on MPI Sintel and KITTI2015 show that STaRFlow improves OF quality on degraded images and on small objects thanks to temporal redundancy, and is also able to achieve efficient temporal inpainting in occluded areas. Our experiments also confirm conclusions of [18], [17] that learning to predict occlusions consistently improves OF estimation. Moreover, our implementation, based on sharing almost all weights between OF and occlusion estimation, further indicates that these two tasks are closely related one to the other.

## ACKNOWLEDGMENT

The authors gratefully thank the French Agence de l'Innovation de Défense (AID) for its fundings.

## REFERENCES

[1] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[2] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art," *arXiv preprint arXiv:1704.05519*, 2017.

[3] Z. Chen, J. Cao, Y. Tang, and L. Tang, "Tracking of moving object based on optical flow detection," in *International Conference on Computer Science and Network Technology*, vol. 2. IEEE, 2011, pp. 1096–1099.

[4] R. Xu, X. Li, B. Zhou, and C. C. Loy, "Deep flow-guided video inpainting," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3723–3732.

[5] W. Zhao and H. S. Sawhney, "Is super-resolution with optical flow feasible?" in *European Conference on Computer Vision*. Springer, 2002, pp. 599–613.

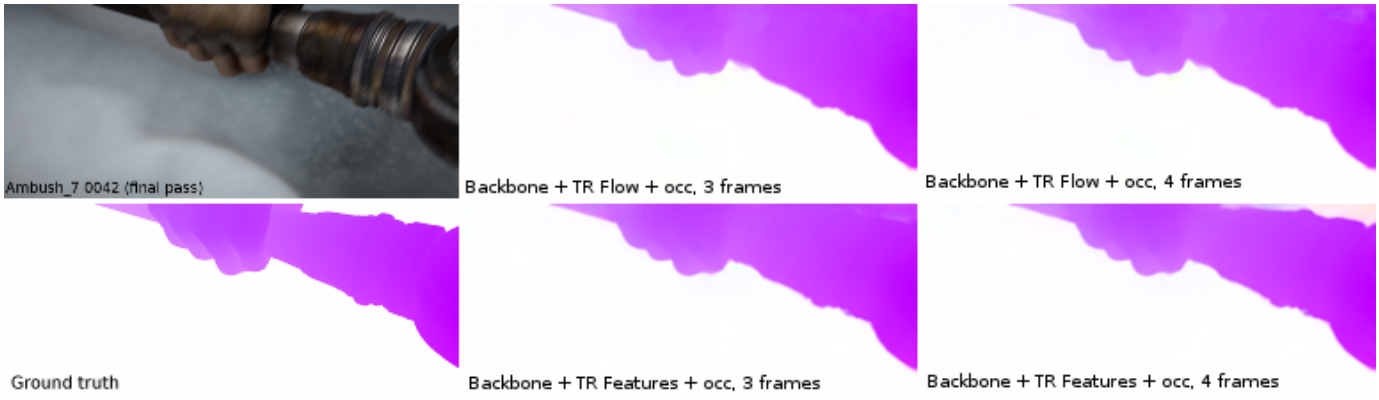


Fig. 7. The benefit of exploiting more frames in OF estimation for sequence Ambush7 of Sintel Final.


[6] D. Mitzel, T. Pock, T. Schoenemann, and D. Cremers, "Video super resolution using duality based TV-L1 optical flow," in *Joint Pattern Recognition Symposium*. Springer, 2009, pp. 432–441.

[7] K. Nasrollahi and T. B. Moeslund, "Super-resolution: a comprehensive survey," *Machine vision and applications*, vol. 25, no. 6, pp. 1423–1468, 2014.

[8] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "DeMoN: Depth and motion network for learning monocular stereo," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5038–5047.

[9] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[10] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.

[11] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.

[12] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017, p. 6.

[13] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[14] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conference on Computer Vision*. Springer, 2012, pp. 611–625.

[15] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition*, 2015.

[16] E. Ilg, T. Saikia, M. Keuper, and T. Brox, "Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation," in *European Conference on Computer Vision*. Springer, 2018, pp. 626–643.

[17] M. Neoral, J. Šochman, and J. Matas, "Continual occlusion and optical flow estimation," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 159–174.

[18] J. Hur and S. Roth, "Iterative residual refinement for joint optical flow and occlusion estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5754–5763.

[19] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[20] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2017.

[21] Z. Ren, O. Gallo, D. Sun, M.-H. Yang, E. Sudderth, and J. Kautz, "A fusion approach for multi-frame optical flow estimation," in *IEEE Winter Conference on Applications of Computer Vision*. IEEE, 2019, pp. 2077–2086.

[22] P. Liu, M. R. Lyu, I. King, and J. Xu, "SelFlow: Self-supervised learning of optical flow," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[23] A. Bar-Haim and L. Wolf, "ScopeFlow: Dynamic scene scoping for optical flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7998–8007.

[24] C.-M. Wang, K.-C. Fan, C.-T. Wang *et al.*, "Estimating optical flow by integrating multi-frame information." *J. Inf. Sci. Eng.*, vol. 24, no. 6, pp. 1719–1731, 2008.

[25] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," 1981.

[26] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer, "Modeling temporal coherence for optical flow," in *IEEE International Conference on Computer Vision*. IEEE, 2011, pp. 1116–1123.

[27] R. Kennedy and C. J. Taylor, "Optical flow with geometric occlusion estimation and fusion of multiple frames," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer, 2015, pp. 364–377.

[28] D. Maurer and A. Bruhn, "ProFlow: Learning to predict optical flow," in *British Machine Vision Conference*, 2018.

[29] J. Janai, F. Guney, A. Ranjan, M. Black, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions," in *European Conference on Computer Vision*, 2018, pp. 690–706.

[30] L. Liu, J. Zhang, R. He, Y. Liu, Y. Wang, Y. Tai, D. Luo, C. Wang, J. Li, and F. Huang, "Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6489–6498.

[31] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European conference on computer vision*. Springer, 2004, pp. 25–36.

[32] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez, "Symmetrical dense optical flow estimation with occlusions detection," *International Journal of Computer Vision*, vol. 75, no. 3, pp. 371–385, 2007.

[33] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[34] S. Zhao, Y. Sheng, Y. Dong, E. I. Chang, Y. Xu *et al.*, "MaskFlowNet: Asymmetric feature matching with learnable occlusion mask," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6278–6287.

[35] T.-W. Hui, X. Tang, and C. C. Loy, "A lightweight optical flow CNN–revisiting data fidelity and regularization," *arXiv preprint arXiv:1903.07414*, 2019.

[36] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Models matter, so does training: An empirical study of CNNs for optical flow estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[37] P. Hu, G. Wang, and Y.-P. Tan, "Recurrent spatial pyramid CNN for optical flow estimation," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2814–2823, 2018.