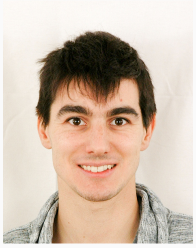

STaRFlow: A SpatioTemporal Recurrent Cell for Lightweight Multi-Frame Optical Flow Estimation

Pierre Godet* (https://pgodet.github.io), Alexandre Boulch†, Aurélien Plyer*, Guy Le Besnerais*
* DTIS, ONERA, Université Paris-Saclay, FR-91123 Palaiseau, France ; † valeo.ai, Paris, France

ICPR2020, paper **#786**, poster session **T3.1**

Previous work on optical flow estimation

- Optical flow is the apparent displacement field between two frames of a video sequence.
- Current state-of-the-art results are obtained with CNNs[1] [7, 6, 3].
- Our work is mainly based on:

  → **IRR-PWC** [3] which **iterates on the same weights** for the different levels of a **multi-scale** estimation process.

  → **ContinualFlow** [3] which proposes a recurrent **multi-frame** process by giving the **optical flow estimation from the previous frame** as an input for the next estimation.

[1] CNNs : Convolutional Neural Networks

Training data, schedule and multi-frame loss

Training data and schedule:

- We first pre-train on **image pairs** from **FlyingChairs** [2].
- We then train on **sequences of** $N = 4$ **images** from **FlyingThings3D** [4].
- We optionally finetune on **sequences of** $N = 4$ **images** from **MPI Sintel** [1] or **KITTI** [5].

Multi-frame and multi-scale loss (over $N-1$ consecutive image pairs and L scale levels):

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

The STaRFlow architecture

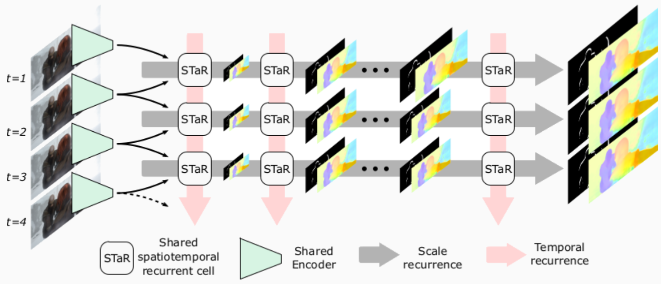

Fig. 1: STaRFlow is reccurent both in space and time. It is based on a unique SpatioTemporal recurrent cell. The same weights are used for optical flow and occlusion estimation, in the whole network except from the very last layer.

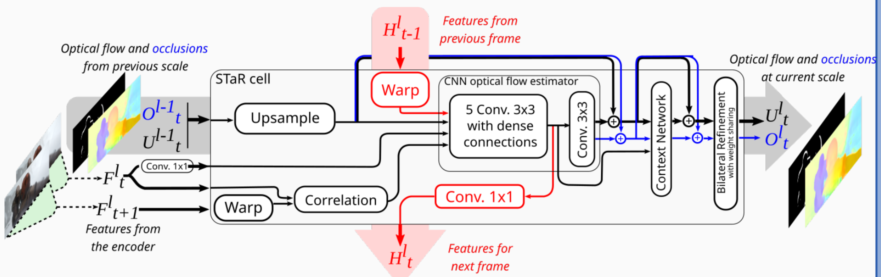

Fig. 2: The STaRCell uses a memory state that accumulates information from the past and helps the estimation process in the future. To do so, a **learned feature map** is passed from one time step to the next.

Qualitative results
(obtained after training on FlyingChairs → FlyingThings3D)

Fig. 3: Results on a sample from Sintel Training.

Fig. 4: Results on real data, from the nuScenes dataset.

Ablation study

Method	Cat.	Sintel Clean [px] all	noc	occ	Sintel Final [px] all	noc	occ	KITTI 2015 epe-all	Fl-all	Parameters number	relative
Without joint occlusion estimation.											
Backbone (PWC-Net [7])	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 %

Fig. 5: Endpoint error (epe) on MPI Sintel and KITTI 2015 training sets.

	Backbone + occ + *TRFlow* + SR					Backbone + occ + *TRFeat* + SR				
	Sintel Final			Kitti15		Sintel Final			Kitti15	
N'	all	noc	occ	epe-all	Fl-all	all	noc	occ	epe-all	Fl-all
2	4.05	2.57	20.06	12.53	35.95 %	4.04	2.55	20.12	12.01	34.22 %
3	**3.95**	**2.56**	**19.03**	11.26	35.35 %	3.58	2.35	16.90	9.95	31.49 %
4	3.98	**2.56**	19.38	11.01	35.27 %	3.49	2.32	16.15	9.26	30.78 %
5	3.98	**2.56**	19.30	**10.94**	**35.17 %**	**3.43**	**2.27**	**15.99**	9.17	**30.66 %**
6	3.98	2.58	19.11	**10.94**	35.19 %	3.50	2.32	16.25	**9.14**	30.69 %

Fig. 6: Impact of the number of frames N' used at test time.

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*.
SR stands for *scale recurrence* (that is iterating on the same weights for all levels).
TRFlow is our re-implementation of the multi-frame process of ContinualFlow.
TRFeat is the multi-frame process of STaRFlow, based on learned features.

Take-home points

- Our **temporal recurrence** using **learned features** improves estimation in occluded regions, on small objects, and on degraded quality images.
- **Sharing weights** for occlusion and optical flow estimation, and for every scale and frame, leads to a **lightweight** and **state-of-the-art** method.
- STaRFlow presents **very good results on real data** even when **trained exclusively on synthetic data**.

References

[1] Daniel J Butler et al. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.
[2] A. Dosovitskiy et al. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, 2015.
[3] J. Hur and S. Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, 2019.
[4] N. Mayer et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.
[5] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.
[6] M. Neoral et al. Continual occlusions and optical flow estimation. *ACCV*, 2018.
[7] D. Sun et al. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.