OmniFlowNet: a Perspective Neural Network Adaptation for Optical Flow Estimation in Omnidirectional Images

Submission 833

Charles-Olivier Artizzu[1], Haozhou Zhang[2], Guillaume Allibert[1] and Cédric Demonceaux[2]

[1]Université Côte d'Azur, CNRS, I3S, France.

[2]VIBOT ERL CNRS 6000, ImViA, Université Bourgogne Franche-Comté, France.

25th International Conference on Pattern Recognition
10-15 January 2021, Milan, Italy

Context

Goal:
Optical flow estimation in spherical images using deep learning approaches.

Issues:
- No omnidirectional annotated dataset;
- Strong images distortions due to projection.

Solutions: **Adaptation** of perspective networks taking into account distortions by modifying the convolution.

Equirectangular image of an outdoor car scene.
Polar regions are highly distorted

Optical flow: estimation of the apparent motion between two images.

Methods of computation:

- 1981 Variatonal: (Horn and Schunck 1981) (Lucas and Kanade 1981)
- 2011 Phase-Based methods: (Radgui et al. 2011)
- 2013 Unsupervised Deep Learning: DeepFlow (Weinzaepfel et al. 2013)
- 2015 Supervised Deep Learning: FlowNet (Dosovitskiy et al. 2015)
- 2020 LiteFlowNet2 (Hui, Tang, and Loy 2020)

Optical Flow from Sintel Dataset (Butler et al. 2012)

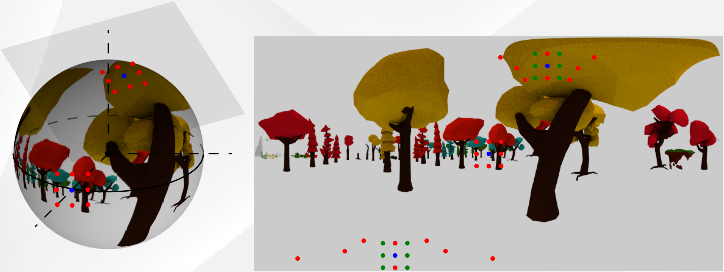
Adaptation to spherical

Solutions to adapt to spherical distortions:

- **Build an Omnidirectional annotated dataset**
- **Change the entire features map**:
  Fast Fourier Transformation (Cohen et al. 2018),
  Polyhedra (Lee et al. 2018)
- **Adapt the perspective CNN kernels to equirectangular**:
  Linear Adaptation (Su and Grauman 2017),
  Spherical Adaptation (Fernandez et al. 2020)

Proposed solution

Local perspective projection of CNN kernels on the sphere:

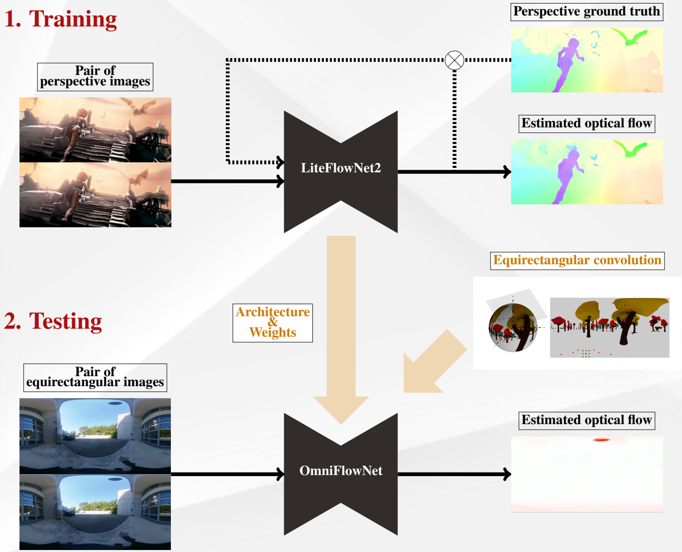


Example of kernels with different latitude and longitude.
Blue: center of kernel. Green: Perspective kernel. Red: Spherical kernel.

Proposed solution available on any CAFFE (Jia et al. 2014) based CNN.

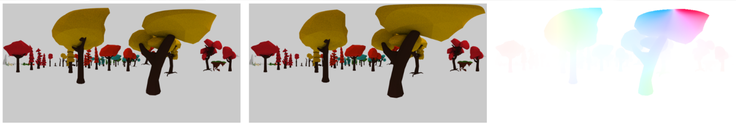
Proposed solution

Implemented on **LiteFlowNet2** (Hui, Tang, and Loy 2020).
Weights from **LiteFlowNet2** authors (training MPI Sintel).



Validation on vitual datasets

Virtual scenes built on **Blender**: 3 scenes, 4 camera orientations.
Ground truth optical flow extracted using *Vector Pass* (Ranjan et al. 2020).



Comparison metrics used (Baker et al. 2007):
Angular Error (in degrees) and **Endpoint Error** (in pixels):

$$AE = cos^{-1}\left(\frac{1 + u \cdot u_{gt} + v \cdot v_{gt}}{\sqrt{1 + u^2 + v^2}\sqrt{1 + u_{gt} + v_{gt}}}\right)$$

$$EE = \frac{1}{N}\sum\sqrt{(u_{gt} - u)^2 + (v_{gt} - v)^2}$$

with (u, v) the estimated flow and (u_{gt}, v_{gt}) the ground truth.

Average on 1200 frames	Cartoon Tree		Forest		Low Poly Model	
	AE	EE	AE	EE	AE	EE
LiteFlowNet2	63.07	5.60	58.11	10.61	58.03	7.66
OmniFlowNet	**54.74**	**4.49**	**55.05**	**9.72**	**55.21**	**7.23**

Validation on real videos

Real scenes were filmed using a Theta Ricoh Z1:

- *Apple 1* and *Ball 3*: moving indoor scene / fixed camera;
- *Ball 1* and *Ball 2*: moving outdoor scene / fixed camera;
- *Car 1* and *Car 2*: fixed outdoor scene / moving camera.

Same performances in the equatorial region.

Better performances of OmniFlowNet in the polar regions.

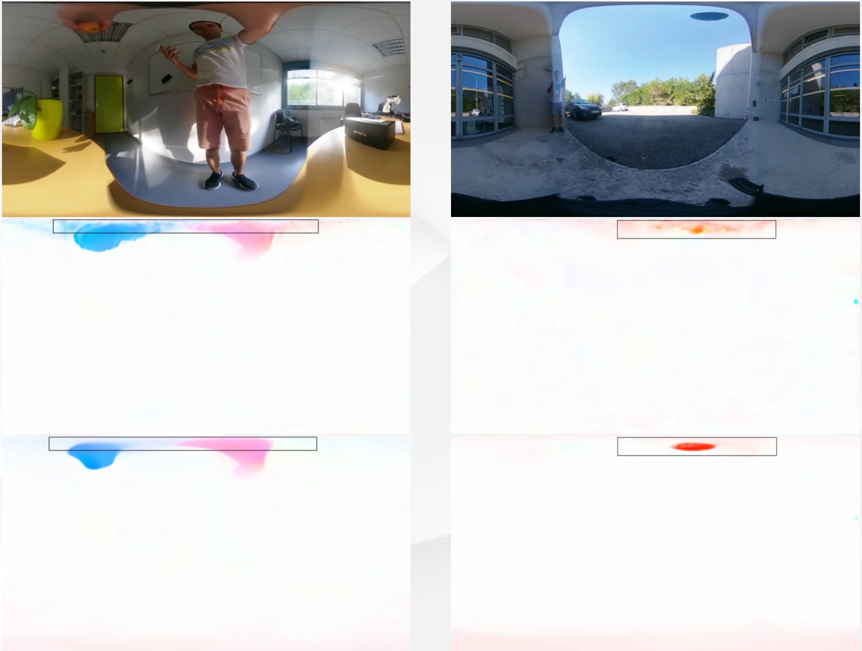
Complete video available on:
`http://www.i3s.unice.fr/~allibert/Videos/icpr20_video.mp4`.

Validation on real videos

RGB Input

LiteFLowNet2

OmniFlowNet



Conclusion

OmniFlowNet:

- perspective CNN adapted to equirectangular images;
- plugin transferable on any CAFFE network;
- no extra training needed;
- no slowdown in the time execution;
- proven performances on real and virtual datasets.

Code and dataset available on: `https://github.com/coatz/OmniFlowNet`.

Thank you for your attention.