Optical Flow Estimation using Spatial Pyramid Network

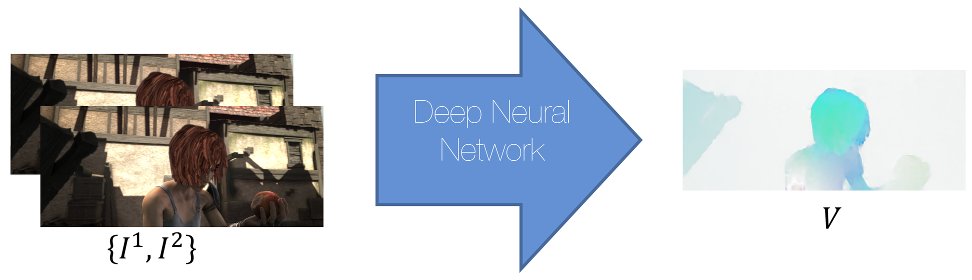
Anurag Ranjan and Michael Black
Perceiving Systems, MPI for Intelligent Systems

MAX-PLANCK-GESELLSCHAFT

Problem

Estimate two-frame optical flow using an end-to-end deep learning approach.

Given two image frames $\{I^1, I^2\}$, estimate the motion of each pixel in I^1 to I^2 given by the flow field V.

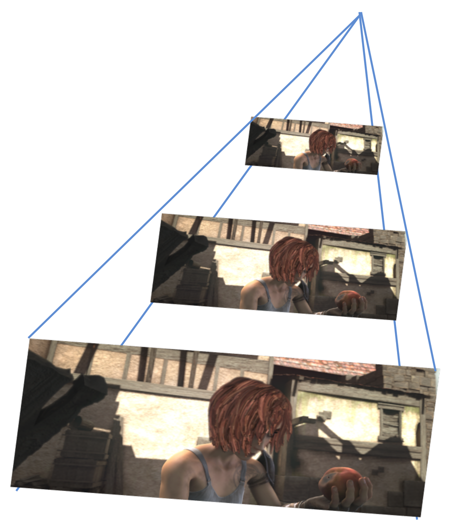


Challenges:

- Convolutions over two frames does not make sense due to large motions.
- Previous attempts at solving using deep networks has poor performance on slow motions [2].
- Shallow networks can not resolve long range matching.

Idea

- What if the motions were small?
- Use classical pyramid flow with deep learning.
- Train a deep network, G_k at each level of the pyramid.
- G_k predicts a small residual flow, v_k at each level given frames $\{I_k^1, I_k^2\}$.
- Compute the full flow V_k at a level iteratively using



$$V_k = V_{k-1} + v_k$$

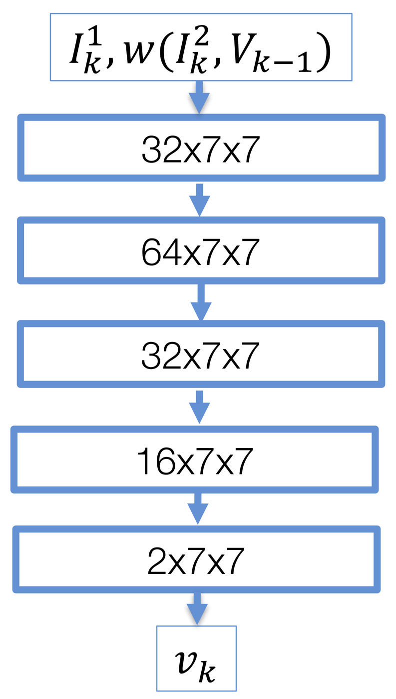
- **Warping**: Warp the second frame I_k^2 input to network G_k with flow V_{k-1} of the previous level. Residual flow at level k,

$$v_k = G_k(I_k^1, w(I_k^2, V_{k-1}))$$

where, w is the warping function.

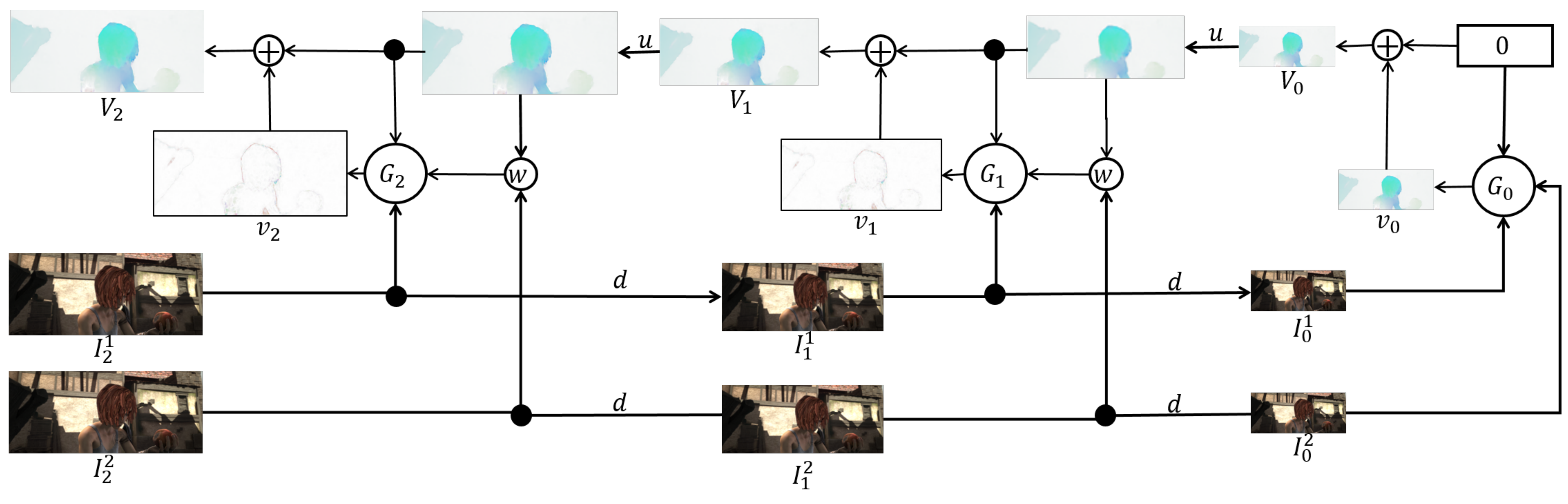
Network Training

- Each network is trained to estimate residual flow.
- Minimizing the End-point-error (EPE) loss.
- Trained using Flying Chairs dataset [2].



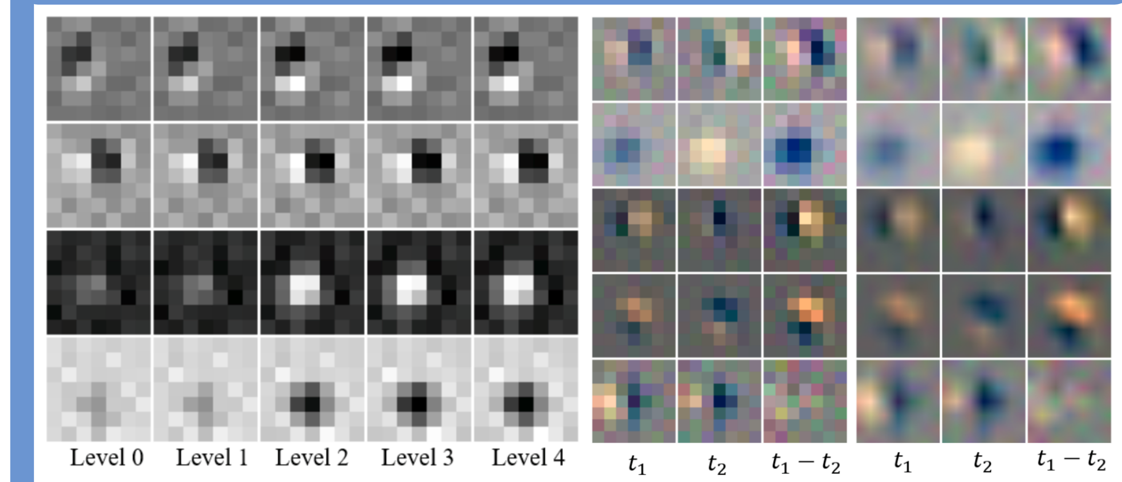
The Network G_k

Spatial Pyramid Network



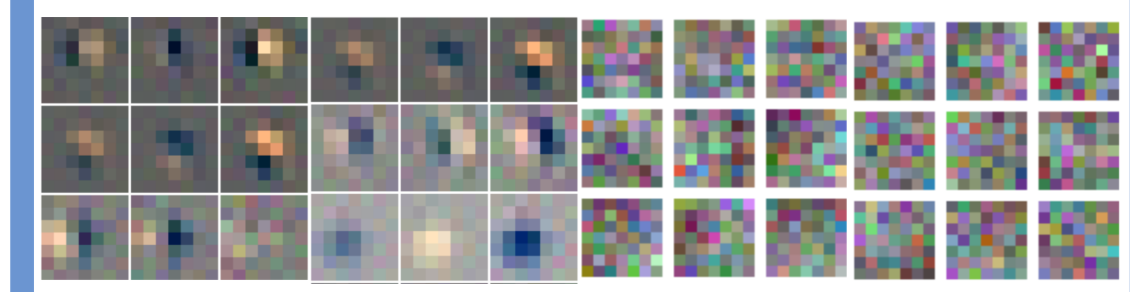
The network G_0 computes the residual flow v_0 at the lowest level of the pyramid using the low resolution images. At each pyramid level, the network G_k computes a residual flow v_k using $\{I_k^1,\ w(I_k^2,\ V_{k-1})\}$ which propagates to the upper levels of the pyramid.

Spatio-temporal Filters



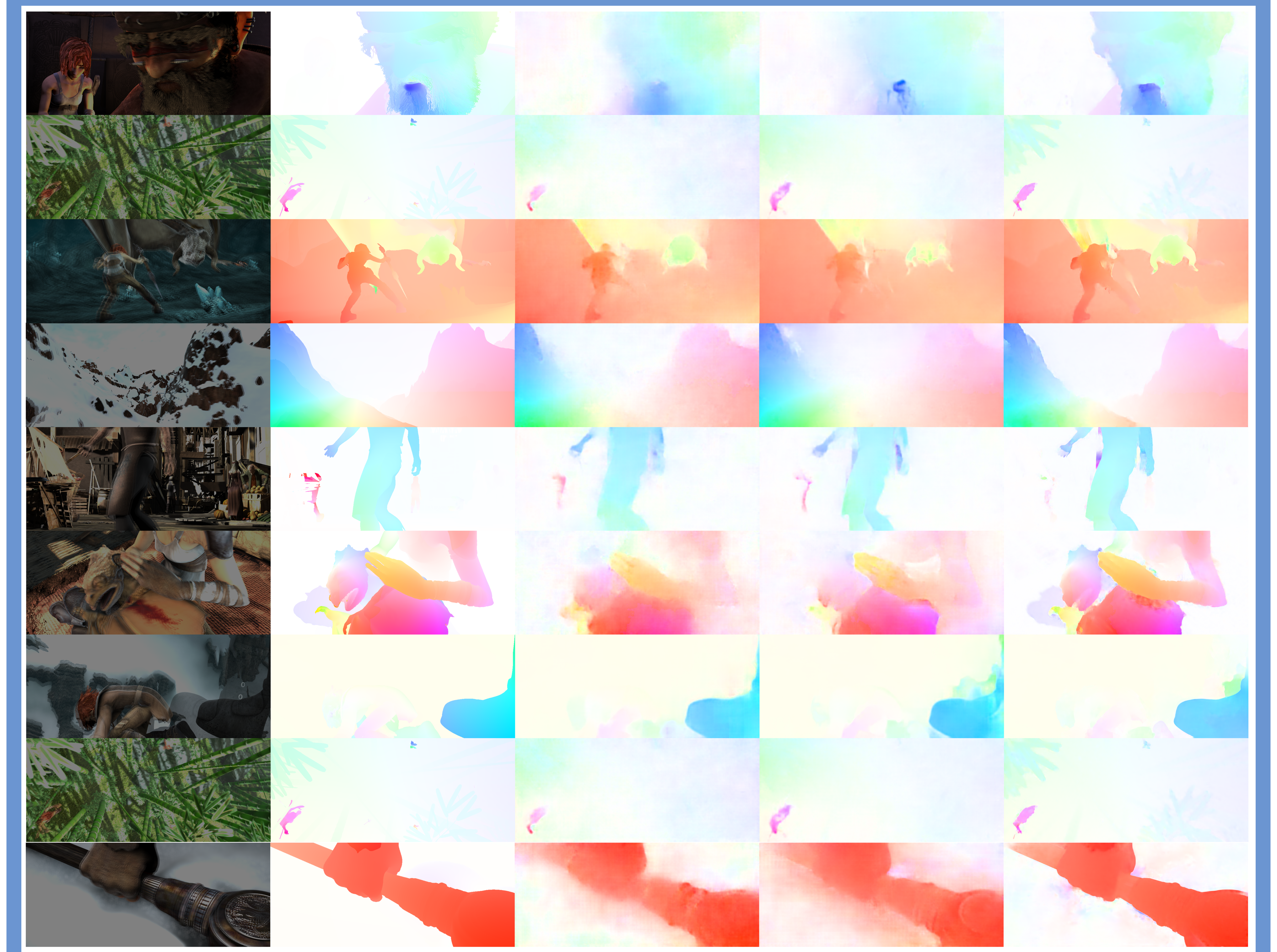
Evolution of Filters across pyramid levels. The filters become sharper on the lower levels to capture higher resolution features.

Spatial (t_1, t_2) and temporal $(t_1 - t_2)$ filters. The temporal filters are obtained as a difference of spatial filters of frame pair.



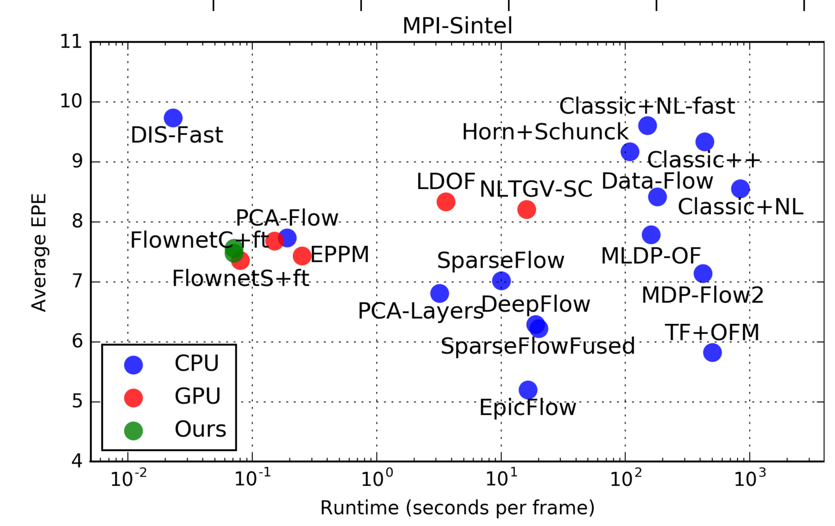
Comparing SPyNet and FlowNet filters from the first layers. While FlowNet's filters are random looking, our filters are more Gabor-like resembling cortical areas MT and V1.

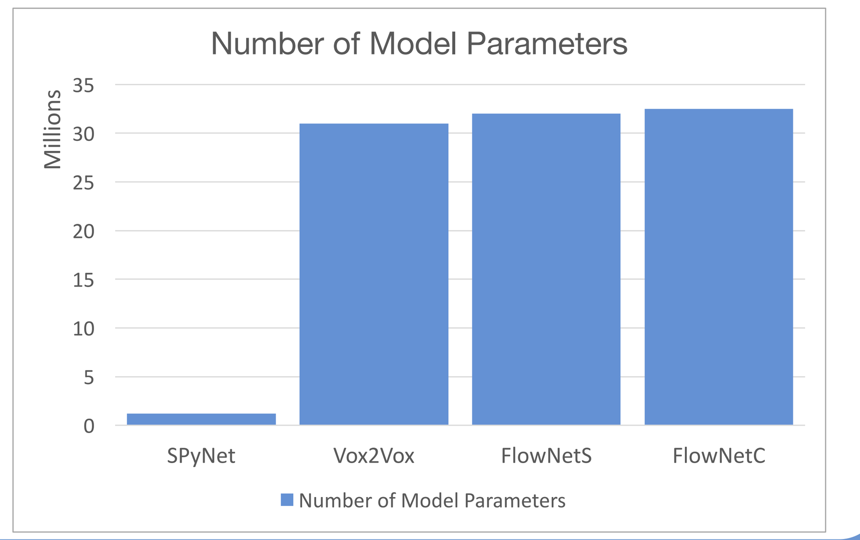
Examples



Evaluation

Method	Sintel Clean Train	Sintel Clean Test	Sintel Final Train	Sintel Final Test	Kitti Train	Kitti Test	Middlebury Train	Middlebury Test	Flying Chairs Test	Time(s) on GPU
Classic+NLP	4.13	6.73	5.90	8.29	-	-	0.22	0.32	3.93	-
FlowNetS	4.50	7.42	**5.45**	**8.43**	**8.26**	-	1.09	-	2.71	0.080
FlowNetC	4.31	7.28	5.87	8.81	9.35	-	1.15	-	2.19	0.150
SPyNet	**4.12**	**6.69**	5.57	**8.43**	9.12	-	**0.33**	**0.58**	2.63	**0.069**
FlowNetS+ft	3.66	6.96	4.44	**7.76**	**7.52**	**9.1**	0.98	-	3.04	0.080
FlowNetC+ft	3.78	6.85	5.28	8.51	8.79	-	0.93	-	2.27	0.150
SPyNet+ft	**3.17**	**6.63**	**4.32**	8.34	8.25	10.1	**0.33**	**0.58**	3.07	**0.069**
SpyNet+ft*	-	-	-	-	**3.36**	**4.1**	**-**	**-**	-	**0.069**





References

[1] Denton, Emily L., Soumith Chintala, and Rob Fergus. "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks."Advances in neural information processing systems. 2015.
[2] Fischer, Philipp, et al. "Flownet: Learning optical flow with convolutional networks." arXiv preprint arXiv:1504.06852 (2015).