Bringing a Blurry Frame Alive at High Frame-Rate with an Event Camera

戴玉超
西北工业大学
daiyuchao@nwpu.edu.cn

What is an event camera?

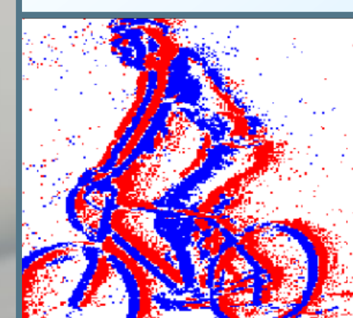
Event Camera

0.02ms →

High Speed Camera (mvBlueFox)

>10ms →

What is an event camera?

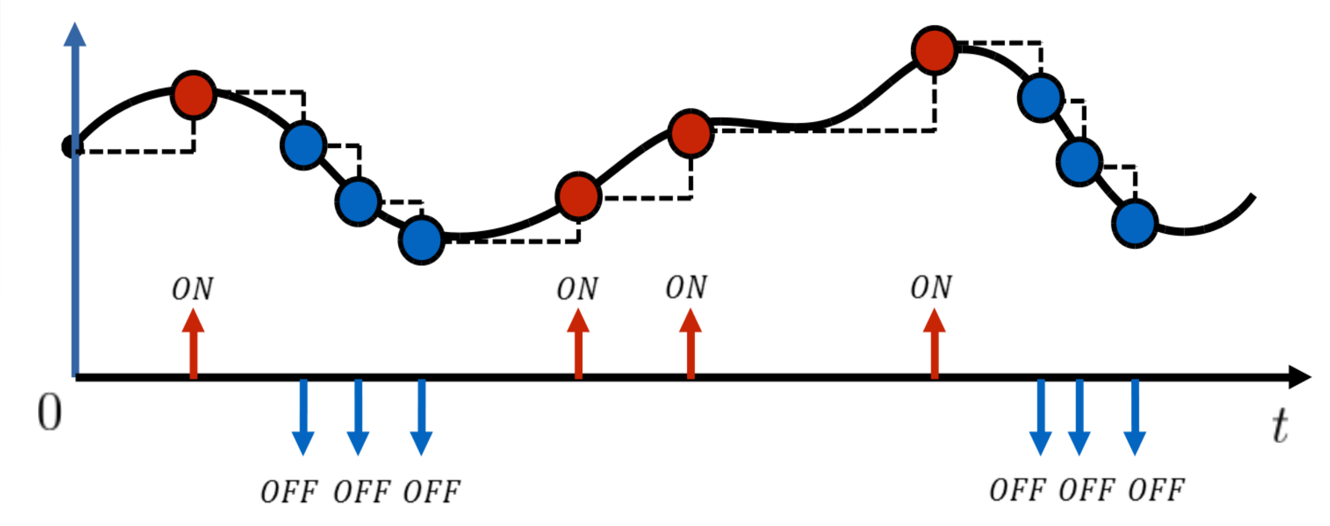


DVS: D

APS: A

DAVIS: Dynamic and Active-pixel Vision Sensor

What is an event camera?

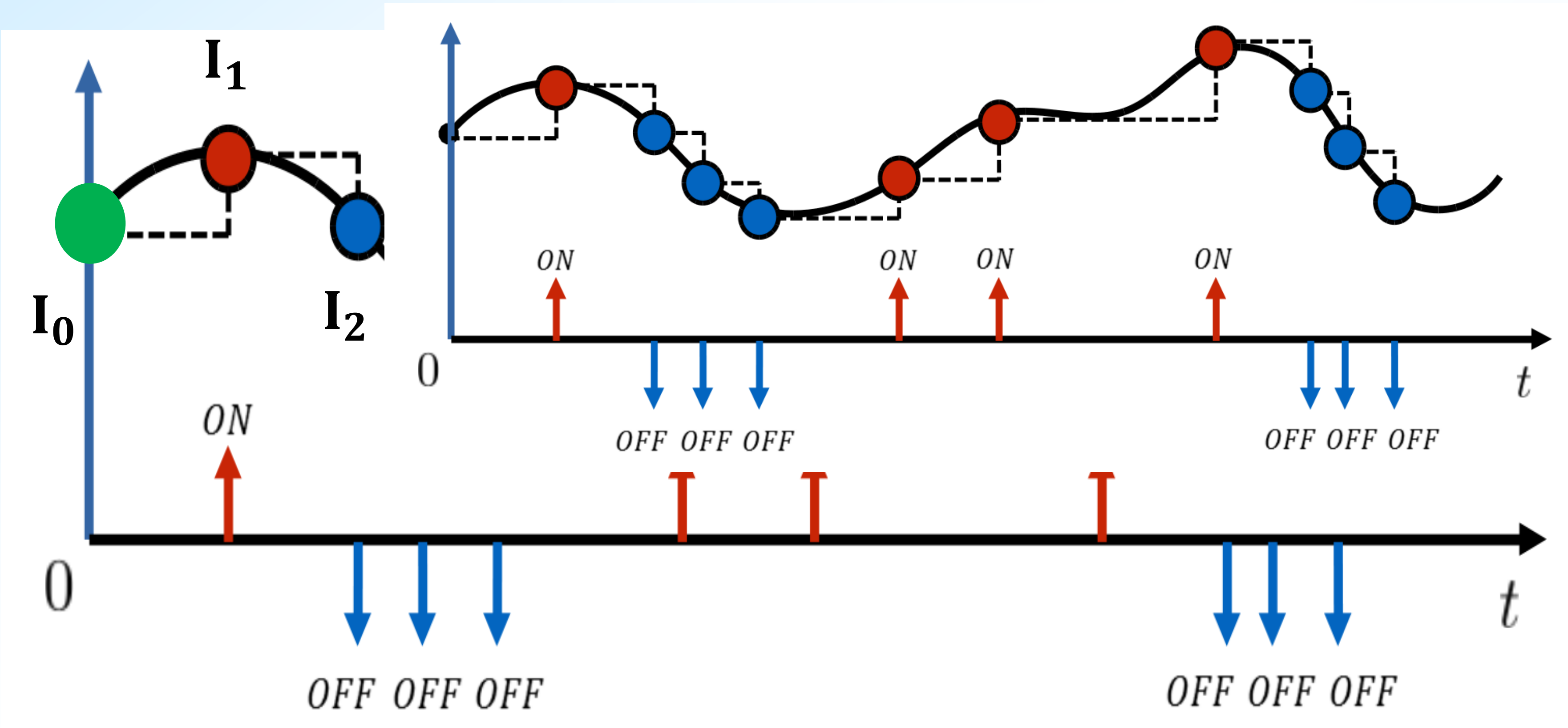


- Each event contains:

Timestamp (µs)	Pixel (x, y)	Polarity (ON, OFF)

- Events signify temporal contrast

What is an event camera?



If $\log(\frac{I_1}{I_0}) > c, \sigma = 1$ If $\log\left(\frac{I_2}{I_1}\right) > -c, \sigma = -1$

Motivation

- Event cameras are more robust to low lighting and highly dynamic scenes than traditional cameras.

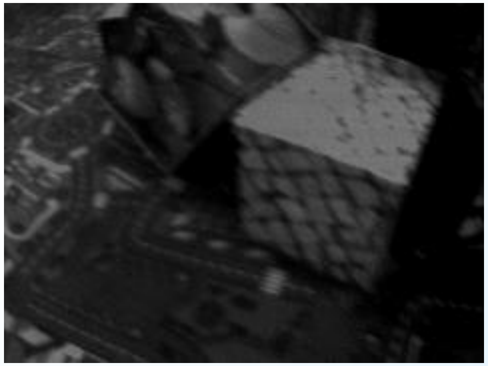
- They are not affected by under/over exposure or motion blur associated with a synchronous shutter.

Motivation

- **Low frame-rate intensity images**

  DVS $\geq 3\mu s$

  APS $\geq 5ms$



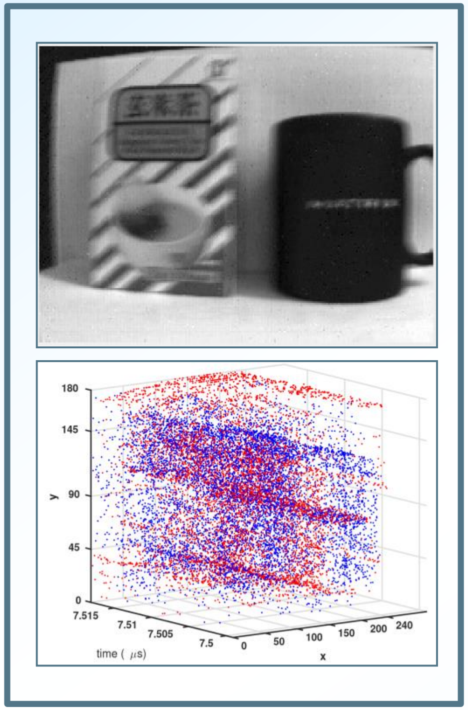
- **Inherent blurry effects**

  Reduce the exposure time – dark and noisy;

  Ignore the blurry effect and find lucky frame in a long video.

Our goal

To reconstruct a **high frame-rate, sharp video** from a single blurry frame and its event data.



Input

Output

Event camera model

The camera outputs a sequence of events *e*, denoted by (x, y, t, σ)

$$e_{xy}(t) = \sigma\delta(t)$$

(x, y) Image coordinates

t The time that the event takes place

σ $\sigma = \pm 1$ is the polarity, denotes the direction (increase or decrease) of the intensity

$\delta(t)$ an impulse function, with unit integral at time t

Event camera Model

$$\sigma = \mathrm{T}\left(\log\left(\frac{\mathrm{L}(t+1)}{\mathrm{L}(t)}\right),\ c\right)$$

c is a **threshold parameter** determining whether an event should be recorded or not;

$\mathbf{L}(t)$ is the latent image at time t

$\mathbf{L}(t+1)$ is the latent image at time $t+1$

When an event is triggered, the intensity at that pixel is updated to a new level.

Event camera Model

$\mathrm{T}\left(\cdot,\cdot\right)$ is a truncation function

$$\mathrm{T}\left(d,c\right)=\begin{cases}+1, & d\geq c\\ 0, & d\in(-c,c)\\ -1, & d\leq -c\end{cases}$$

If no blurry image

Suppose the reference timestamp is f,
$\mathbf{E}(t)$ is the sum of events between time f and t at a given pixel

A tricky way to solve

When the input image is blur, a trivial solution would be:

- Deblur the image with existing deblurring methods
- Reconstruct the video using the first integral model.

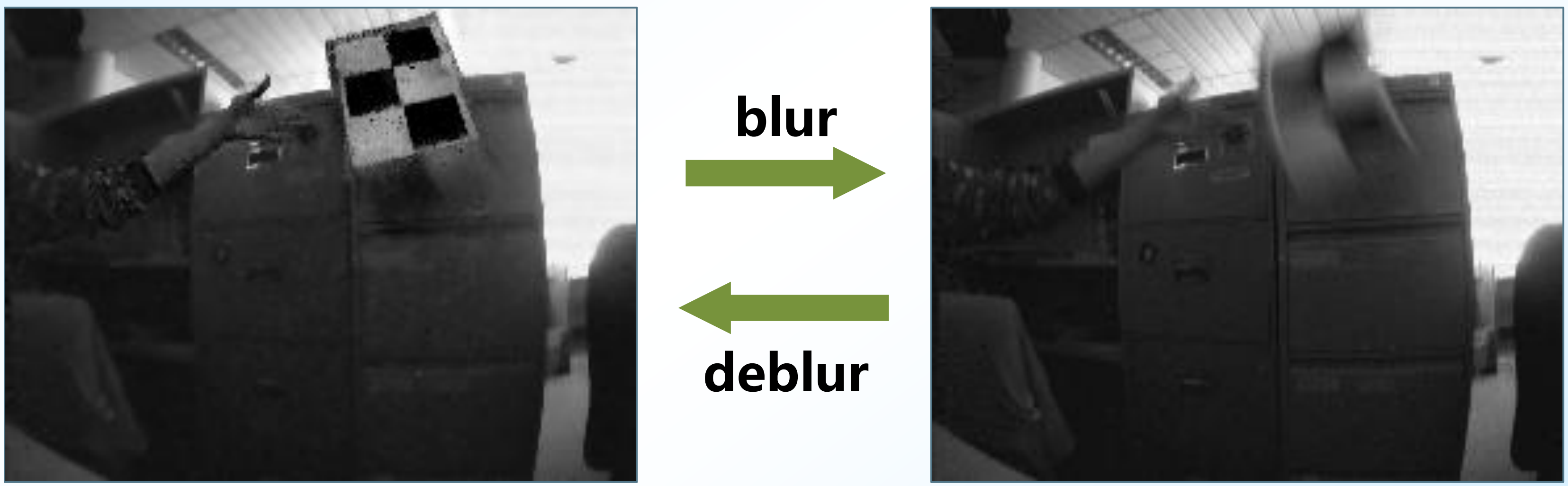
(a) Blur image

(b) Deblur result by [1]

(c) First integral from (b)

The motion blur model

$$\mathbf{B} = \frac{1}{T}\int_{f-T/2}^{f+T/2} \mathbf{L}(t)dt$$



$\mathbf{B}$ is a blurry image, equal to the average of the latent images $\mathbf{L}(t)$ during the exposure time $[f - T/2, f + T/2]$

Is there a connection between them?

- A blurry image can be regarded as the integral of a sequence of latent images
- The events indicate the changes between the latent images.

We are able to model the blur-generation process by associating event data to a latent image.

Our Event based Double Integral Model

Suppose the reference timestamp is f, and $\tilde{\mathbf{L}} = \log(\mathbf{L})$, $\mathbf{E}(t)$ is the sum of events between time f and t at a given pixel

$$\mathbf{E}(t) = \int_f^t e(s)ds$$

$$\mathbf{L}(t) = \mathbf{L}(f)\exp(c\mathbf{E}(t))$$

First integral

$$\mathbf{B} = \frac{1}{T}\int_{f-T/2}^{f+T/2} \mathbf{L}(t)dt$$

$$\mathbf{B} = \frac{\mathbf{L}(f)}{T}\int_{f-T/2}^{f+T/2} \exp\left(c\int_f^t e(s)ds\right)dt$$

Second integral

$$\tilde{\mathbf{L}}(t) = \tilde{\mathbf{L}}(f) + c\mathbf{E}(t) \qquad \tilde{\mathbf{L}}(f) = \tilde{\mathbf{B}} - \log\left(\frac{1}{T}\int_{f-T/2}^{f+T/2} \exp\left(c\int_f^t e(s)ds\right)dt\right)$$

Our Event based Double Integral Model

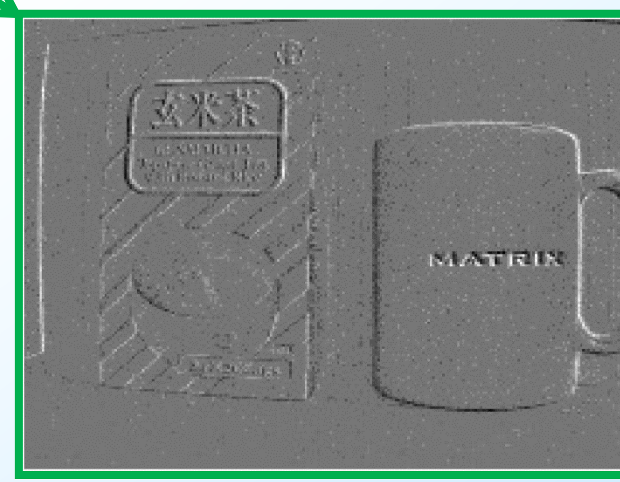
$$\tilde{\mathbf{L}}(t) = \tilde{\mathbf{L}}(f) + c\mathbf{E}(t) \qquad \tilde{\mathbf{L}}(f) = \tilde{\mathbf{B}} - \log\left(\frac{1}{T} \int_{f-T/2}^{f+T/2} \exp\left(c \int_{f}^{t} e(s) ds \right) dt \right)$$

Our Event based Double Integral Model

$$\tilde{\mathbf{L}}(t) = \tilde{\mathbf{L}}(f) + c\mathbf{E}(t)$$

$$\tilde{\mathbf{L}}(f) = \tilde{\mathbf{B}} - \log\left(\frac{1}{T}\int_{f-T/2}^{f+T/2} \exp\left(c\int_{f}^{t} e(s)ds\right)dt\right)$$

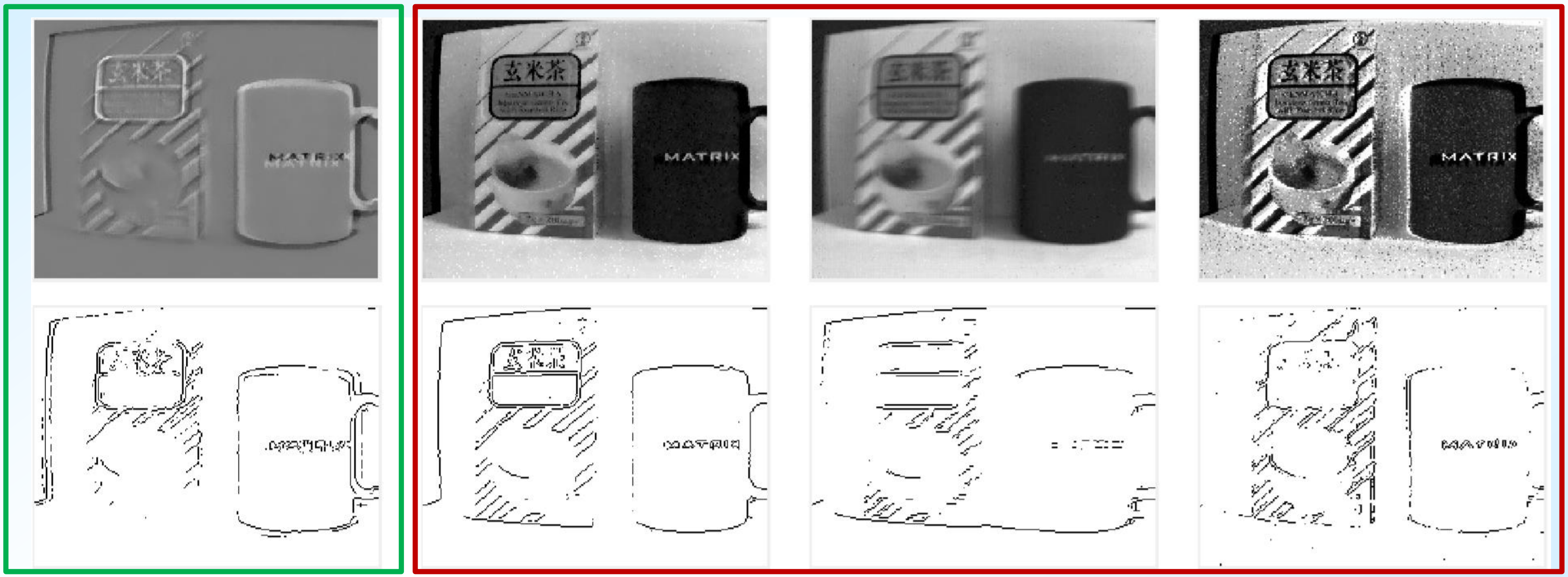
First integral



Second integral

Finding c

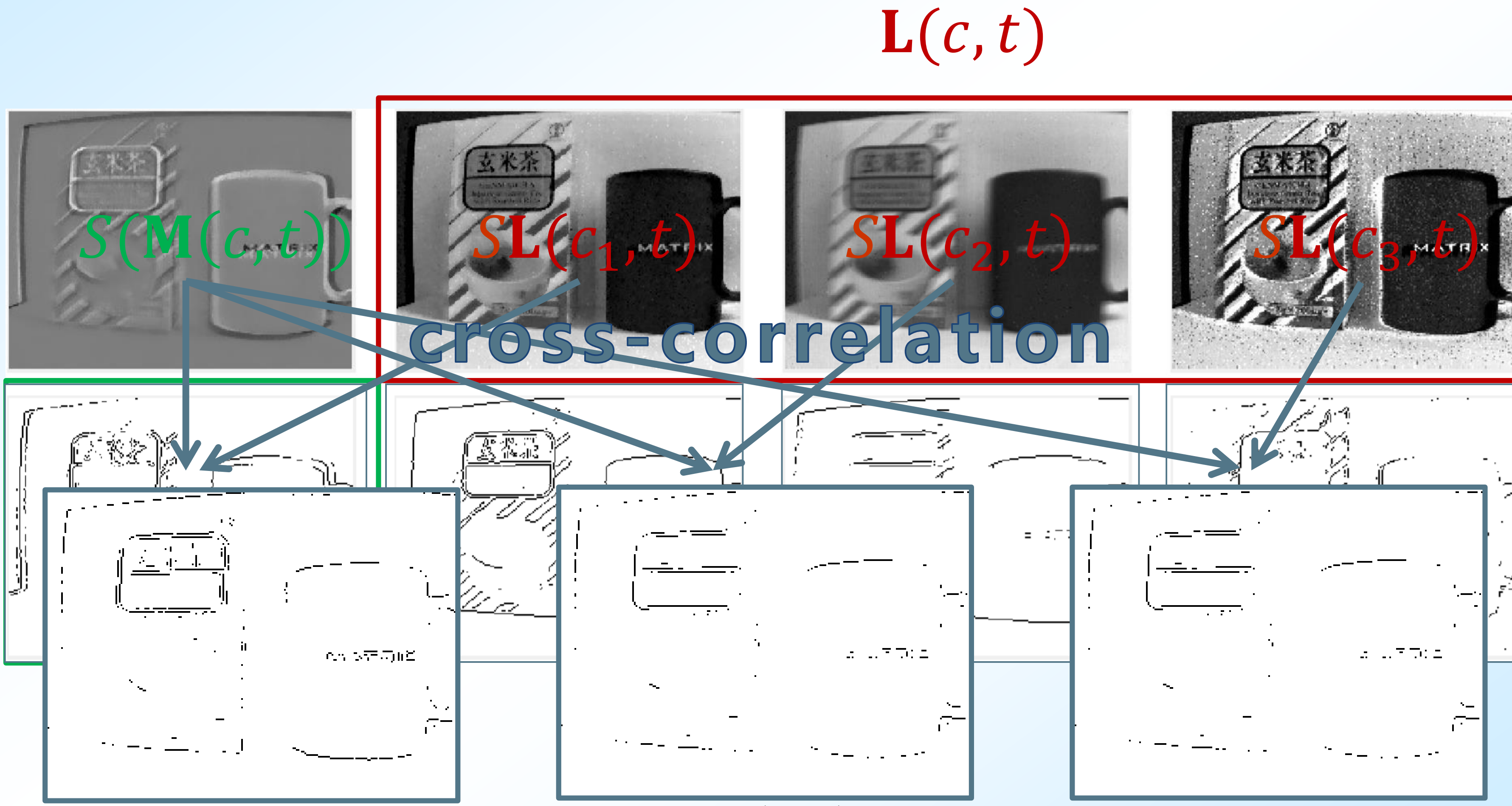
$\mathbf{M}(c,t)$ $\mathbf{L}(c_1,t)$ $\mathbf{L}(c_2,t)$ $\mathbf{L}(c_3,t)$



$S(\mathbf{M}(c,t))$ $S(\mathbf{L}(c_1,t))$ $S(\mathbf{L}(c_2,t))$ $S(\mathbf{L}(c_3,t))$

$S(\cdot)$ means Sobel filter, and $\mathbf{M}(c,t)$ is the polarity integration.

Finding c

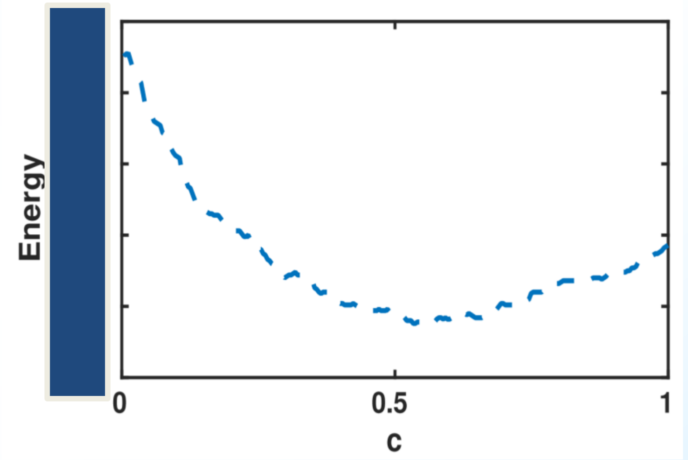


$S(\cdot)$ means Sobel filter, and $\mathbf{M}(c,t)$ is the polarity integration.

Finding *c*

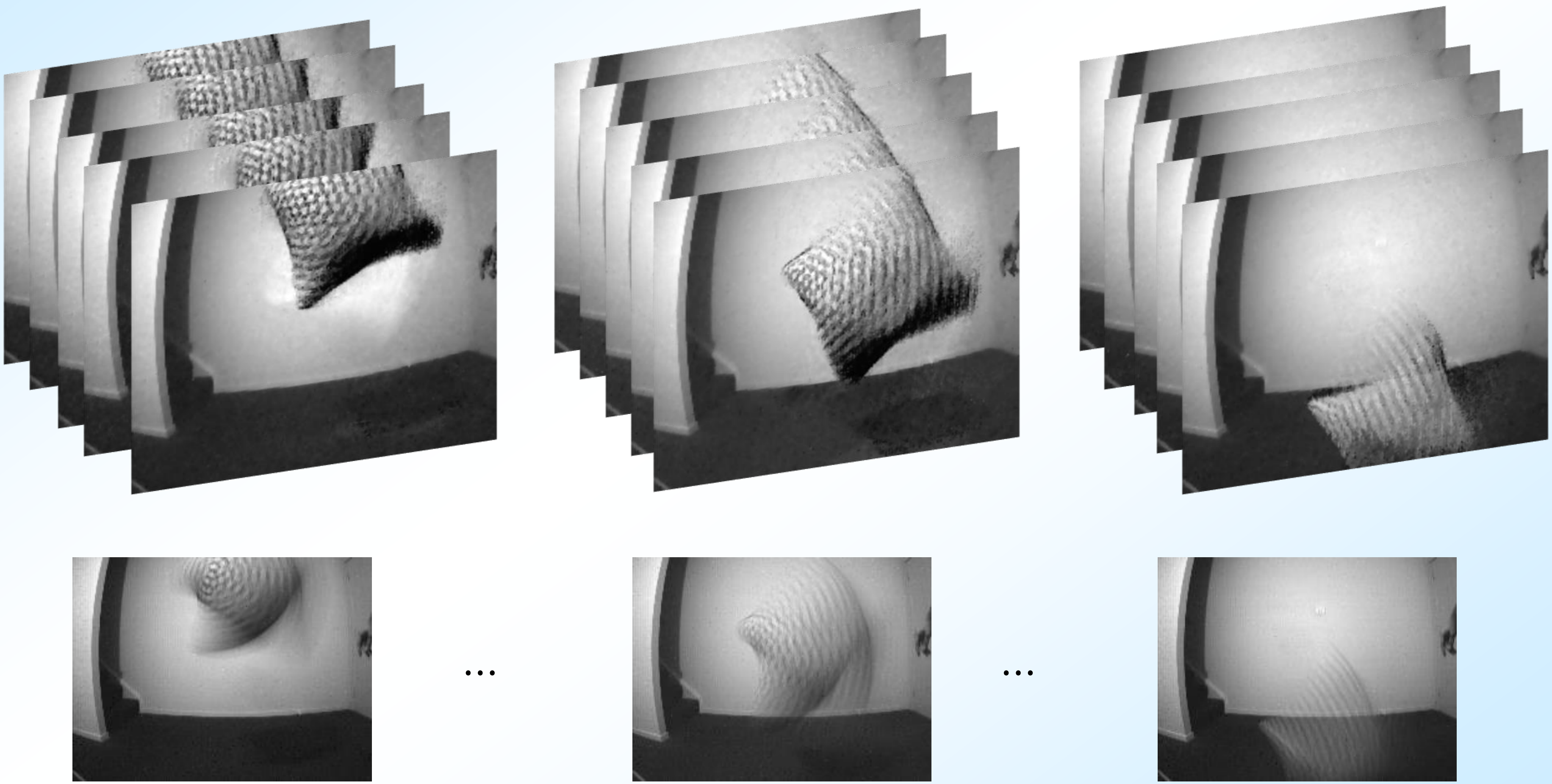
$$\min_{c} \lambda \sum S(\mathbf{L}(c,t)) \cdot S(\mathbf{M}(c,t)) + \left| \nabla \mathbf{L}(c,t) \right|_1$$

where $\lambda < 0$



Examples show that as a function of *c*, the residual error in solving the equations is not convex. However, in most cases (empirically) it seems to be convex, or at least it has a single minimum.

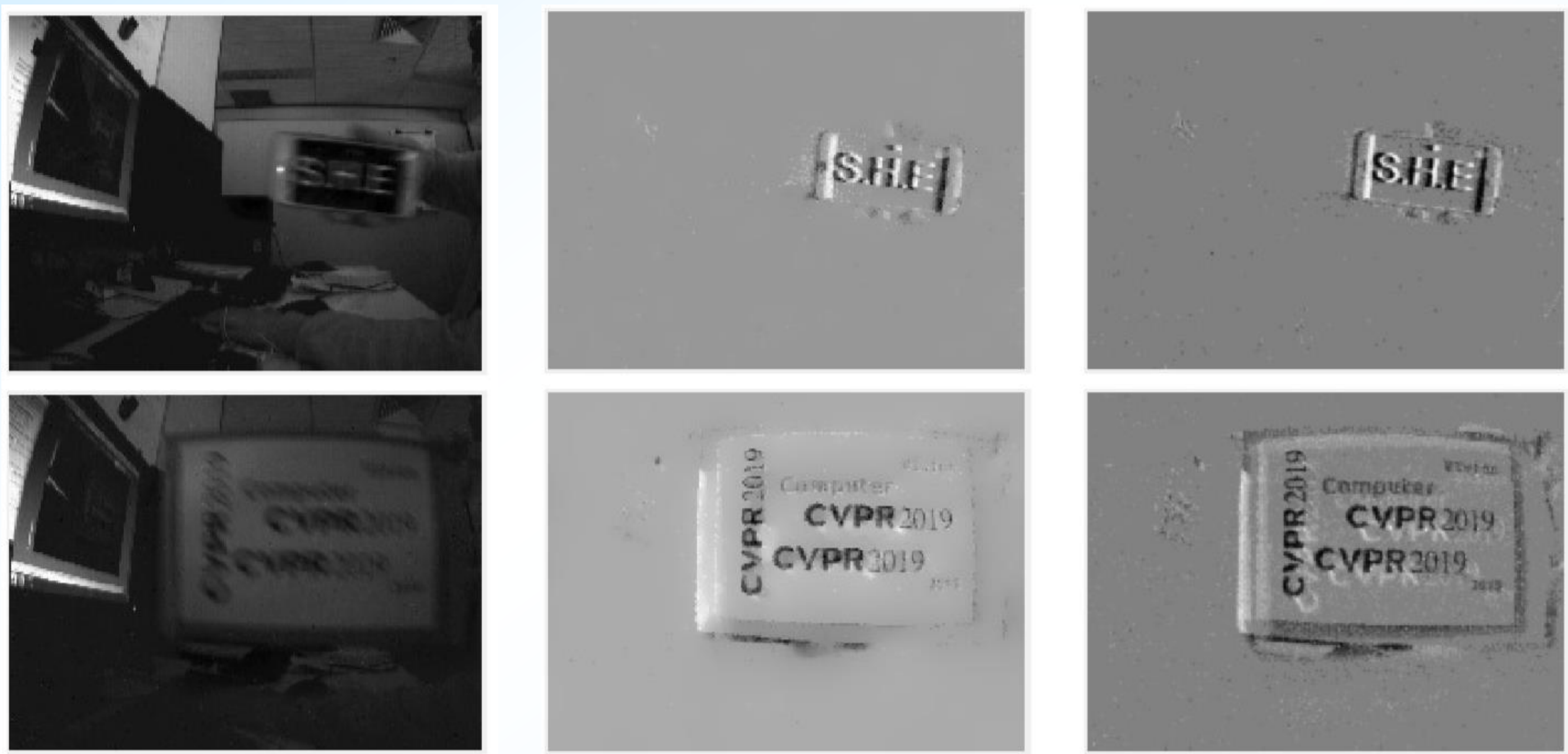
High frame rate video generation



Results-Deblurring

(a) Blur image **(b) Deblur result by [1]** **(c) Deblur result by [2]**

Results-Image reconstruction



(a) Blur image **(b) Reconstructed by [3]** **(c) Reconstructed by [4]**

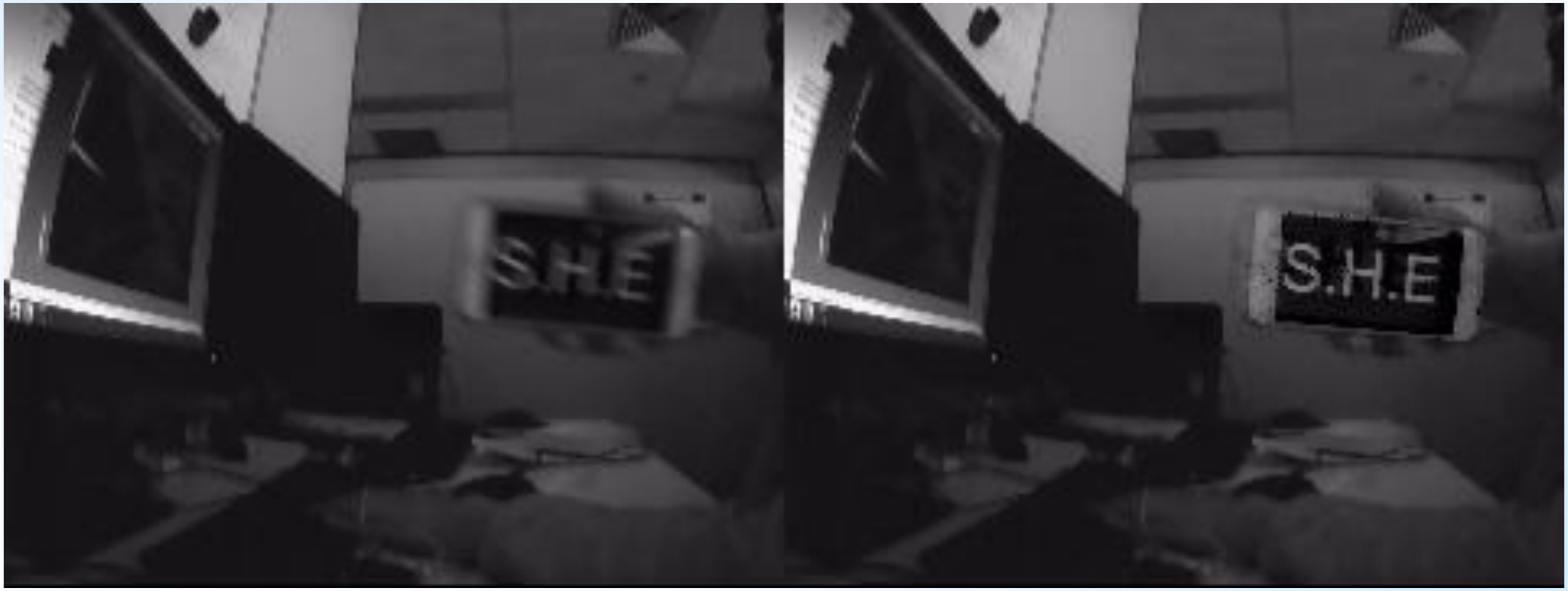
Results-Image reconstruction

(a) Blur image **(b) Reconstructed by [4]** **(c) Our**

Results

Results

Results



Extension: Using More than One Frame

Suppose a frame with N events, then the **EDI** gives N equations with N + 1 unknowns. We need to add regularization terms with unexpected weight parameters when solving the problem.

In addition, noise from events can easily degrade the quality of reconstructed videos, especially at transitions between images.

If n (n > 1) blurred images are available, then this gives sufficiently many nN equations for the unknowns, though they are not linear in c. Thus, we can solve the problem directly, in least-squares, or some method for solving an over-constrained system.

Paper: Bringing a blurry frame alive at high frame-rate with an event camera. Accepted by CVPR2019 (oral)

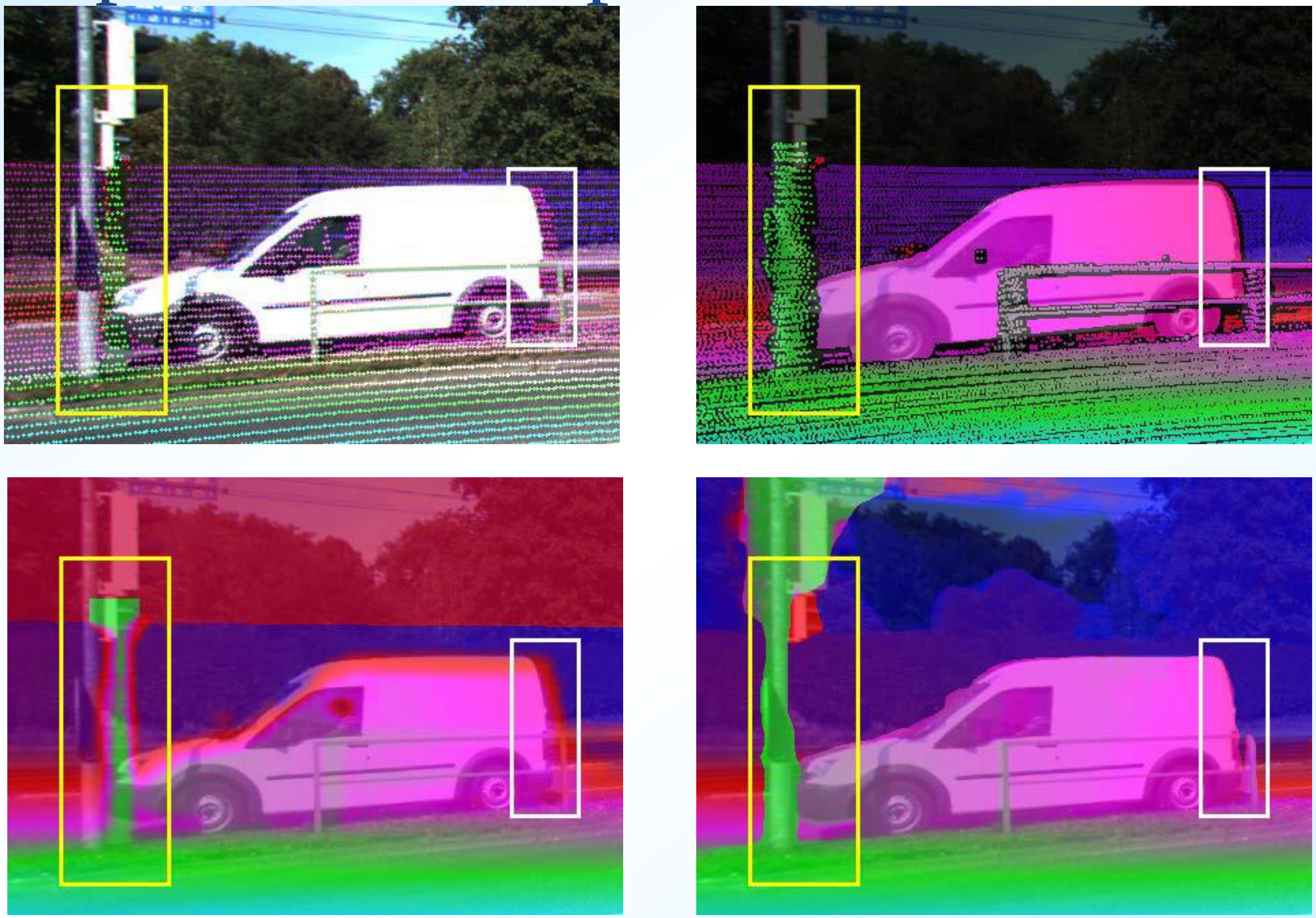
Code, data, and demo

https://github.com/panpanfei/Bringing-a-Blurry-Frame-Alive-at-High-Frame-Rate-with-an-Event-Camera

Other CVPR 19 Papers

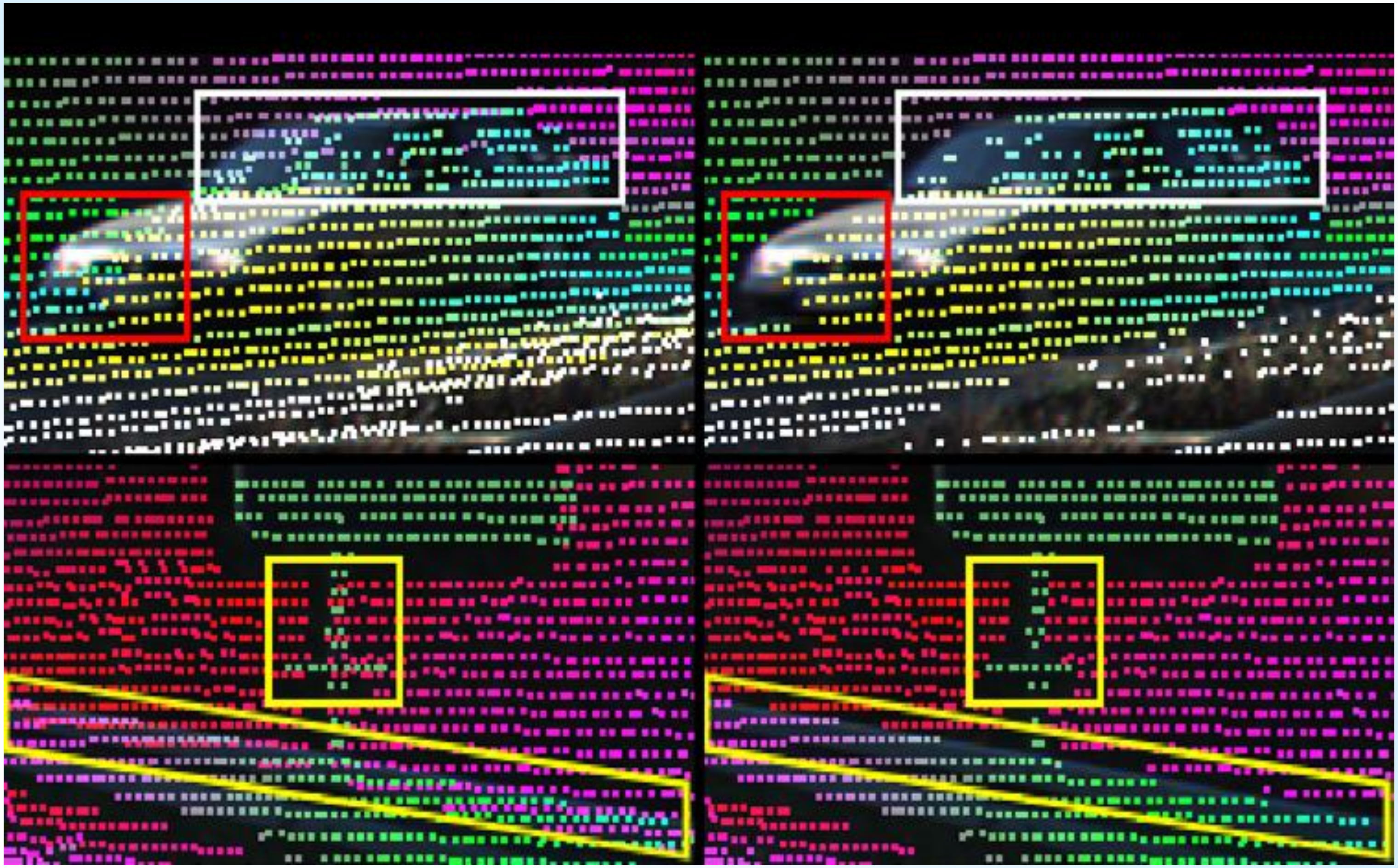
- Hongguang Zhang, **Yuchao Dai**, Hongdong Li, Piotr Koniusz. Deep Stacked Hierarchical Multi-Patch Network for Image Deblurring,.
- Xuelian Cheng, Yiran Zhong, **Yuchao Dai**, Hongdong Li. Noise-Aware Unsupervised Deep Lidar-Stereo Fusion.
- Yiran Zhong, Pan Ji, **Yuchao Dai**, Jianyuan Wang, Hongdong Li. Unsupervised Deep Epipolar Flow for Stationary or Dynamic Scenes.
- L Pan, R Hartley, M Liu, **Y Dai**. Phase-only Image Based Kernel Estimation for Single-image Blind Deblurring.
- X Song, P Wang, D Zhou, R Zhu, C Guan, **Y Dai**, H Su, H Li, R Yang. ApolloCar3D: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving..

Unsupervised Deep Lidar-Stereo Fusion

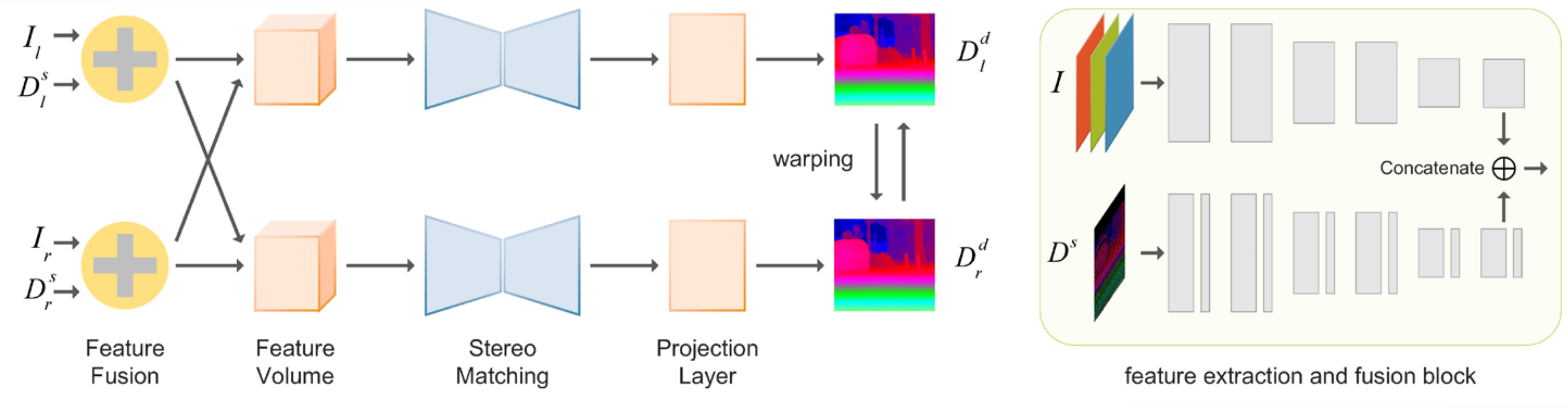


We highlight the displacement error of Lidar points with bounding boxes. Lidar points are dilated for better visualization and we overlay our disparity maps to the colour images for illustration.

Unsupervised Deep Lidar-Stereo Fusion

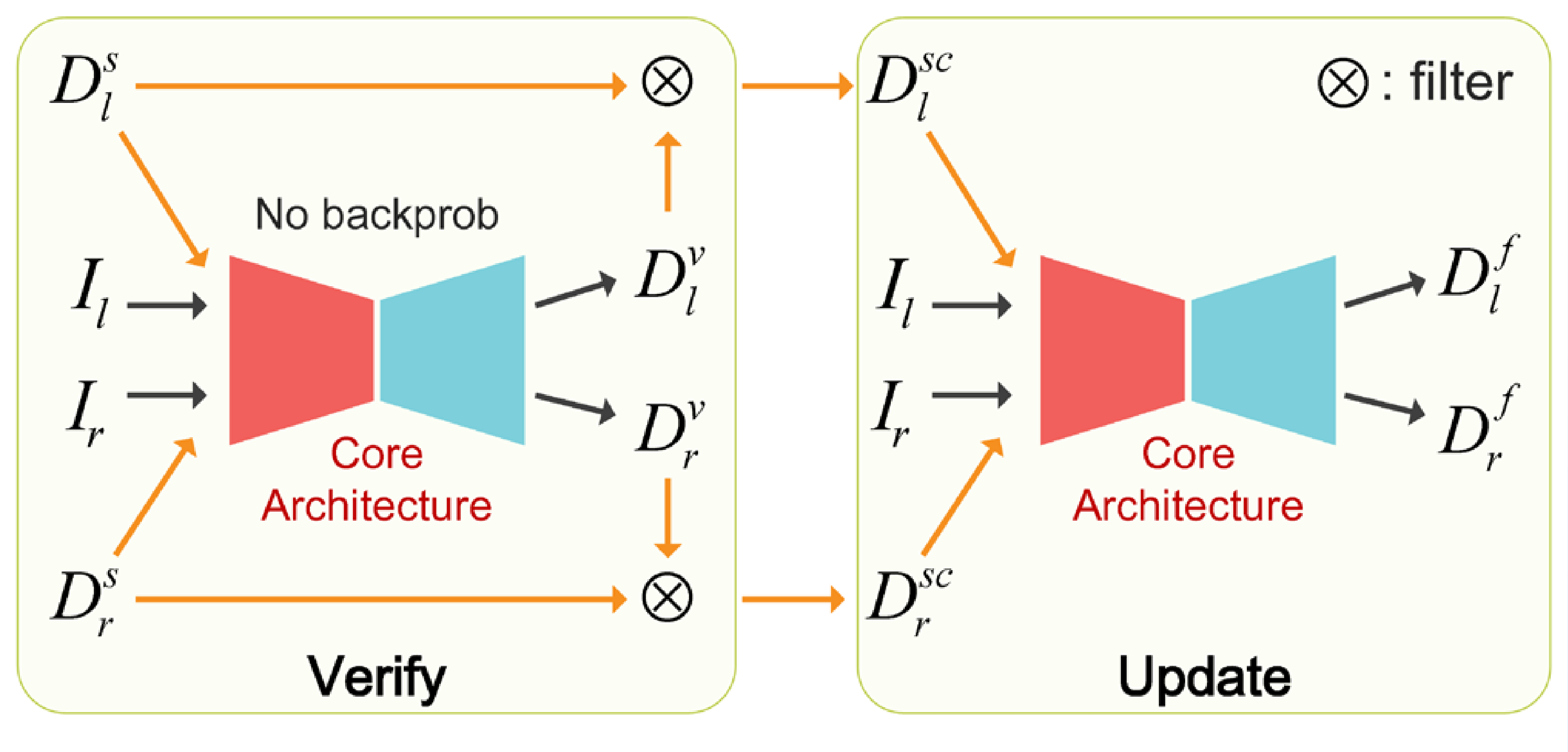


Unsupervised Deep Lidar-Stereo Fusion



Core Architecture of our LidarStereoNet. It consists of a feature extraction and fusion block, a stack-hourglass type feature matching block and a disparity computing layer.

Unsupervised Deep Lidar-Stereo Fusion



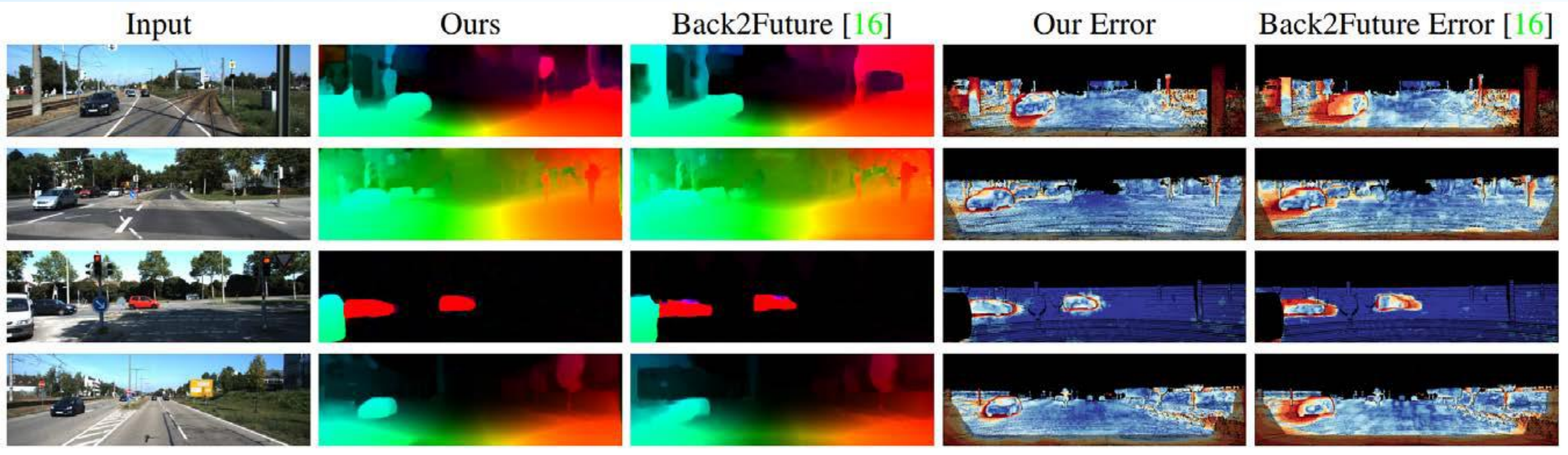
The "Feedback Loop": warping and Lidar should be consistent.

Unsupervised Deep Epipolar Flow for Stationary or Dynamic Scenes

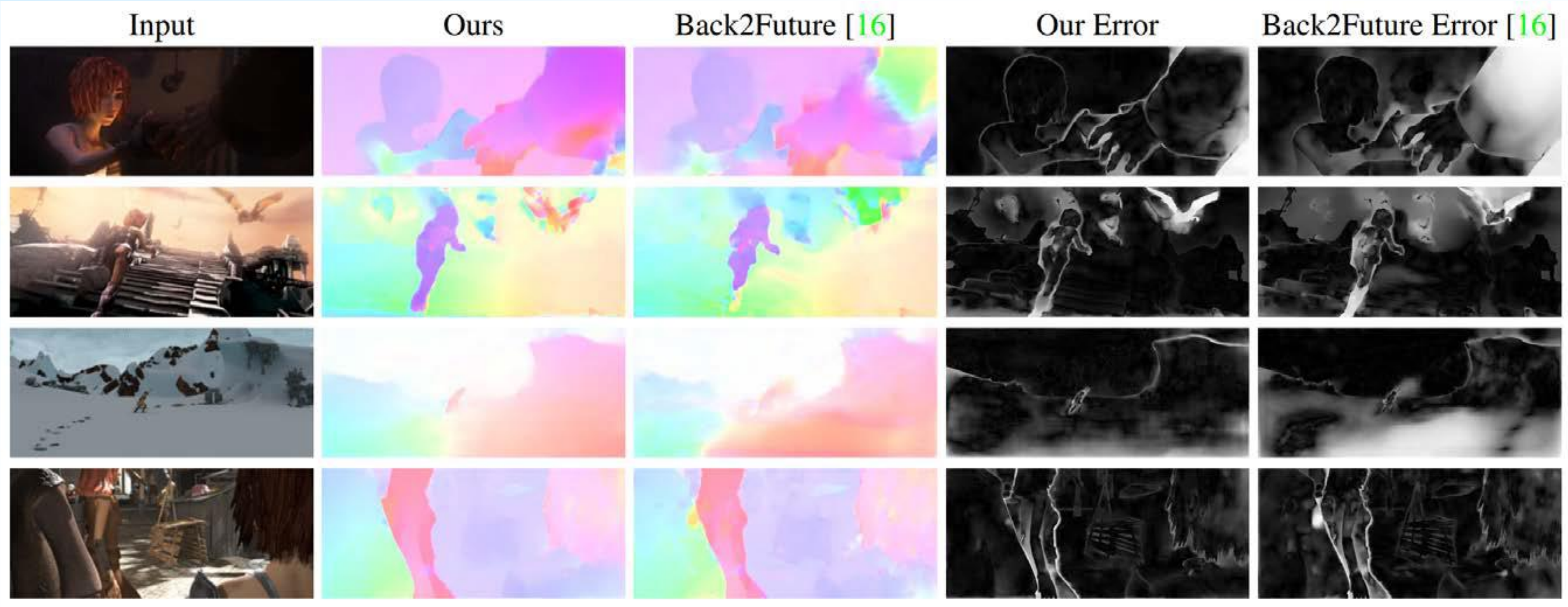
	Method	KITTI 2012				KITTI 2015			Sintel Clean		Sintel Final	
		EPE(all)		EPE(noc)		EPE(all)	EPE(noc)	Fl−all	EPE(all)		EPE(all)	
		train	*test*	*train*	*test*	*train*	*train*	*test*	*train*	*test*	*train*	*test*
Non-deep	EpicFlow [29]	3.47	3.8	–	1.5	9.27	–	26.29%	2.27	4.11	3.56	6.29
	MRFlow [38]	–	–	–	–	–	–	12.19%	(1.83)	2.53	(3.59)	5.38
Supervised	SpyNet-ft [25]	(4.13)	4.1	–	2.0	–	–	35.07%	(3.17)	6.64	(4.32)	8.36
	FlowNet2-ft [8]	(1.28)	1.8	–	1.0	2.30	–	10.41%	(1.45)	4.16	(2.01)	5.74
	PWC-Net [33]	4.14	–	–	–	10.35	–	–	2.55	–	3.93	–
	PWC-Net-ft [33]	(1.45)	1.7	–	0.9	(2.16)	–	9.60%	(1.70)	3.86	(2.21)	5.17
Unsupervised	UnsupFlownet [42]	(11.30)	9.9	(4.30)	4.6	–	–	–	–	–	–	–
	DSTFlow-ft [28]	(10.43)	12.4	(3.29)	4.0	(16.79)	(6.96)	39.00%	(6.16)	10.41	(7.38)	11.28
	DF-Net-ft [44]	(3.54)	4.4	–	–	(8.98)	–	25.70%	–	–	–	–
	GeoNet [41]	–	–	–	–	10.81	8.05	–	–	–	–	–
	UnFlow [31]	(3.29)	–	(1.26)	–	(8.10)	–	–	–	9.38	7.91	10.21
	OAFlow-ft [35]	(3.55)	4.2	–	–	(8.88)	–	31.20%	(4.03)	7.95	(5.95)	9.15
	CCFlow [26]	–	–	–	–	(5.66)	–	25.27%	–	–	–	–
	Back2Future-ft [16]	–	–	–	–	(6.59)	(3.22)	22.94%	(3.89)	7.23	(5.52)	8.81
	Our-baseline	3.23	–	1.04	–	7.93	4.21	–	6.72	–	7.31	–
	Our-gtF	2.61	–	1.04	–	6.03	2.89	–	6.15	–	6.71	–
	Our-F	2.56	–	**0.97**	–	6.42	3.09	–	6.21	–	6.73	–
	Our-low-rank	2.63	–	1.07	–	5.91	3.03	–	6.39	–	6.96	–
	Our-sub	2.62	–	1.03	–	6.02	2.98	–	6.15	–	6.83	–
	Our-sub-test-ft	2.61	**(3.2)**	1.03	**(1.1)**	5.56	2.56	**(16.24%)**	3.94	**(6.84)**	5.08	**(8.33)**
	Our-sub-train-ft	**(2.51)**	3.4	(0.99)	1.3	**(5.55)**	**(2.46)**	16.95%	**(3.54)**	7.00	**(4.99)**	8.51

Table 1. **Performance comparison on the KITTI and Sintel optical flow benchmarks.** The metric EPE(noc) indicates the average endpoint error of non-occluded regions while the term EPE(all) is that for all pixels. The KITTI 2015 testing dataset evaluates results by the percentage of flow outliers (Fl). The baseline, gtF, F, low-rank, and sub models were trained on the KITTI VO dataset. The parentheses indicate the corresponding models that were trained on the same data and the missing entries (-) indicate the results were not reported. Note that the current STOA unsupervised method Back2Future Flow [16] uses three frames as input. Best results are marked by bold fonts.

Unsupervised Deep Epipolar Flow for Stationary or Dynamic Scenes



Unsupervised Deep Epipolar Flow for Stationary or Dynamic Scenes



Thanks！