Asynchronous Convolutions and Image Reconstruction

Robert Mahony

Second International Workshop on Event-based Vision and Smart Cameras
17 June 2019

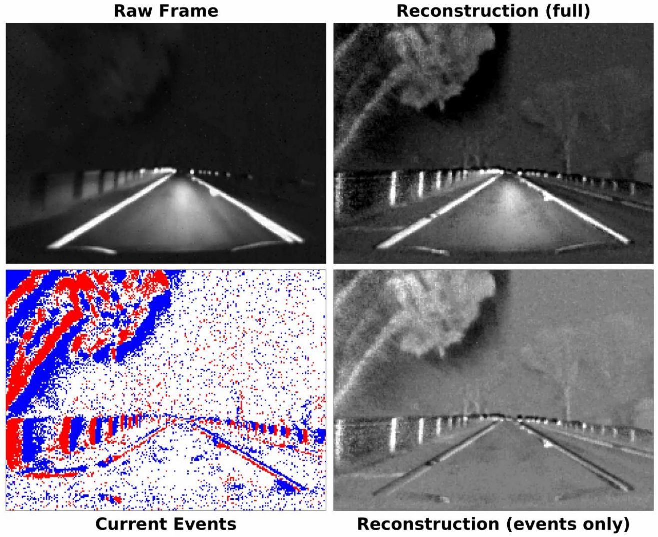
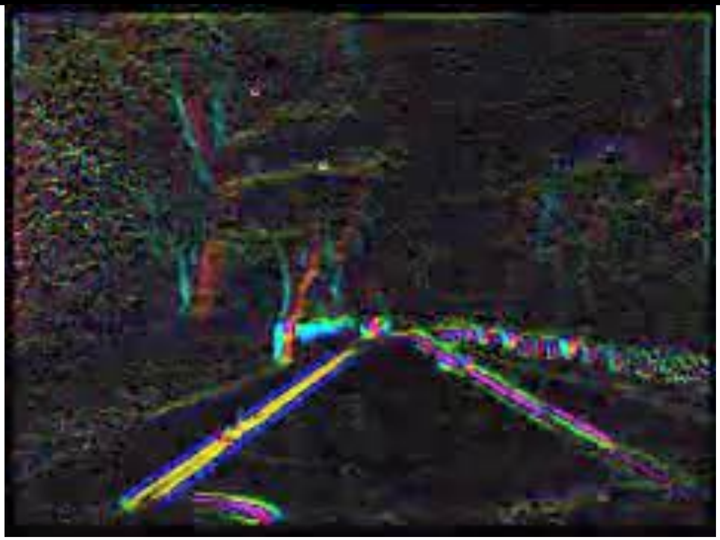
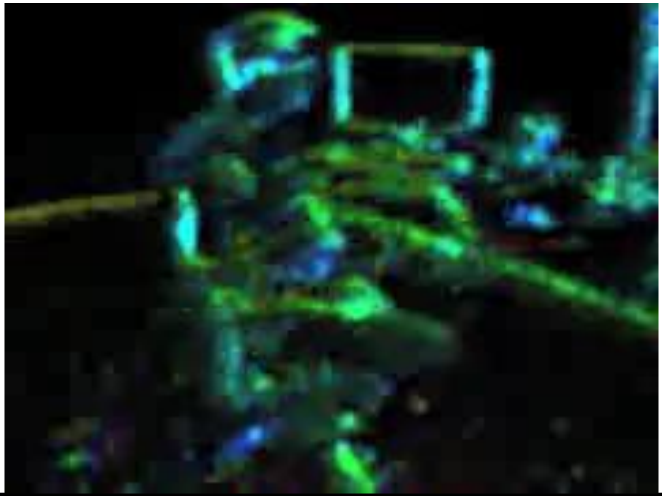


Image reconstruction

Image Gradient

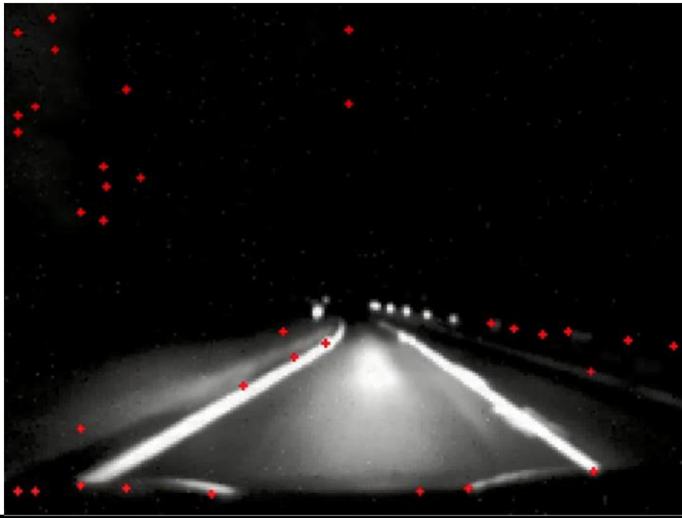


https://github.com/cedric-scheerlinck/dvs_image_reconstruction



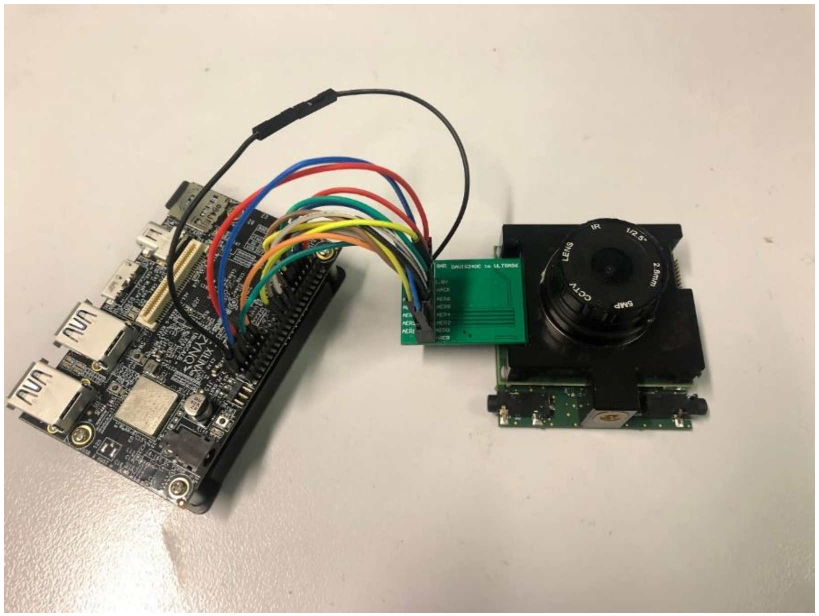
Optic Flow

Feature Tracking

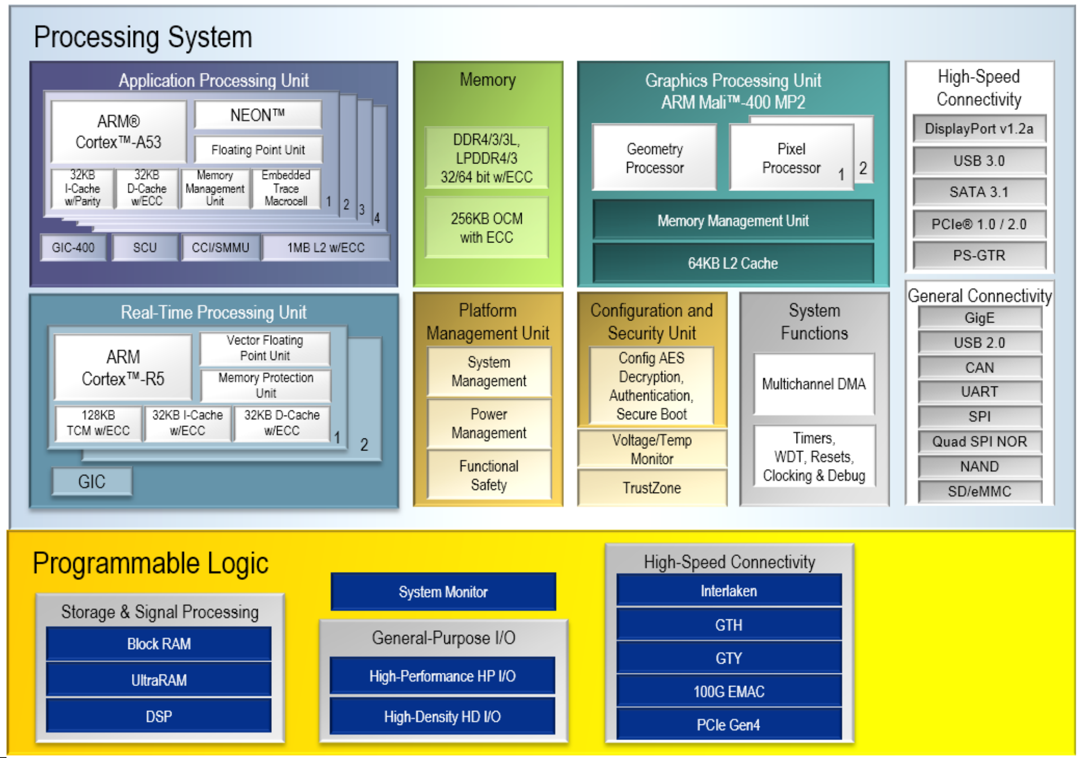


Embedded implementation

DAVIS240C
- Direct access to events through parallel bus.

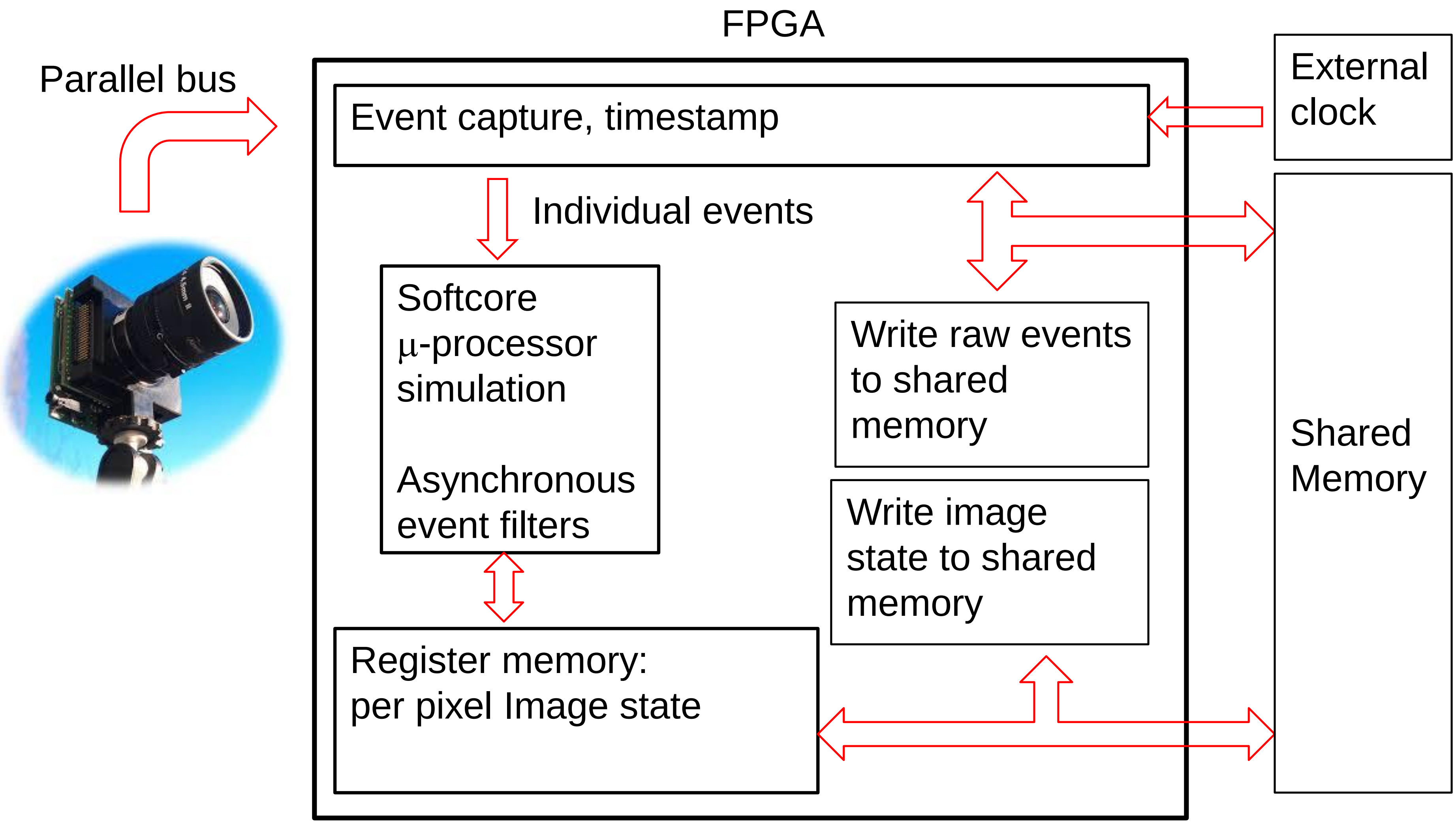


Feature	ULTRA96
Processor	1.5 GHz quad-core Cortex-A53
Memory	2G LPDDR4
PL	192,000 logic slices
PL Clock	<300MHz



ULTRA96
- Read events direct to FPGA.
- Write events and derived information to shared memory.

FPGA architecture

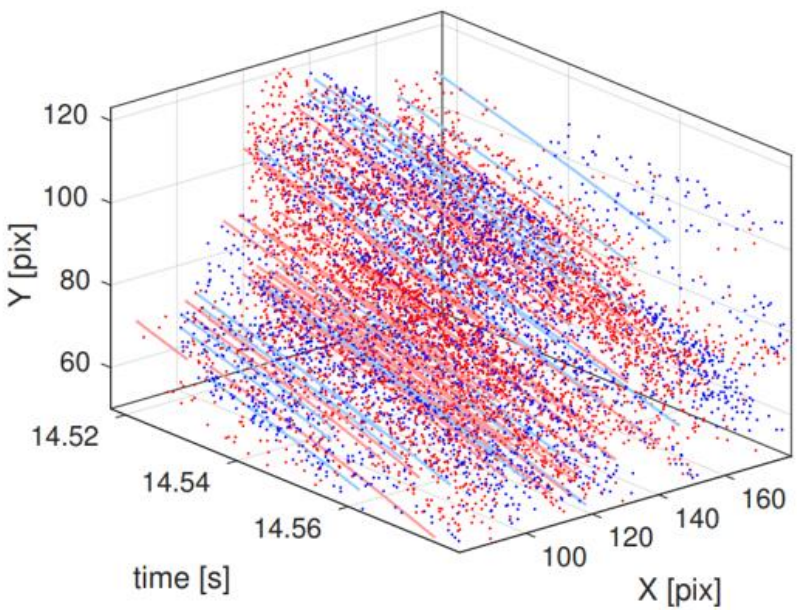


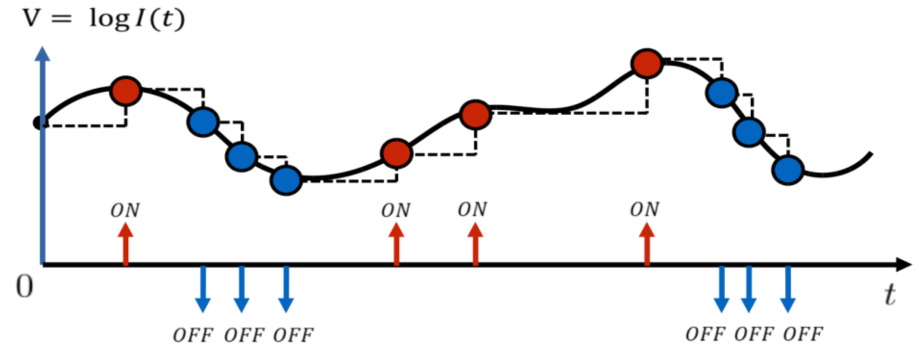
Output of an event camera

An event camera yields a series of events $\{e_k\}$

$$e_k = (\sigma_k, t_k, u_k, v_k)$$

where $(\sigma_k, t_k, u_k, v_k)$ are the polarity, time stamp and pixel location of event k.





What is an event

Define a function $E_k : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ by

$$\boxed{E_k(u,v) := \sigma_k \delta_{(u_k,v_k)}(u,v)}$$

where $\delta_{(u_k,v_k)}$ is the Kroneker delta function

$$\delta_{(u_k,v_k)}(u,v) = \begin{cases} 1 & (u,v) = (u_k,v_k) \\ 0 & (u,v) \neq (u_k.v_k) \end{cases}$$

Event stream $$\boxed{E(t,u,v) := \sum_{k=1}^{\infty} E_k(u,v)\delta(t-t_k)}$$

where δ is the Dirac delta function

$$\int_{-\infty}^{\infty} \delta(t-\tau) f(\tau) \mathrm{d}\tau = f(t)$$

The time coordinate *t* is a continuous-time variable

Image reconstruction from integration

Direct integration

$$\hat{L}(t;u,v) = \sum_{\{k \mid t_k \le t\}} E_k(u,v) = \int_{-\infty}^{t} E(\tau,u,v)\mathrm{d}\tau$$

$$E_{(u,v)}(t) \longrightarrow \boxed{\frac{1}{s}} \longrightarrow \hat{L}_{(u,v)}(t)$$

Transfer function interpretation of direct integration

High levels of noise in the event stream stay in the image stream and make direct integration impractical.

High pass the state estimate filter

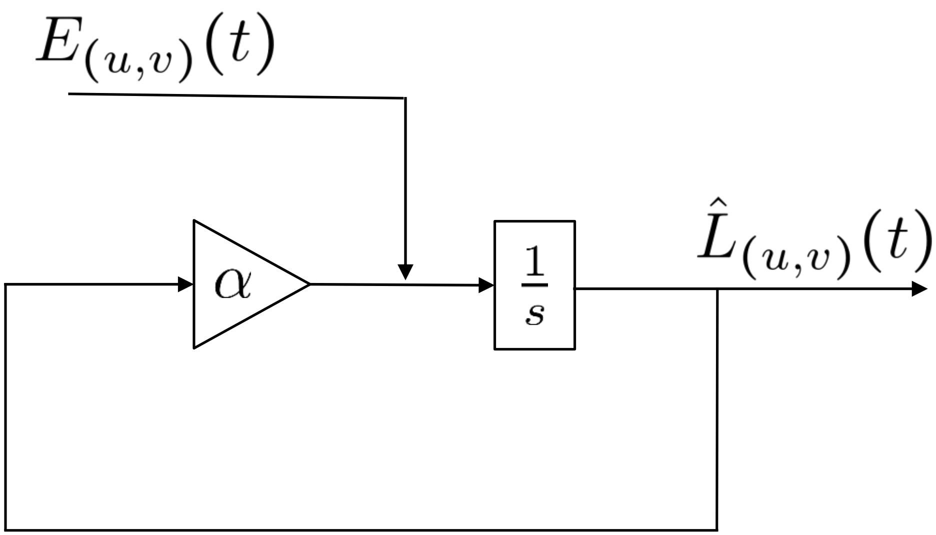
$$E_{(u,v)}(t) \longrightarrow \boxed{\frac{1}{s}} \longrightarrow \boxed{\frac{s}{s+\alpha}} \longrightarrow \hat{L}_{(u,v)}(t)$$

Integration High pass filter

Integration without high pass

Integration with high pass

High-pass filter realisation



Transfer function realisation

$$\hat{L}_{(u,v)}(s) = \frac{s}{s+\alpha}\frac{1}{s}E_{(u,v)}(s)$$

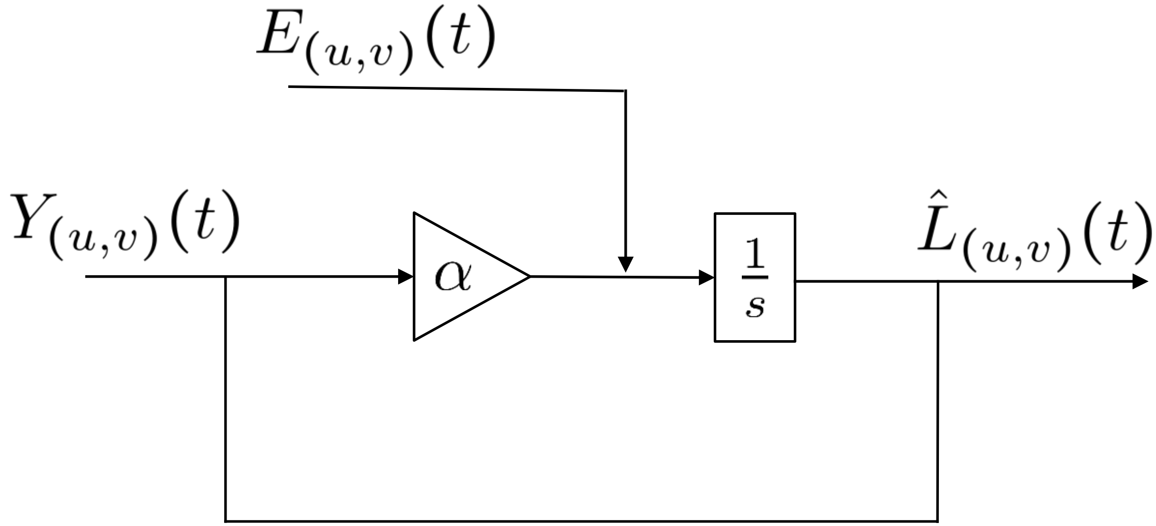
ODE system realisation

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{L}_{(u,v)}(t) = -\alpha\hat{L}_{(u,v)}(t) + E_{(u,v)}(t)$$

Image state:
$\hat{L}_{(u,v)}(t)$ is the internal state of the filter.

Complementary filter

If an estimate $Y_k(u,v)$ of the conventional image is available



Transfer function realisation

$$\hat{L}_{(u,v)}(s) = \frac{s}{s+\alpha}\frac{1}{s}E_{(u,v)}(s) + \frac{\alpha}{s+\alpha}Y_{(u,v)}(s)$$

ODE system realisation

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{L}_{(u,v)}(t) = -\alpha(\hat{L}_{(u,v)}(t) - Y_{(u,v)}(t)) + E_{(u,v)}(t)$$

If

$$\begin{aligned}\int_{-\infty}^{t_k} E(\tau,u,v)\mathrm{d}\tau &\approx L(t_k,u,v) \\ &\approx Y_k(u,v)\end{aligned}$$

$$\begin{aligned}\hat{L}(s,u,v) &\approx \left(\frac{s}{s+\alpha} + \frac{\alpha}{s+\alpha}\right)L(t,u,v) \\ &\approx L(t,u,v)\end{aligned}$$

Asynchronous implementation

Ordinary differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{L}_{(u,v)}(t) = -\alpha(\hat{L}_{(u,v)}(t) - Y_{(u,v)}(t)) + E_{(u,v)}(t)$$

Solve on the time period $t \in (t_k, t_{k+1})$ for $\hat{L}_{(u,v)}(t_k)$ known.
For $Y_{(u,v)}(t)$ constant (zero-order-hold) and $E_{(u,v)}(t) \equiv 0$ then

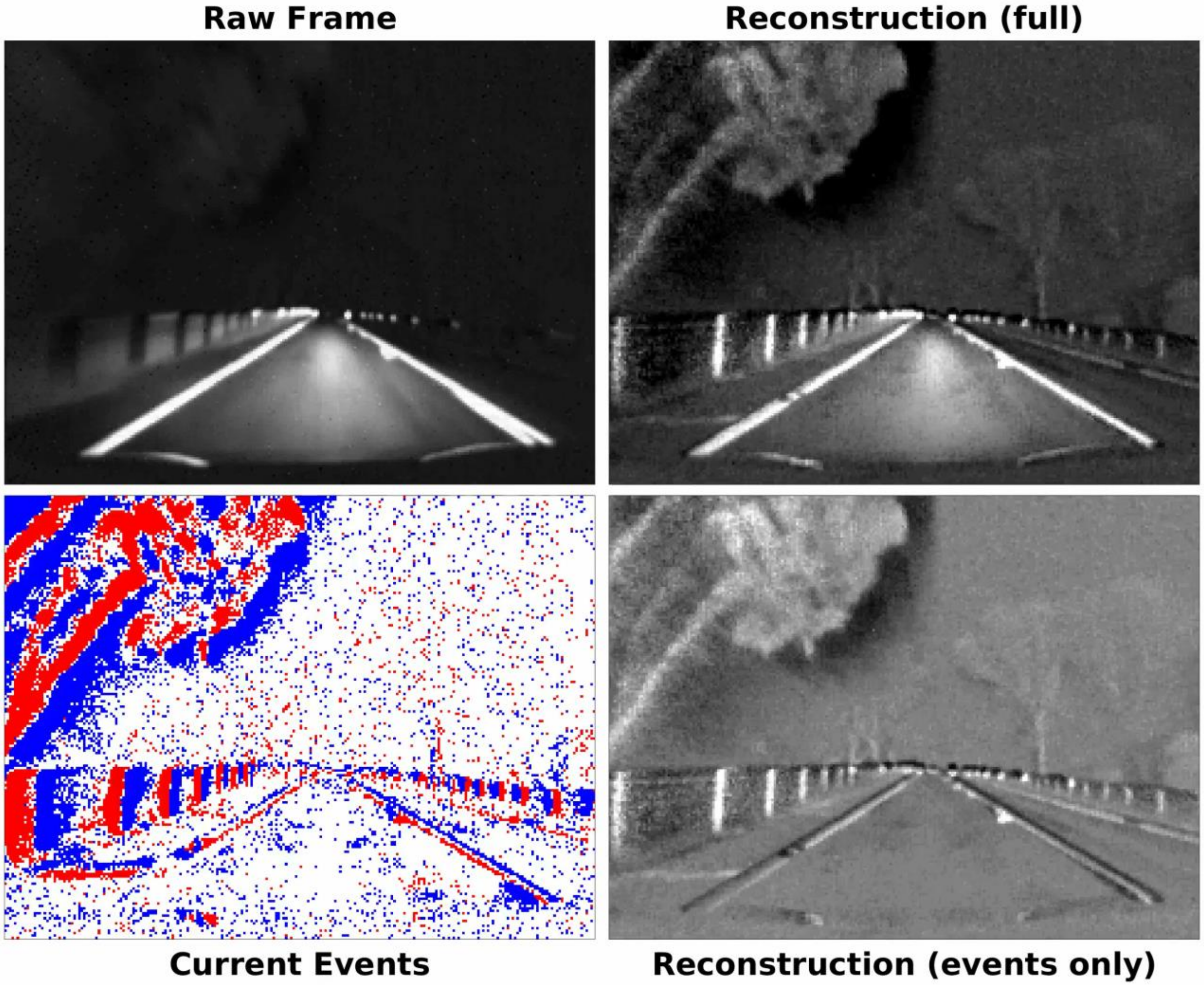
A
$$\boxed{\hat{L}_{(u,v)}(t_{k+1}) = e^{-\alpha(t_{k+1}-t_k)}\hat{L}_{(u,v)}(t_k) + (1 - e^{-\alpha(t_{k+1}-t_k)})(\hat{L}_{(u,v)}(t_k) - Y_{(u,v)}(t_k))}$$

Solve on the time period $t \in [t_{k+1}, t_{k+1}]$ for $\hat{L}_{(u,v)}(t^-_{k+1})$ known.
Integrate through the Dirac delta function

B
$$\boxed{\hat{L}_{(u,v)}(t^+_{k+1}) = \hat{L}_{(u,v)}(t^-_{k+1}) + \sigma_k \delta_{(u_k,v_k)}(u,v)}$$

- Asynchronous: Only compute when an event arrives.
- Computationally efficient: One scalar exponential.
- Image state: Estimate $\hat{L}(t,u,v)$ is stored in memory and can be accessed whenever required.

Image reconstruction

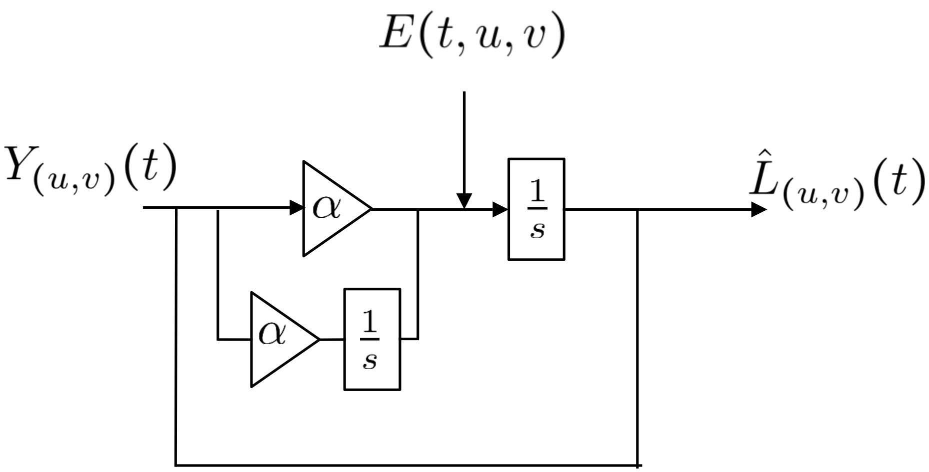


Using the DAVIS 240C raw images as $Y_{(u,v)}(t)$ ground truth.

Using events only. Setting $Y_{(u,v)}(t) = 0$.

https://github.com/cedric-scheerlinck/dvs_image_reconstruction

Bias and hot pixels



Hot pixels, that tend to fire all the time, or bias that causes a pixel to fire more in one direction than another.

Add an integrator in the "controller' gain block to make the system type I.

No bias correction

With bias correction

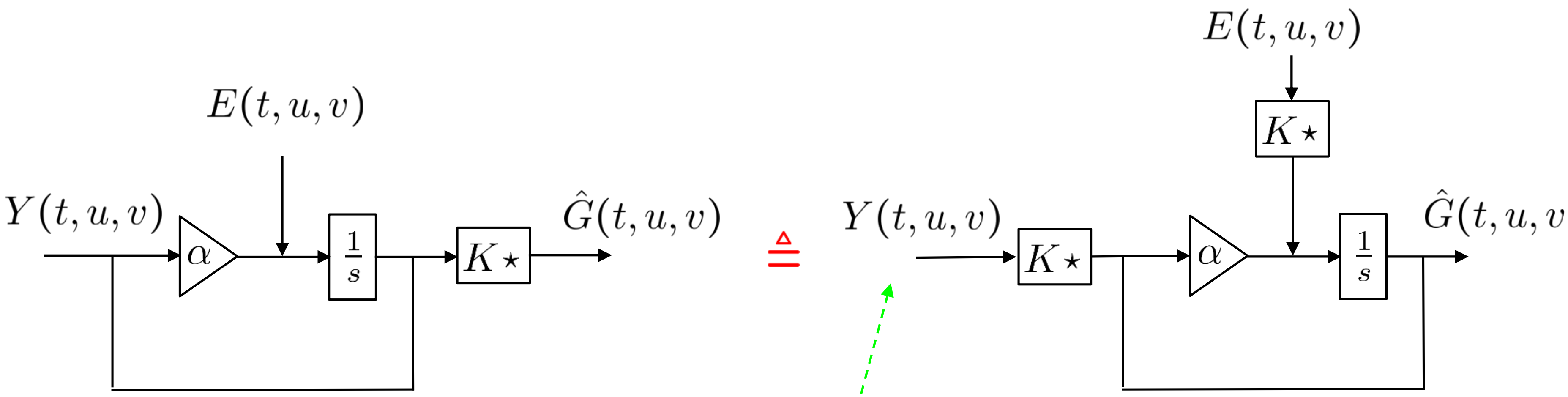
Bias image state

Image gradients

Image gradients computed from a spatial filter applied to image

$$\hat{G}_t(u,v) = K \star \hat{L}_t(u,v)$$

Spatial convolution is linear and factors through the linear filter blocks



Set the low-frequency information $K \star Y_t(u,v) \approx 0$ to zero

Convolving the event stream

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} *$$

Gradient kernel

0	0	0	0	0	0
0	0	0	0	0	0
0	0	-1	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

Event occurs at a location (u_k,v_k)

=

0	0	0	0	0	0
0	1	0	-1	0	0
0	2	0	-2	0	0
0	1	0	-1	0	0
0	0	0	0	0	0
0	0	0	0	0	0

A new set of events, all with timestamps t_k, occurring at locations given by K convolved with the Kroneker delta

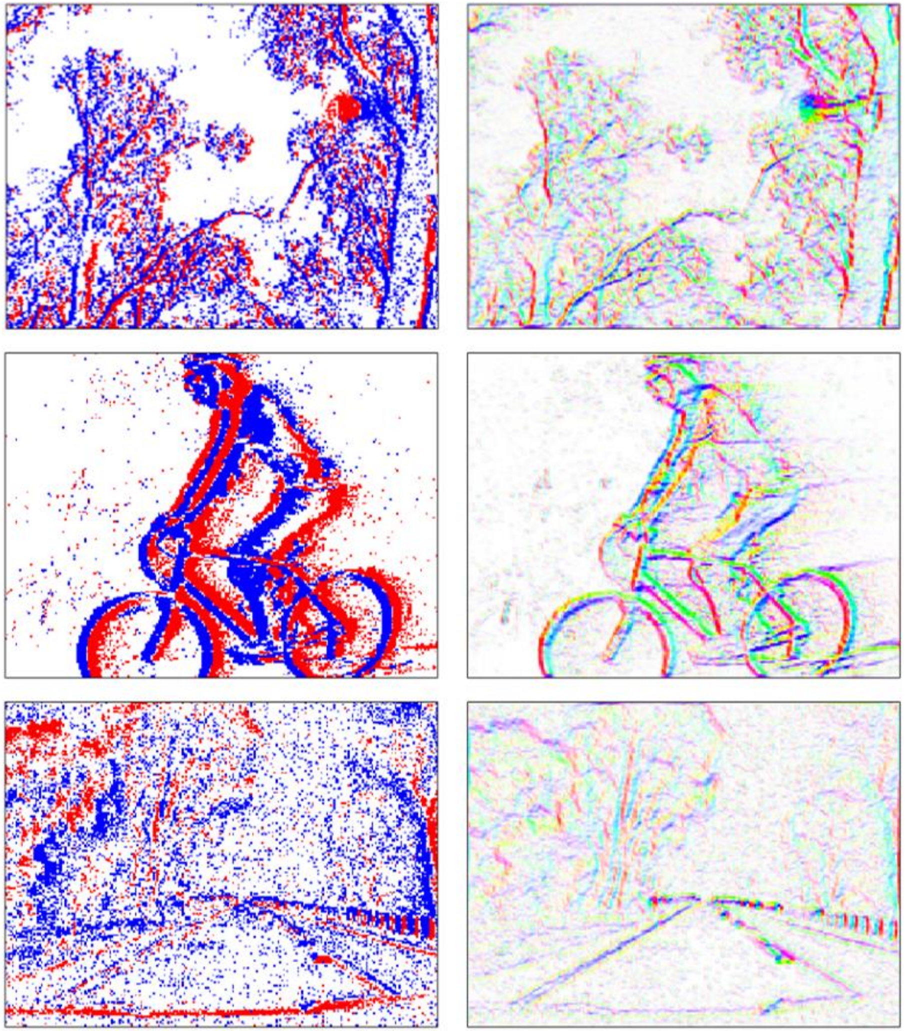
$$E_k \Rightarrow K \star E_k = \{K_{ij}\sigma_k \delta_{(u_k-i,v_k-j)}(u,v)\delta(t-t_k)\}$$

Each event

#(K) events

Image Gradient results

Current events   Gradient estimate



Sobel

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Optical flow

Optical flow $\Phi(t,u,v) = (\Phi_x(t,u,v), \Phi_y(t,u,v))$ can be thought of as an image state.

The constant brightness assumption yields

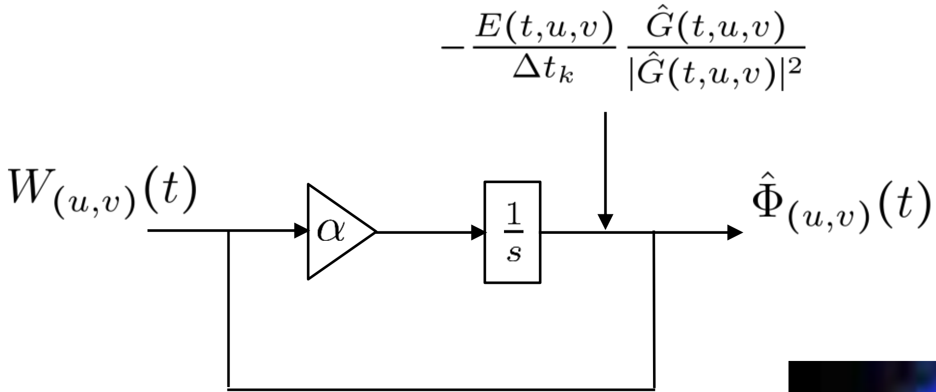
$$\frac{\partial}{\partial t}\hat{L}(t,u,v) = -\nabla \hat{L}(t,u,v)^\top \Phi(t,u,v)$$

Approximating $\frac{\partial}{\partial t}\hat{L}(t,u,v) \approx E(t,u,v)/\Delta t_k$ and using $\hat{G}(t,u,v) \approx \nabla \hat{L}(t,u,v)$

A high frequency, temporally sparse, measurement of the normal component Φ_n of Φ is

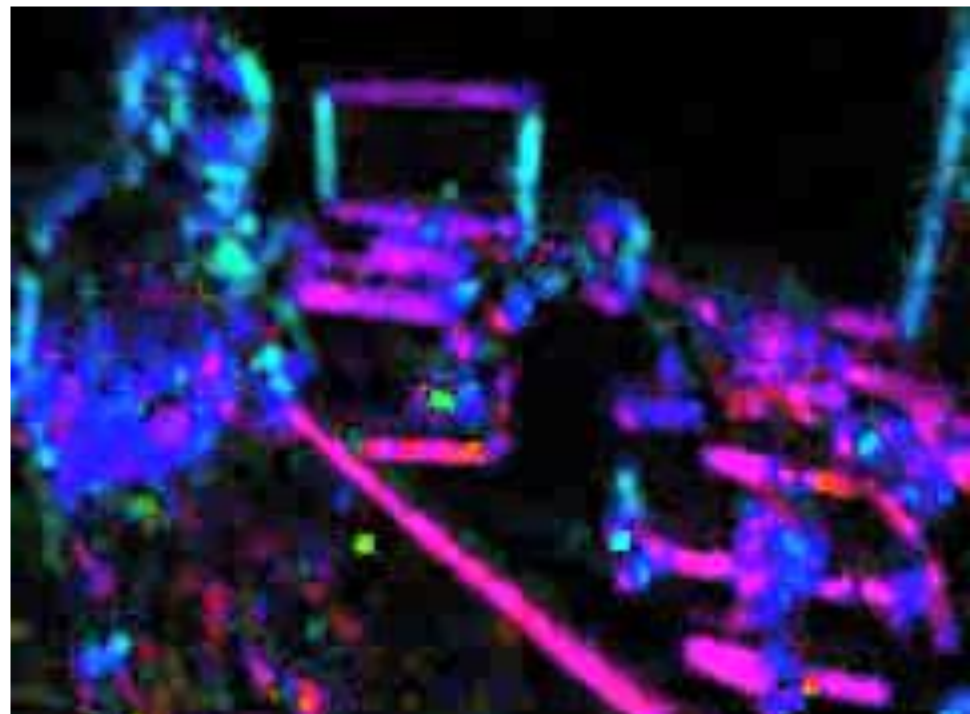
$$\Phi_n(t,u,v) = -\frac{E(t,u,v)}{\Delta t_k}\frac{\hat{G}(t,u,v)}{|\hat{G}(t,u,v)|^2}$$

Complementary filter for Optic Flow



By moving the high-frequency input in front of the integerator then it high-pass filtered $\frac{s}{s+\alpha}$ without being integrated.

If a low-frequency estimate of flow (CNN) is available $W_{(u,v)}(t)$ this can be fused.



Harris Corner tracking

Let $\hat{G}(t,u,v)$ denote the gradient image state.

A modified Harris matrix is given by

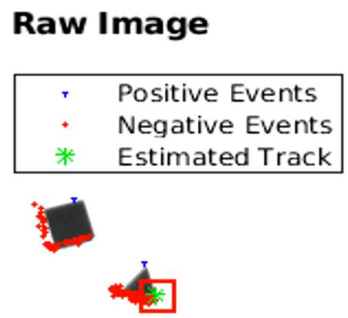
$$M(t,u,v) := (K * \hat{G}(t,u,v))(K * \hat{G}(t,u,v))^\top.$$

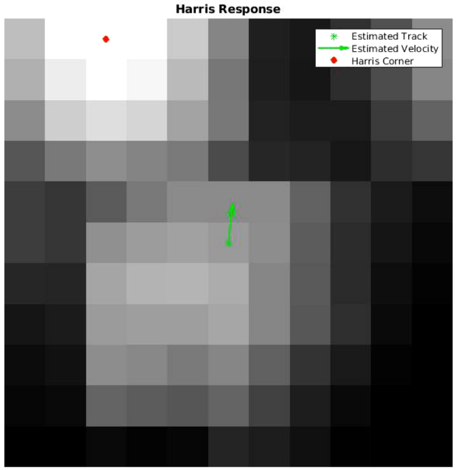
where K is a smoothing kernel

The Harris corner-response is

$$H(t,u,v) := \det(M(t,u,v))/\operatorname{tr}(M(t,u,v)),$$

The smoothed gradient $(K \star \hat{G}(t,u,v))$ is linear. The Harris matrix $M(t,u,v)$ and the Harris response $H(t,u,v)$ are both nonlinear and must be computed from filtered variables as required.





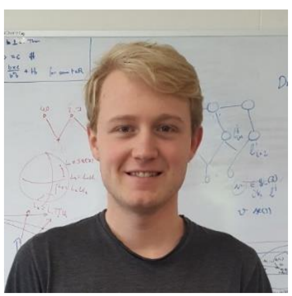
THANKS

Yonhon Ng

Cedric Scheerlinck

Anton Gvozdev

Nick Barnes



Pieter van Goor

Ziwei Wang