Fusing Frame and Event data for High Dynamic Range Video

Robert Mahony

THE AUSTRALIAN NATIONAL UNIVERSITY

Systems Theory and Robotics group

CVPR 2021 Workshop on Event-based Vision
Saturday, 19 June 2021
(Sunday 20 June, 12:30am, AEST)

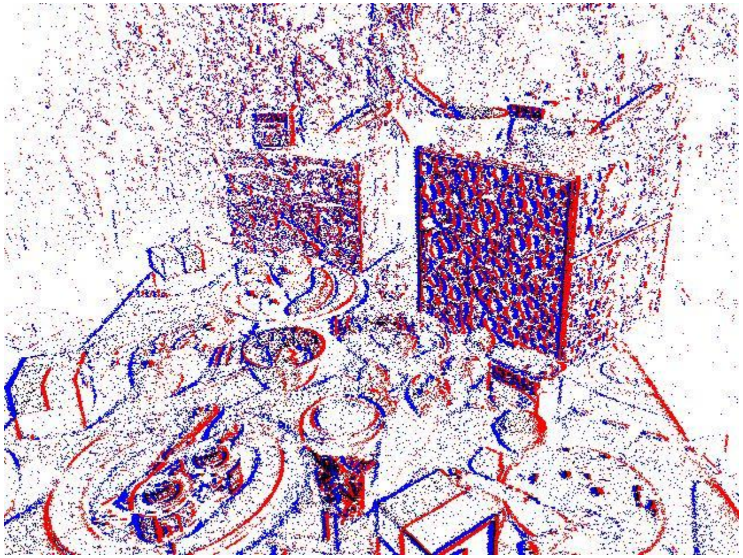
Event versus Frame Cameras

Event Camera

- Asynchronous events
- Temporally dense information
- No image blur.
- High dynamic range

Frame camera

- Synchronous images
- Spatially dense information
- Adjustable exposure
- Images absolute intensity
- Images static scenes.



Events and Frame data from picnic dataset

By fusing event and frame data it should be possible to have it all

- Images that are spatially and temporally dense
  - Full image available at any time stamp.
- Images with High Dynamic Range (HDR) in absolute intensity scale.
- Able to image both static scenes and highly dynamic scenes without blur.

High Dynamic Range Reconstruction

Raw
Frame

E2VID
Event
only

Asynchronous
Kalman
Filter
Event-Frame

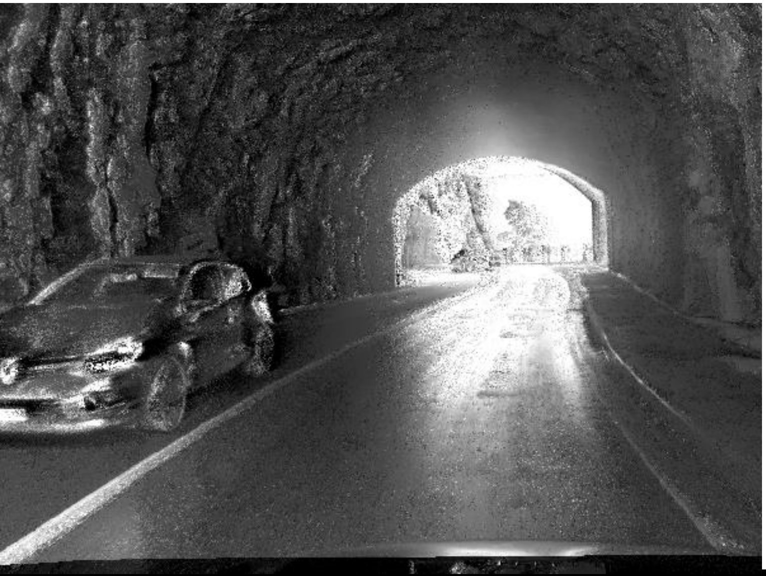
ECNN
Event
only

DSEC dataset
Gehrig et. al

High Dynamic Range Reconstruction

Raw Frame

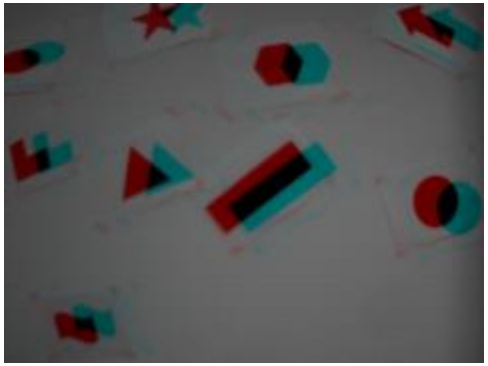
E2VID Event only



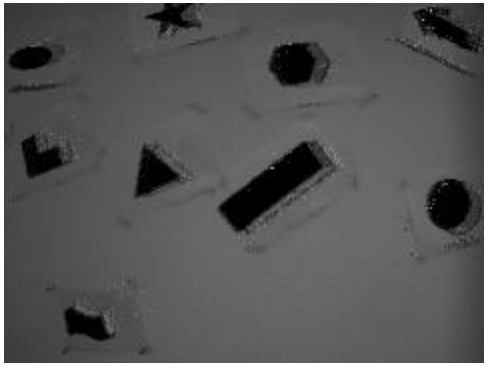
Asynchronous Kalman Filter Event-Frame

ECNN Event only

DSEC dataset
Gehrig et. al



Raw
Frame



Event-based
Double
Integral (EDI)

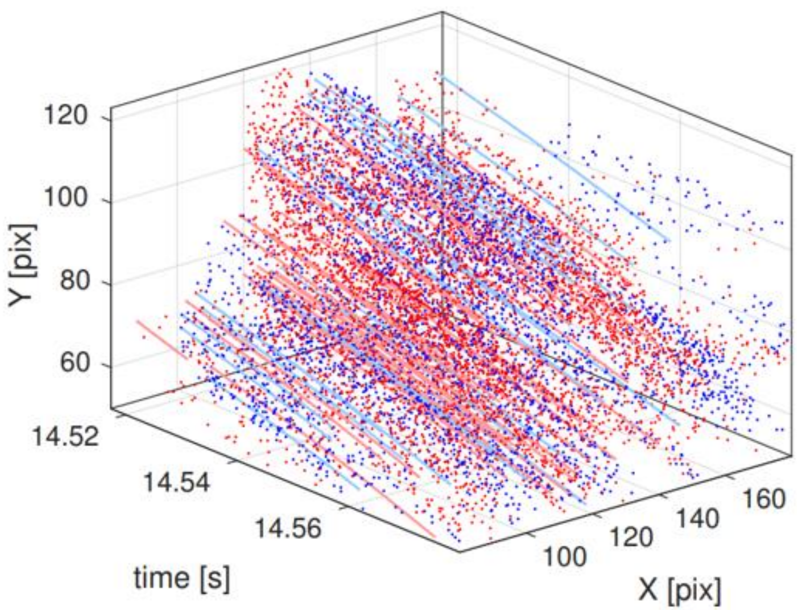
Asynchronous
Kalman
Filter
Event-Frame

Shapes data set
Mueggler et al

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.



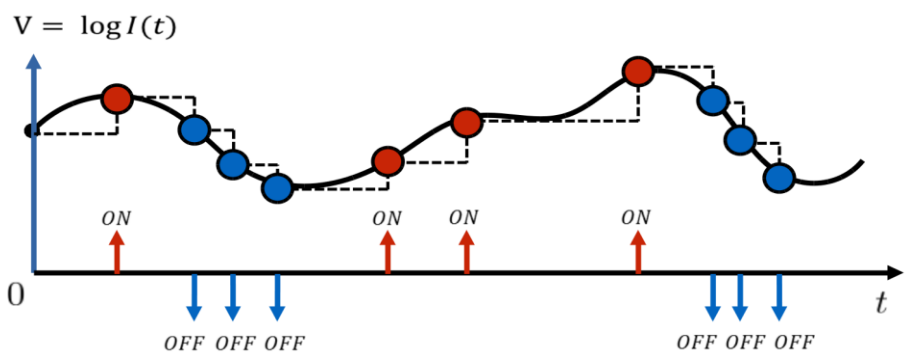
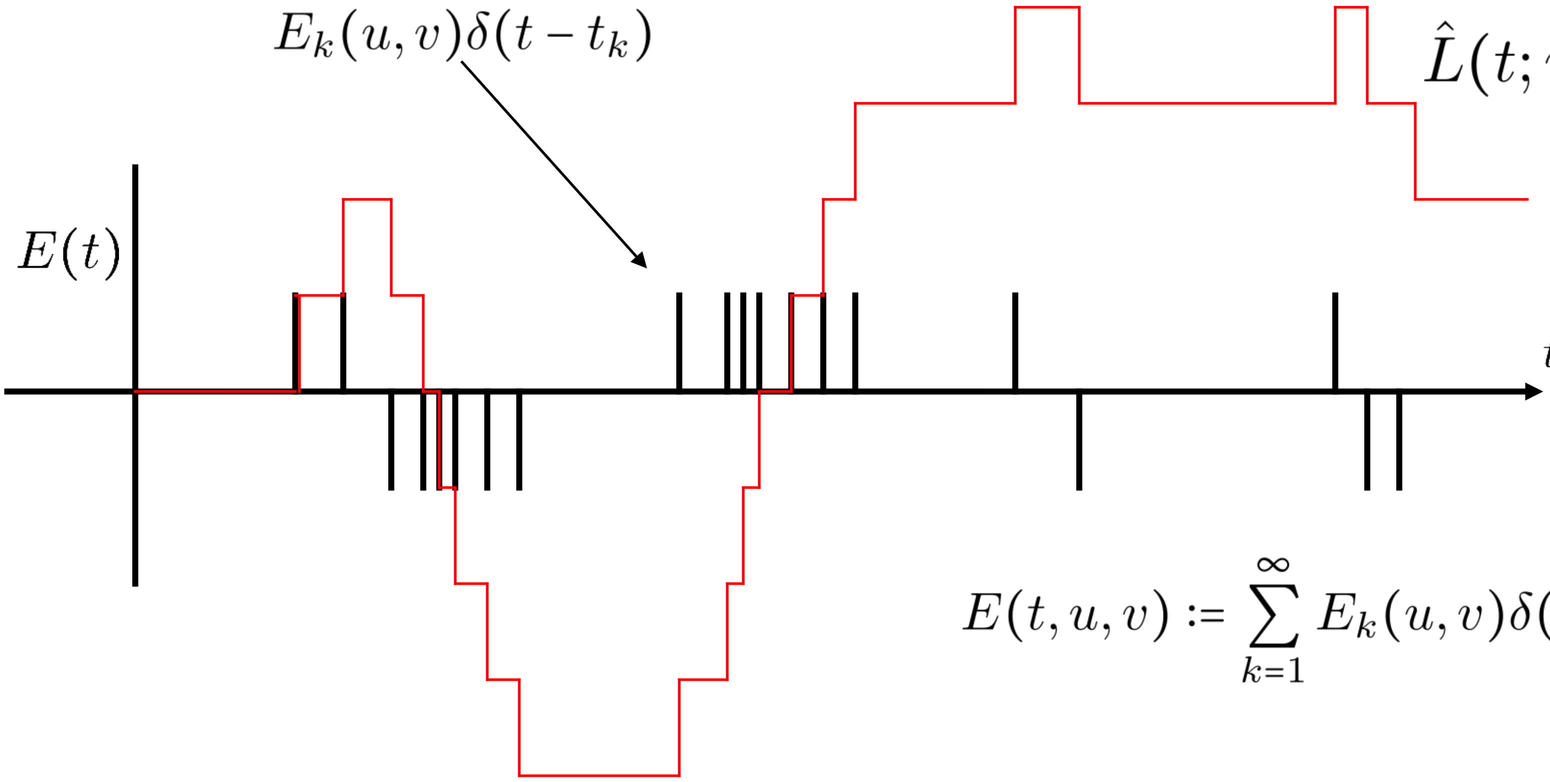


Image reconstruction



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

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

Direct integration

$$\hat{L}(t; u, v) = \int_{-\infty}^{t} E(\tau, u, v) \mathrm{d}\tau$$

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

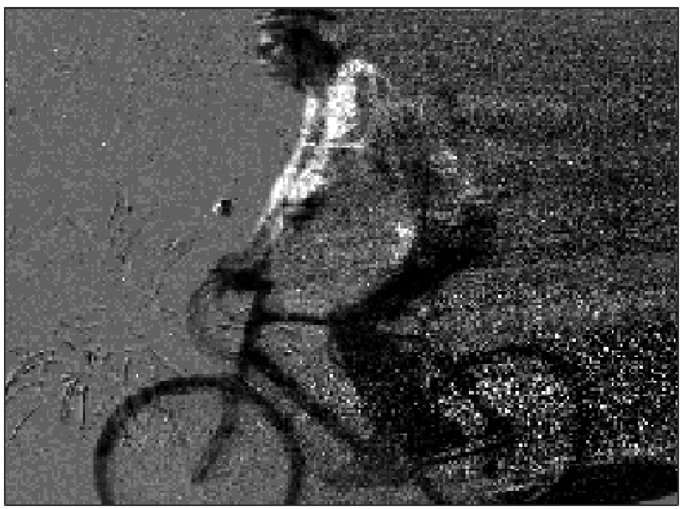
Transfer function interpretation of direct integration

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

High pass the state estimate filter

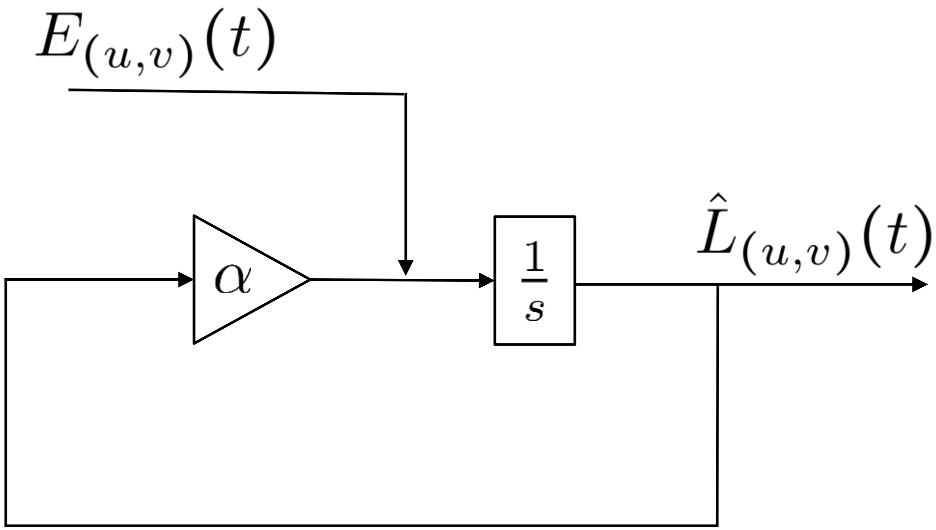
$$E_{(u,v)}(t) \longrightarrow \boxed{\frac{1}{s}} \longrightarrow \boxed{\frac{s}{s+\alpha}} \longrightarrow \hat{L}_{(u,v)}(t)$$

Integration High pass filter



Integration without high pass

Integration with high pass



Transfer function realisation

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

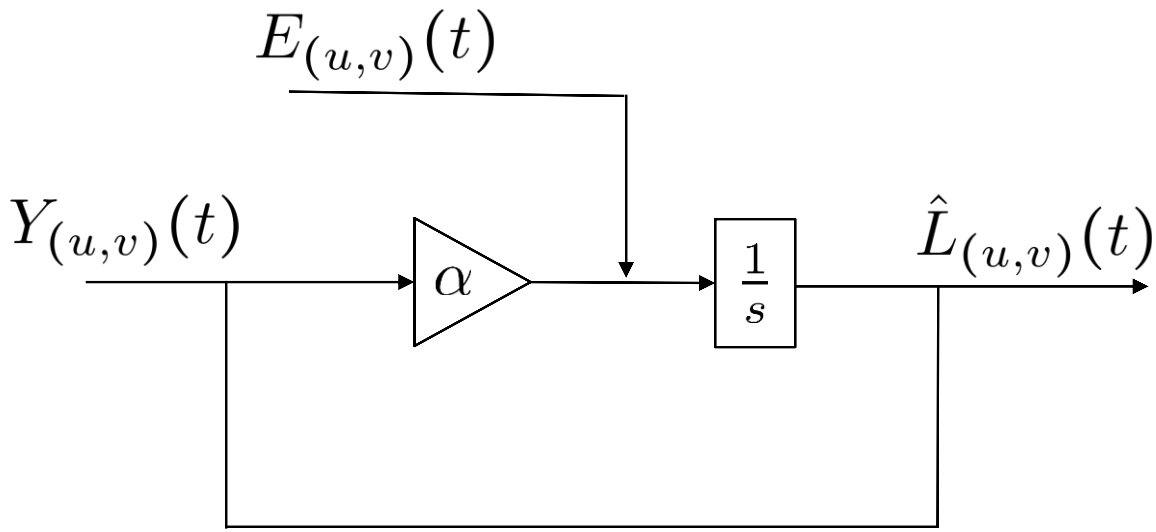
ODE system realisation

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

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

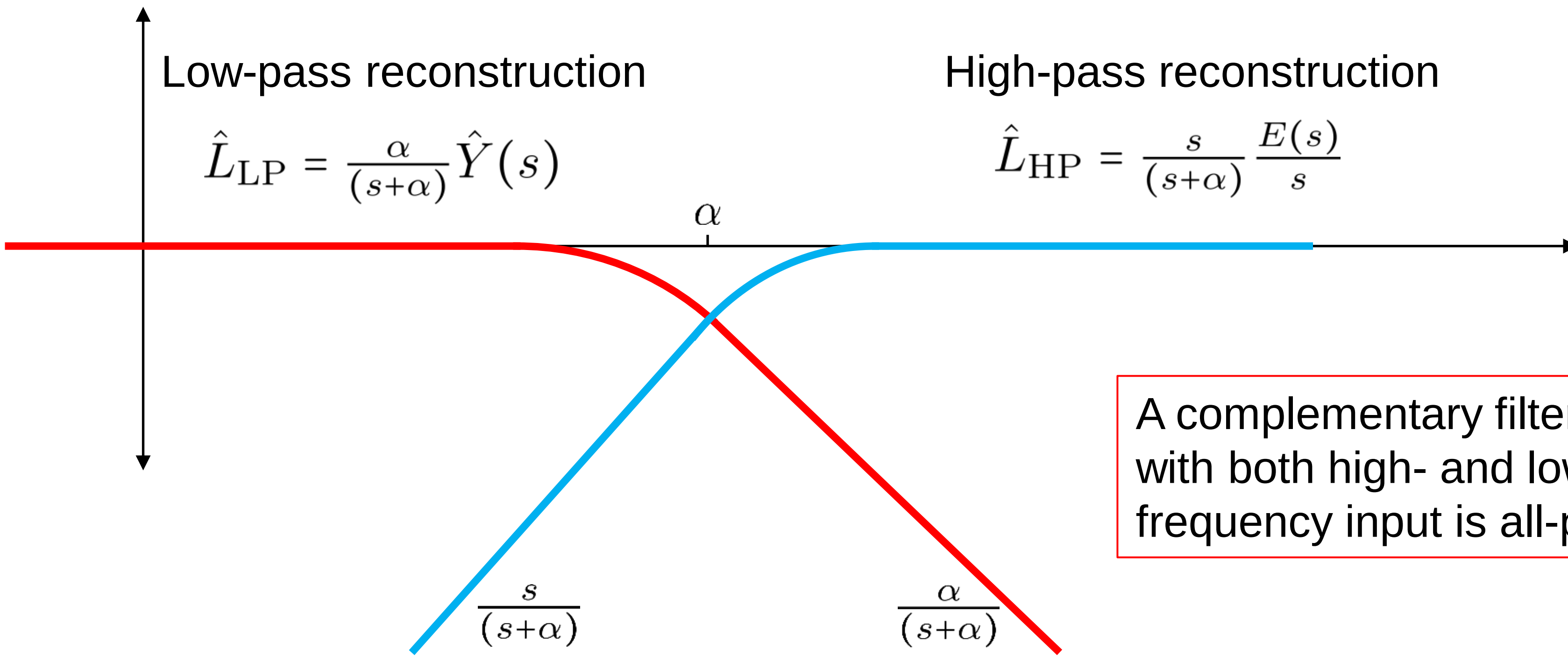
$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{L}_{(u,v)}(t) = -\alpha(\hat{L}_{(u,v)}(t) - Y_{(u,v)}(t)) + E_{(u,v)}(t)$$

If

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

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

Frequency interpretation



A complementary filter with both high- and low-frequency input is all-pass

$$\hat{L}(s) = \hat{L}_{\mathrm{LP}} + \hat{L}_{\mathrm{HP}} = \frac{\alpha}{(s+\alpha)}\hat{Y}(s) + \frac{s}{(s+\alpha)}\frac{\hat{E}(s)}{s}$$

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

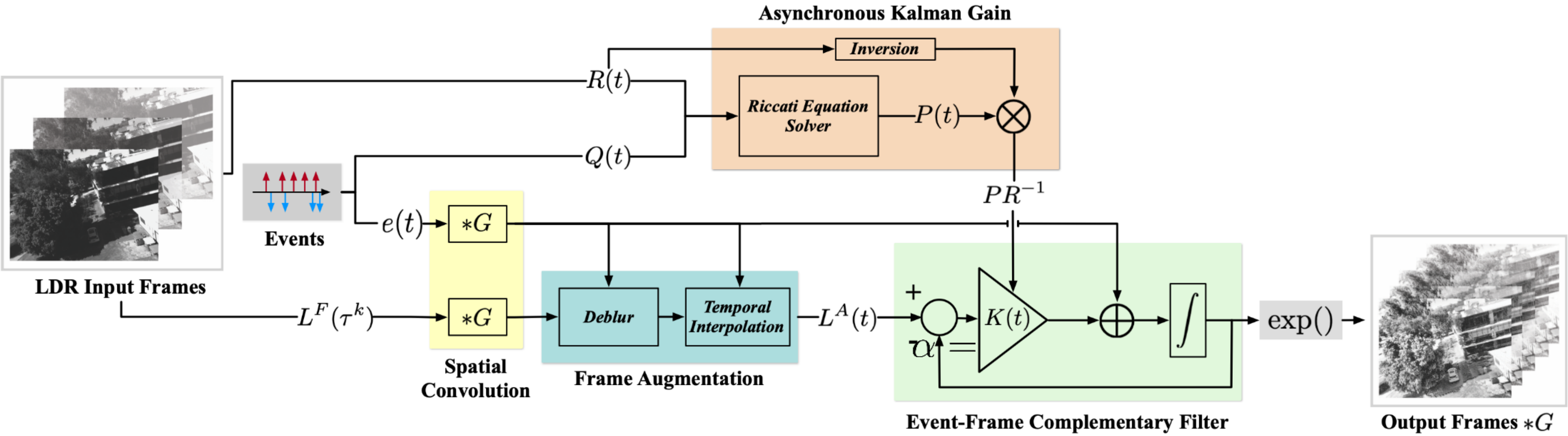
- Asynchronous: Only compute when an event arrives.
- Computationally efficient: One scalar exponential.
- Image state: Estimate $\hat{L}(t,u,v)$ is stored in memory and can be accessed whenever required.

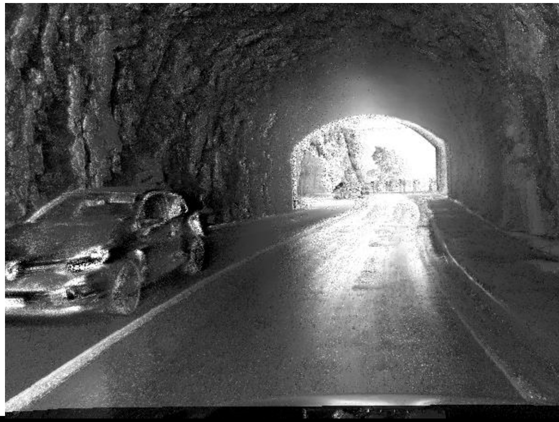
The gain α used in the image reconstruction filter is tuned by hand. Typical value $\alpha = 6\text{rad/s}$.

However, adaptively tuning the gain will produce much better response across the full image.

- Pixels where the conventional camera is properly exposed should trust the conventional camera response.
- Pixels where the conventional camera is under or over exposed should trust the event camera response.

How should the gain α be adaptively tuned.

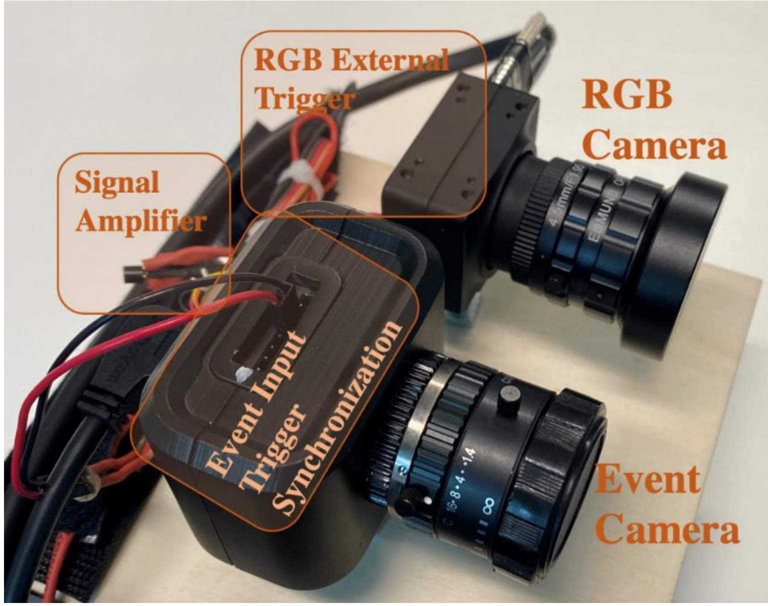

Asynchronous Kalman Gain
Inversion
Riccati Equation Solver
$R(t)$
$Q(t)$
$P(t)$
PR^{-1}
LDR Input Frames
Events
$e(t)$
$*G$
$*G$
$L^F(\tau^k)$
Spatial Convolution
Deblur
Temporal Interpolation
Frame Augmentation
$L^A(t)$
+
$-\alpha$
$K(t)$
$\int$
exp()
Event-Frame Complementary Filter
Output Frames $*G$



Results

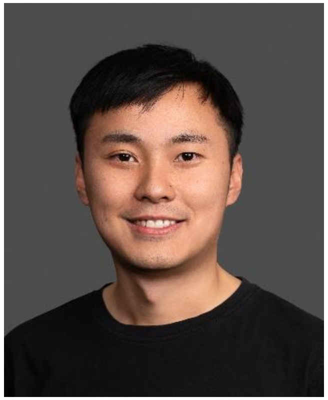
Asynchronous Kalman filter (AKF)

- Exploit the asynchronous nature of the sensor with algorithm design and implementation.
- Use both frame and event data.



- A shallow algorithm (no deep learning)

As the quality of event camera sensors improves so will the output of the AKF



Yonhon Ng

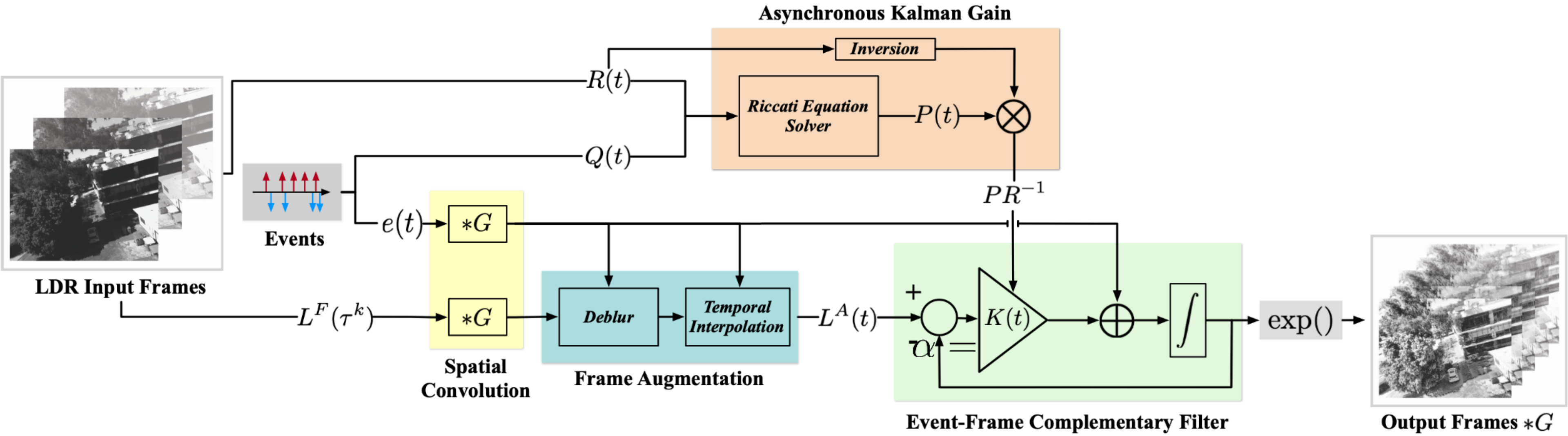
System Theory and Robotics group

THANKS

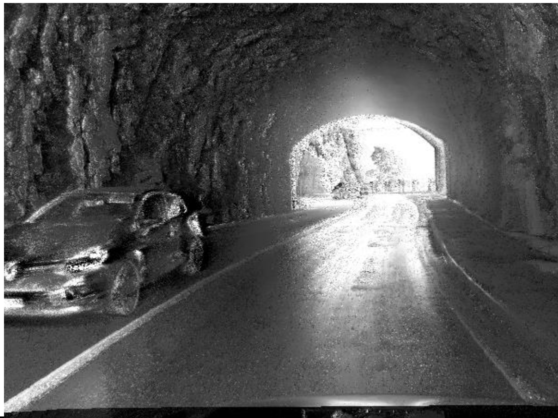
Cedric Scheerlinck

Ziwei Wang

Asynchronous Kalman Filter



Details



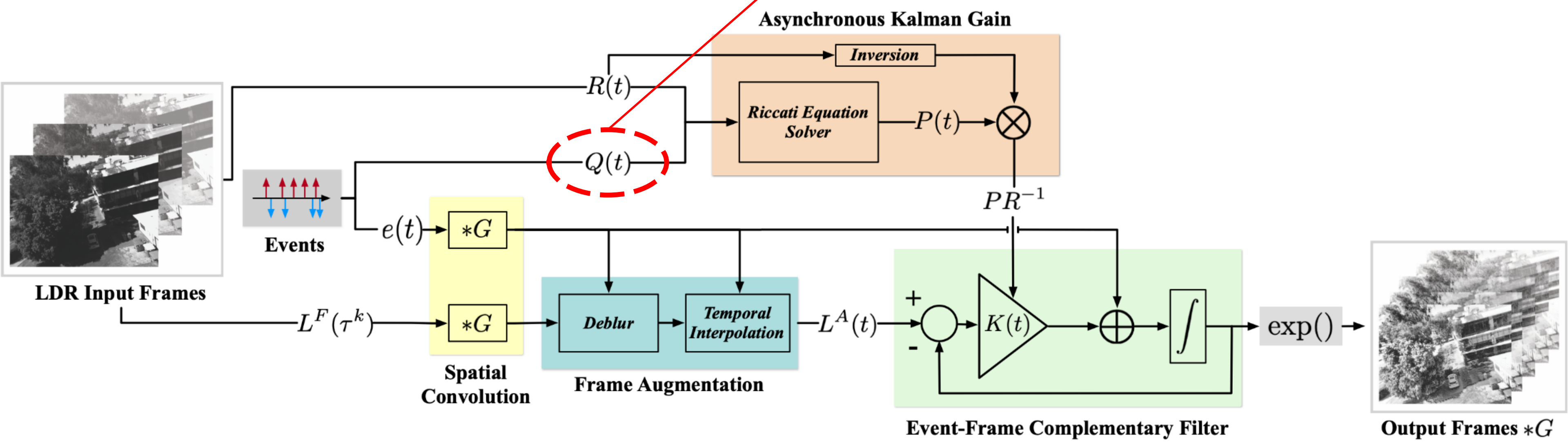
Noise in the event data

$$e_{\vec{p}}(t) = \sum_{i=1}^{\infty} (c\sigma^i_{\vec{p}} + \eta^i_{\vec{p}})\delta(t - t^i_{\vec{p}}), \qquad \eta^i_{\vec{p}} \sim \mathcal{N}\left(0, Q^{\text{proc.}}_{\vec{p}}(t) + Q^{\text{iso.}}_{\vec{p}}(t) + Q^{\text{ref.}}_{\vec{p}}(t)\right)$$

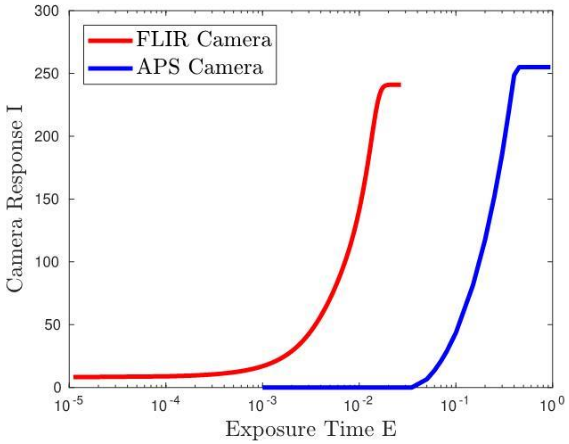
Process noise: $Q^{\text{proc.}}_{\vec{p}}(t^i_{\vec{p}}) = \sigma^2_{\text{proc.}}(t^i_{\vec{p}} - t^{i-1}_{\vec{p}})$

Isolated pixel noise: $Q^{\text{iso.}}_{\vec{p}}(t^i_{\vec{p}}) = \sigma^2_{\text{iso.}} \min\{t^i_{\vec{p}} - t^*_{N(\vec{p})}\}$

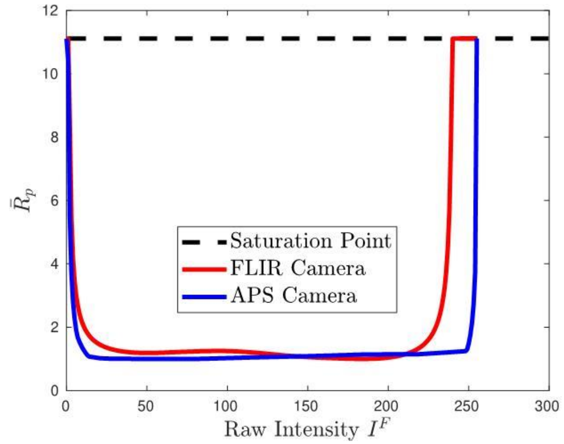
Refractory period noise: $Q^{\text{ref.}}_{\vec{p}}(t^i_{\vec{p}}) = \begin{cases} 0 & \text{if } t^i_{\vec{p}} - t^{i-1}_{\vec{p}} > \bar{\rho} \\ \sigma^2_{\text{ref.}} & \text{if } t^i_{\vec{p}} - t^{i-1}_{\vec{p}} \leq \bar{\rho}, \end{cases}$

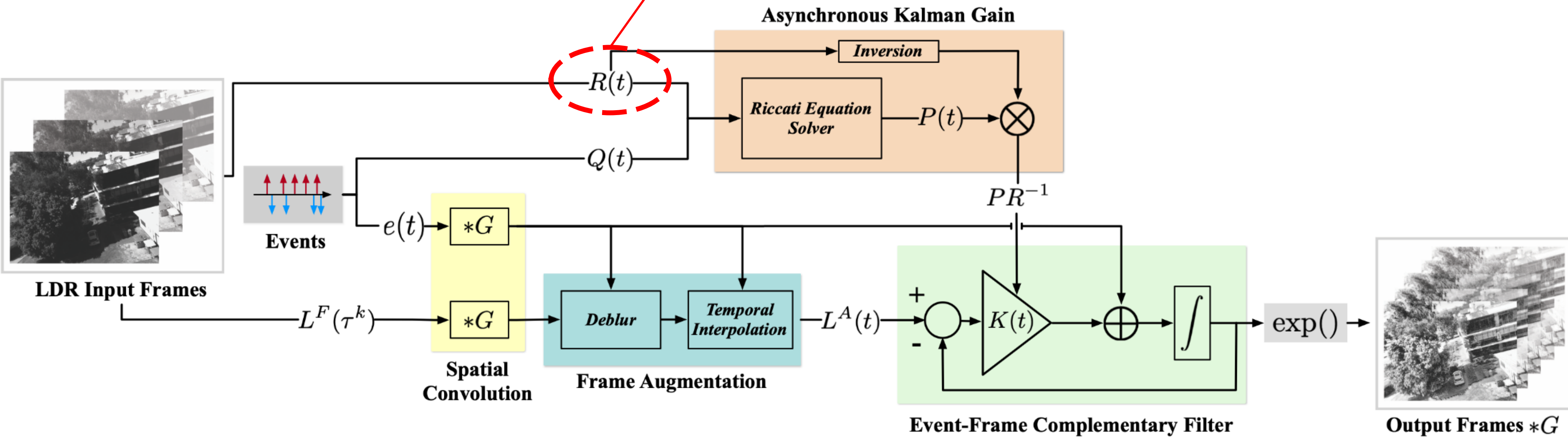


Noise in frame data



$$I^F_{\vec{p}}(\tau^k) = CRF^{-1}\lfloor I_{\vec{p}}(\tau^k)\rfloor + \mu^k_{\vec{p}},$$
$$\mu^k_{\vec{p}} \sim N(0, \bar{R}_{\vec{p}}(\tau^k)),$$



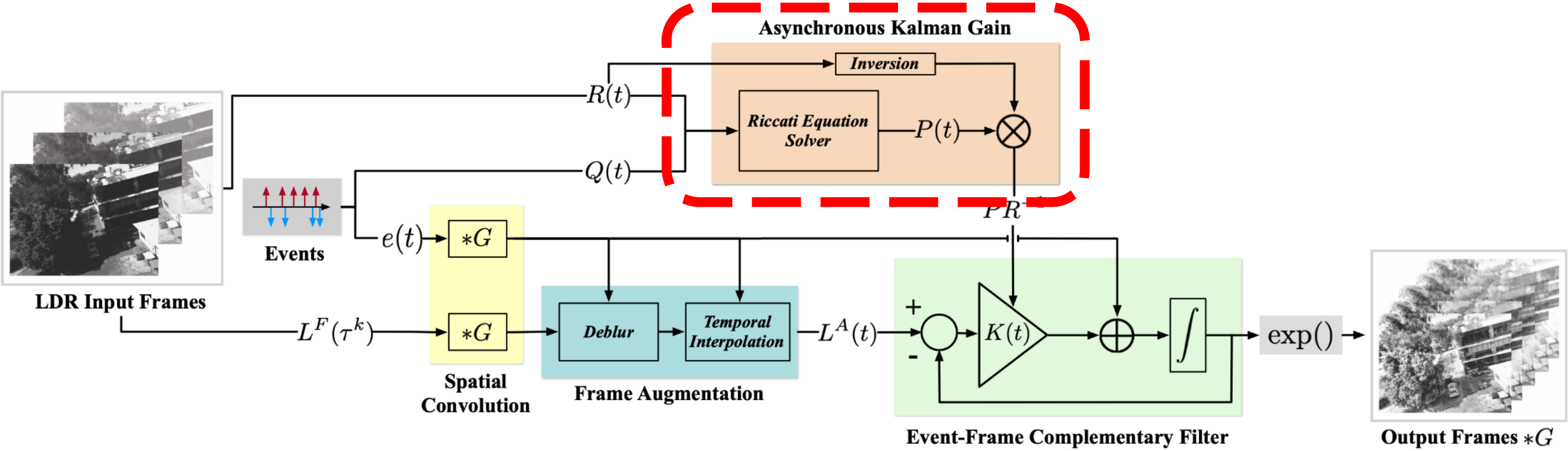


Riccati computation

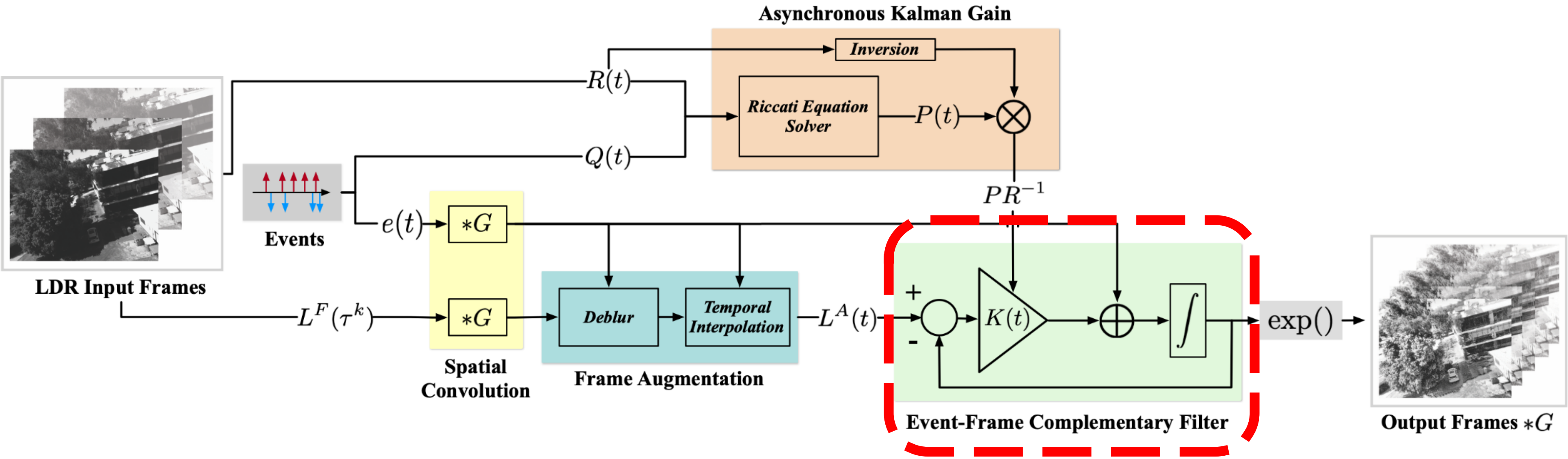
$$P_{\vec{p}}(t) = \frac{1}{P_{\vec{p}}^{-1}(t_{\vec{p}}^{i}) + R_{\vec{p}}^{-1}(t)\cdot(t - t_{\vec{p}}^{i})} \quad \text{for } t \in [t_{\vec{p}}^{i}, t_{\vec{p}}^{i+1})$$

$$\boxed{\dot{P}_{\vec{p}} = -P_{\vec{p}}^{2} R_{\vec{p}}(t) + Q_{\vec{p}}(t)}$$

$$P_{\vec{p}}(t_{\vec{p}}^{i}) = P_{\vec{p}}(t_{\vec{p}}^{i-}) + Q_{\vec{p}}(t_{\vec{p}}^{i})$$



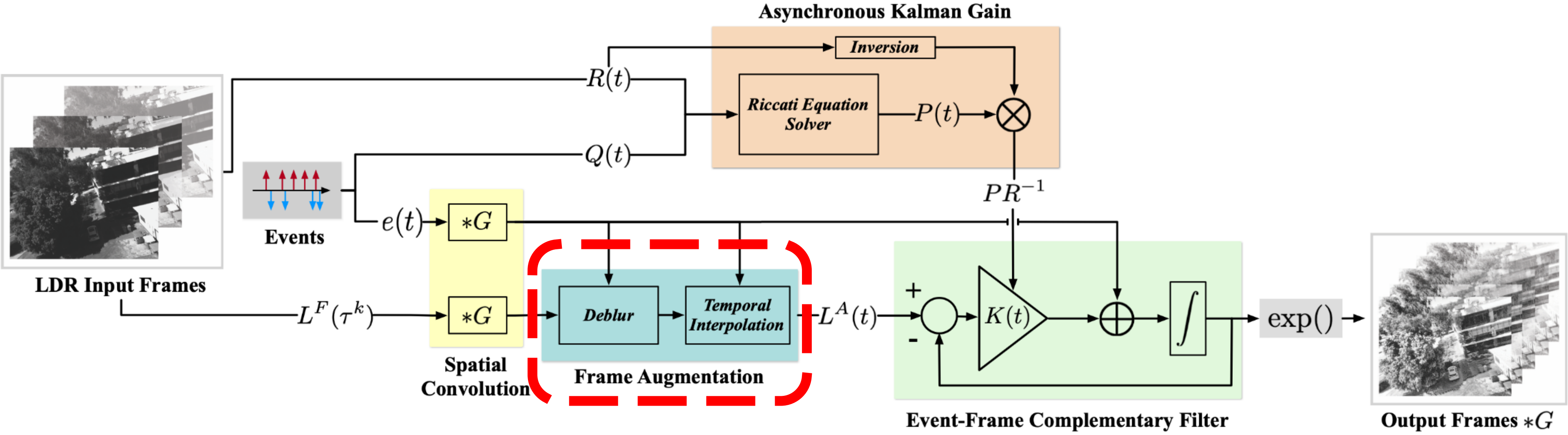
Filter update



$$\hat{L}_{\vec{p}}(t) = \frac{[\hat{L}_{\vec{p}}(t^i_{\vec{p}}) - L^A_{\vec{p}}(t^i_{\vec{p}})] \cdot P^{-1}_{\vec{p}}(t^i_{\vec{p}})}{P^{-1}_{\vec{p}}(t^i_{\vec{p}}) + R^{-1}_{\vec{p}}(t) \cdot (t - t^i_{\vec{p}})} + L^A_{\vec{p}}(t)$$

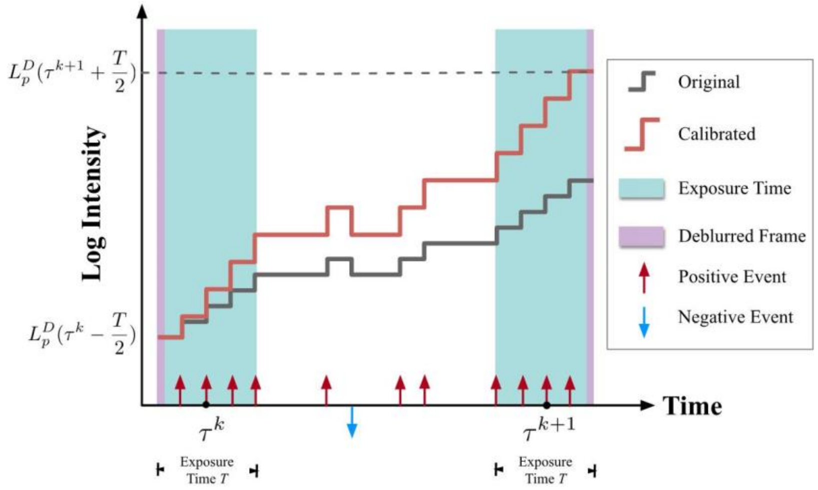
$$\boxed{\dot{\hat{L}}_{\vec{p}}(t) = e_{\vec{p}}(t) - K_{\vec{p}}(t)[\hat{L}_{\vec{p}}(t) - L^A_{\vec{p}}(t)]}$$

$$L_{\vec{p}}(t^i_{\vec{p}}) = \hat{L}_{\vec{p}}(t^{i-}_{\vec{p}}) + e_{\vec{p}}(t^i_{\vec{p}})$$

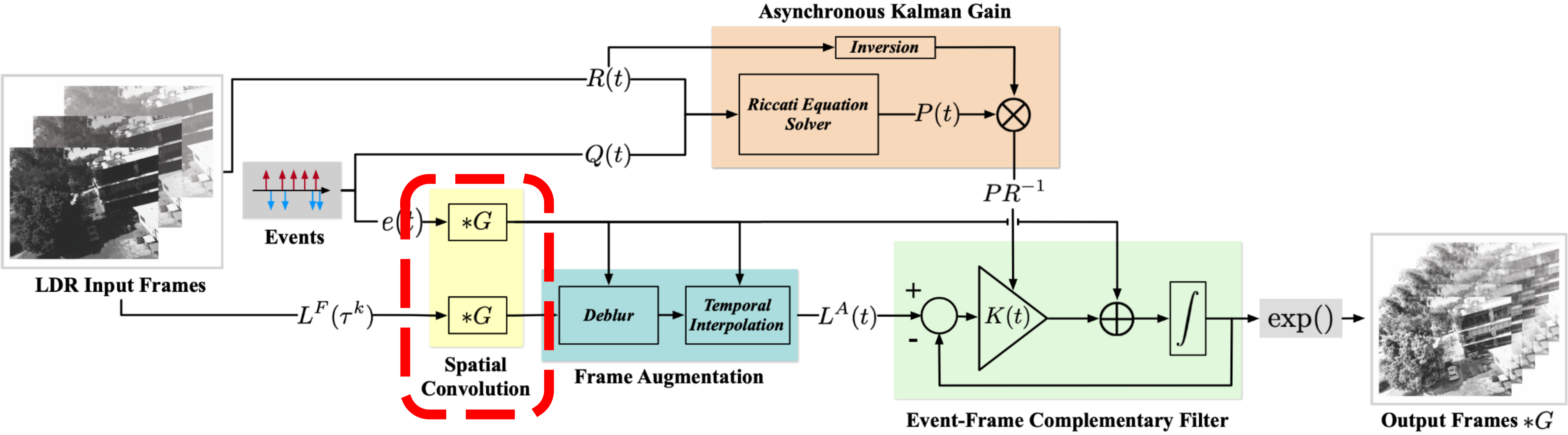


Conventional computer vision image processing:

- Use events to deblur conventional frames
- Use events to interpolate between conventional frames



Linear convolution



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

Each event #(K) events

Results