

Computing Spatial Image Convolutions for Event Cameras

Cedric Scheerlinck

Prof. Rob Mahony
A/Prof. Nick Barnes

Image Convolutions

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



$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

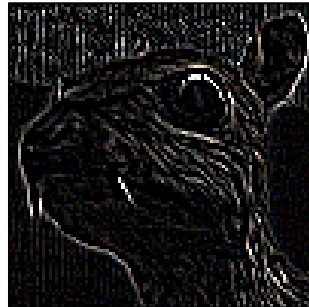


Image Convolutions

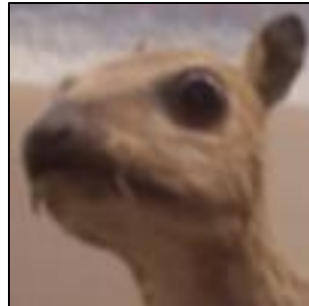
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



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



$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



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



$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

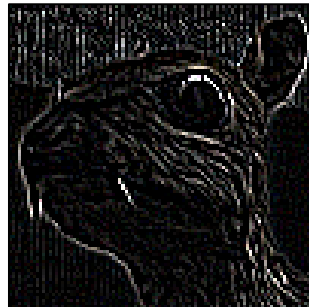


Image Convolutions

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



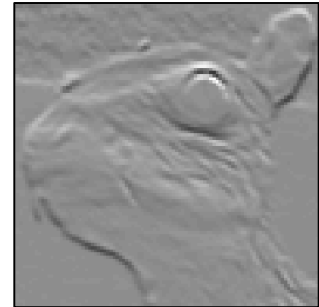
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



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



$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

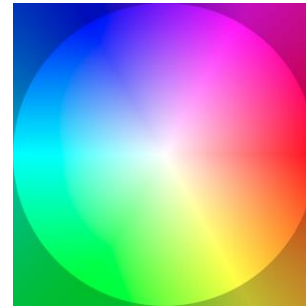
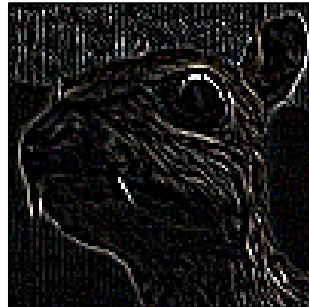


Image Convolutions for Events

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \text{img1} = \text{img2}$$
The diagram illustrates a 2D convolution operation. On the left, a 3x3 kernel matrix is shown: $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$. This is followed by an asterisk (*) indicating convolution with an input image. The input image is a color photograph of a squirrel's head in profile, facing left. To the right of the asterisk is an equals sign (=), followed by the output image. The output image is a grayscale version of the input, where the convolution kernel has been applied to detect edges and features, resulting in a high-contrast, edge-detected representation of the squirrel's head.

Image Convolutions for Events

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \text{Image of a dog's head} = \text{Edge-detected image of the dog's head}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \begin{array}{c} \text{ON} \\ \uparrow \\ \text{---} t \text{---} \\ \downarrow \downarrow \downarrow \\ \text{OFF OFF OFF} \end{array} = ?$$

Image Convolutions for Events

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \text{Image of a dog's head} = \text{Filtered image of the dog's head}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \text{Event stream (ON/OFF events over time } t \text{)} = ?$$

Naïve approach: reconstruct image frames from events then apply convolution.

Can we do better?

Image Convolutions for Events


Consider one event  [timestamp, x, y, ± 1]

Image Convolutions for Events

Consider one event



[timestamp, x, y, ± 1]

Event image

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

Image Convolutions for Events

Consider one event



[timestamp, x, y, ± 1]

Event image

Kernel

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

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

Image Convolutions for Events

Consider one event



[timestamp, x, y, ±1]

Kernel * Event image

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

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

=

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

Image Convolutions for Events

Kernel * Event image

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

Six virtual events, or a **convolved event**, can be generated

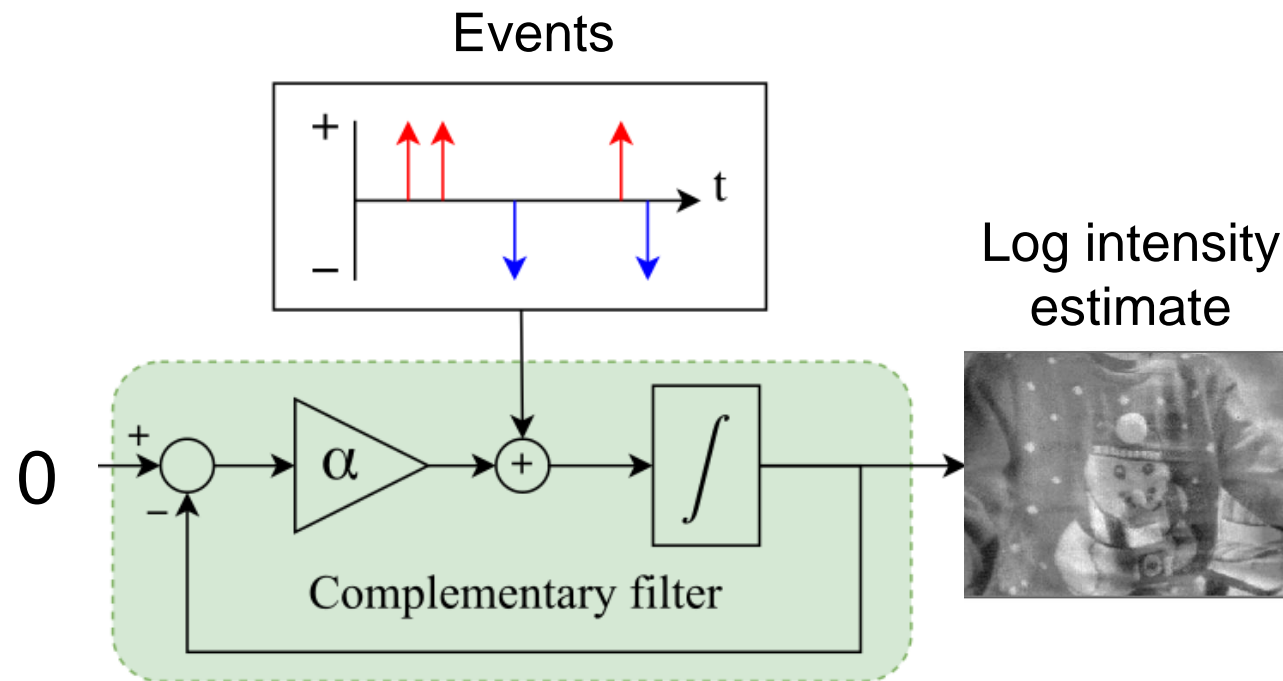
Image Convolutions for Events

Convolved events can be used as input to an event processing algorithm.

Image Convolutions for Events

Convolved events can be used as input to an event processing algorithm.

For example image reconstruction [1]:



[1] (Scheerlinck et al., ACCV, 2018)

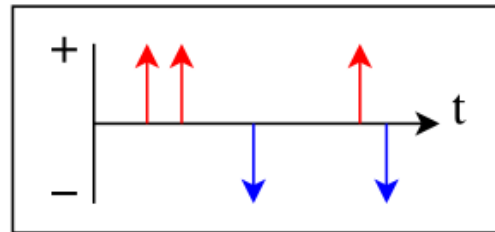
C. Scheerlinck, N. Barnes, R. Mahony, "Computing Spatial Image Convolutions for Event Cameras," arXiv, 2018.

Image Convolutions for Events

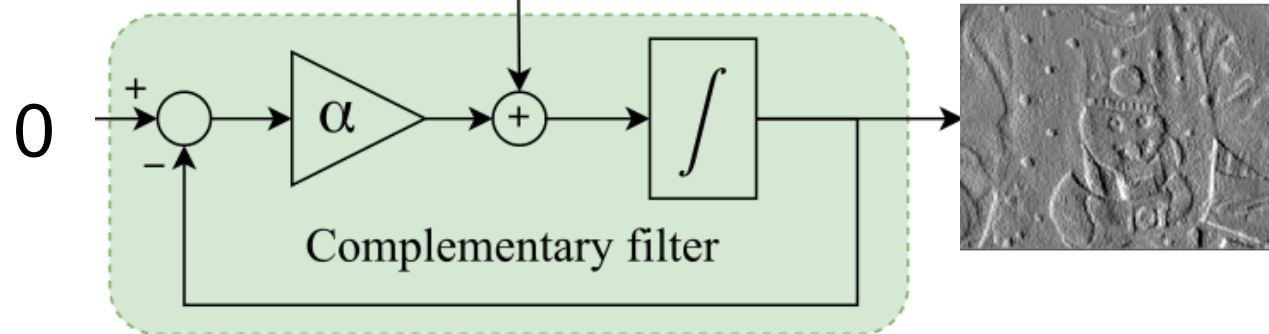
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$



Convolved events



Convolved
estimate



Results

Identity

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



Gaussian

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



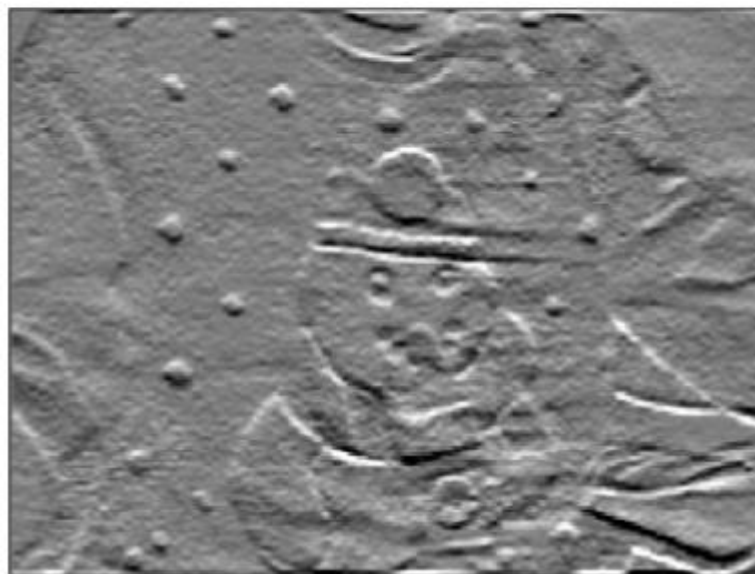
Results

Sobel

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



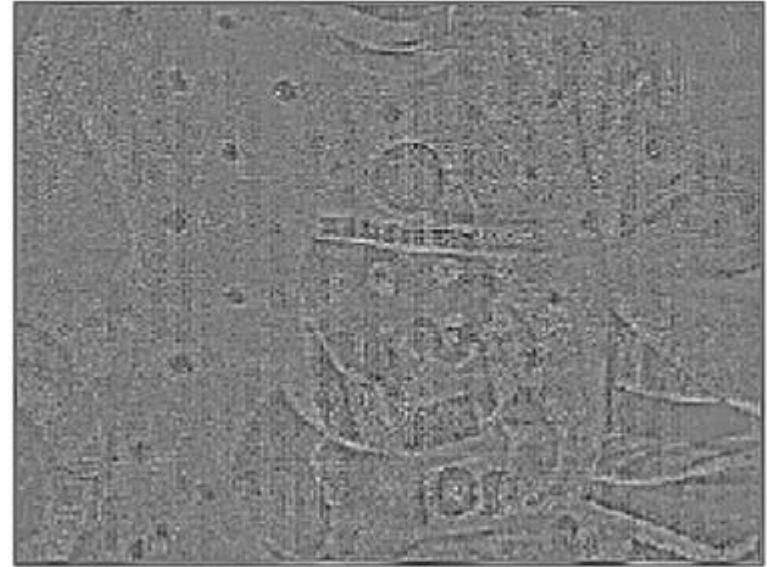
$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$



Results

Laplacian

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

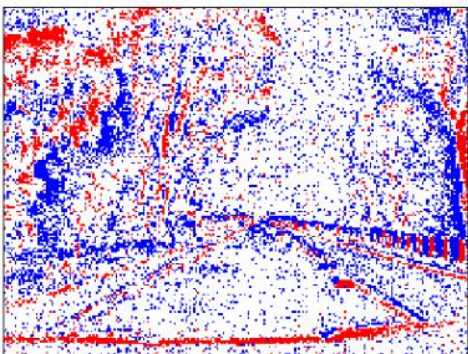
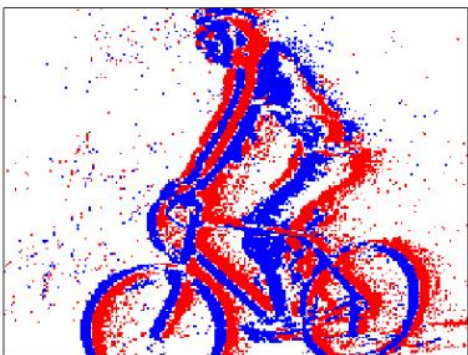
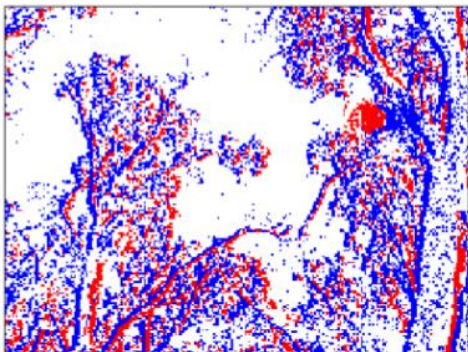


Poisson Reconstruction
from Laplacian

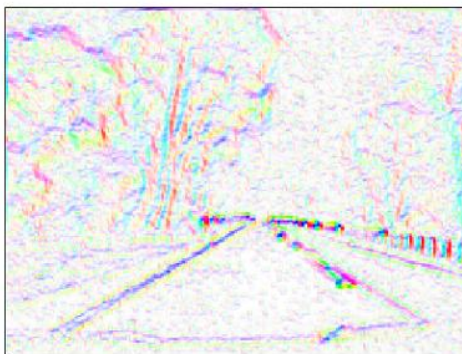
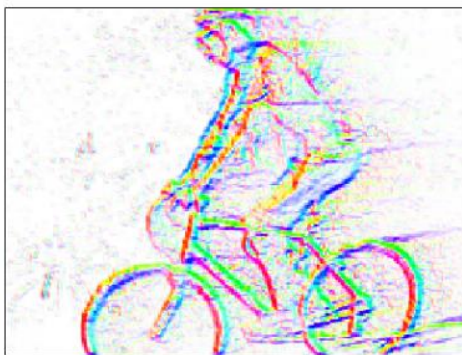
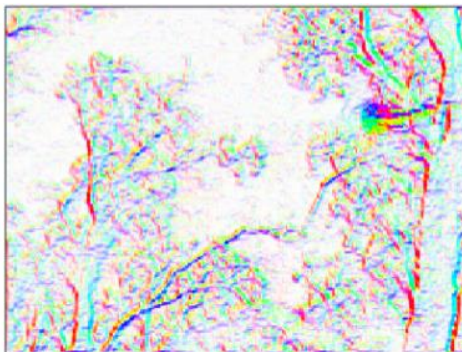
Perez et al. (2003), Agrawal et al. (2006)

Results

Current events



Gradient estimate



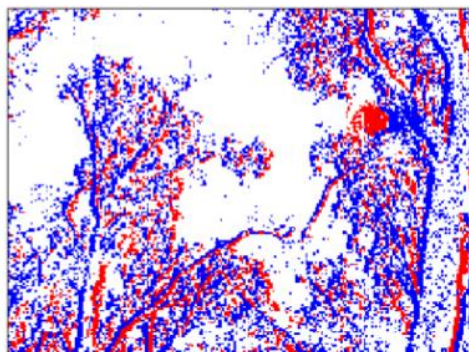
Sobel

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

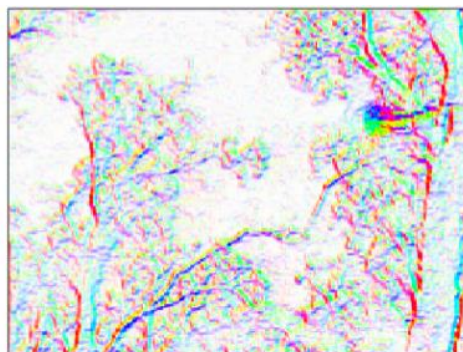
$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Results

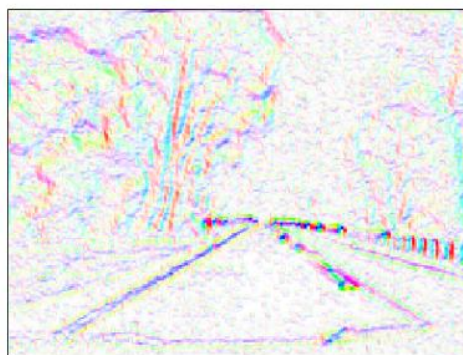
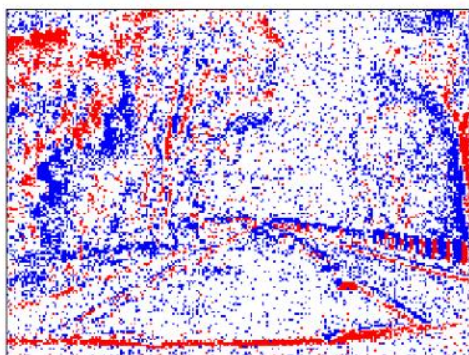
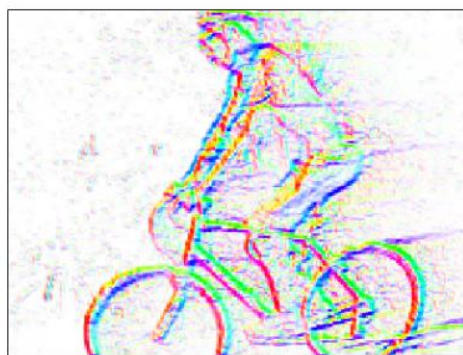
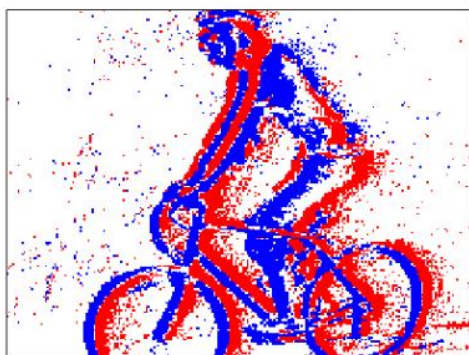
Current events



Gradient estimate



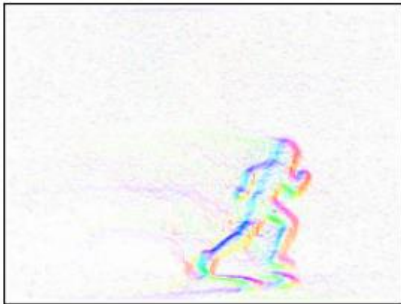
Poisson integration



Results

Gradient can be used as input to asynchronous Harris corner detector.

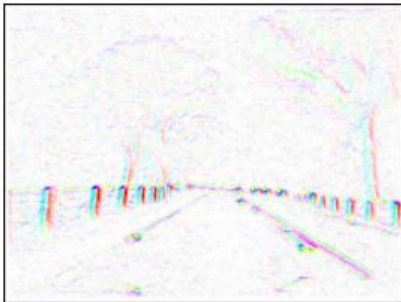
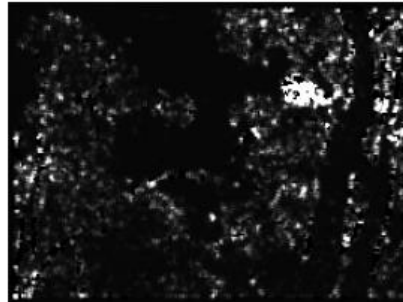
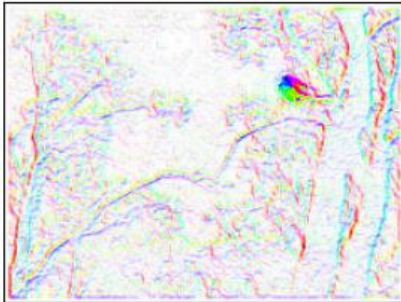
Gradient



Harris response

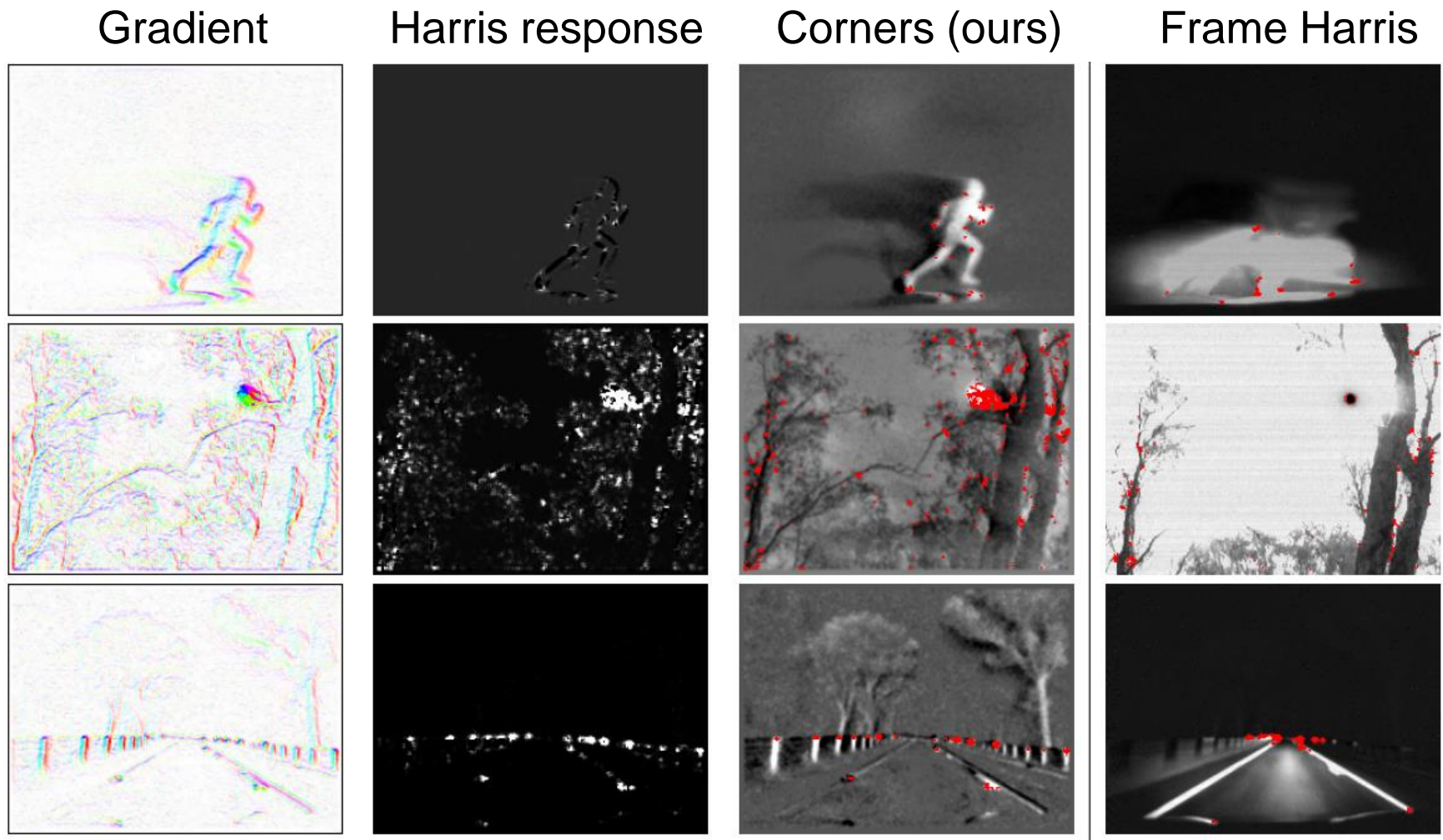


When an event arrives, the Harris response is only updated in a local neighbourhood.



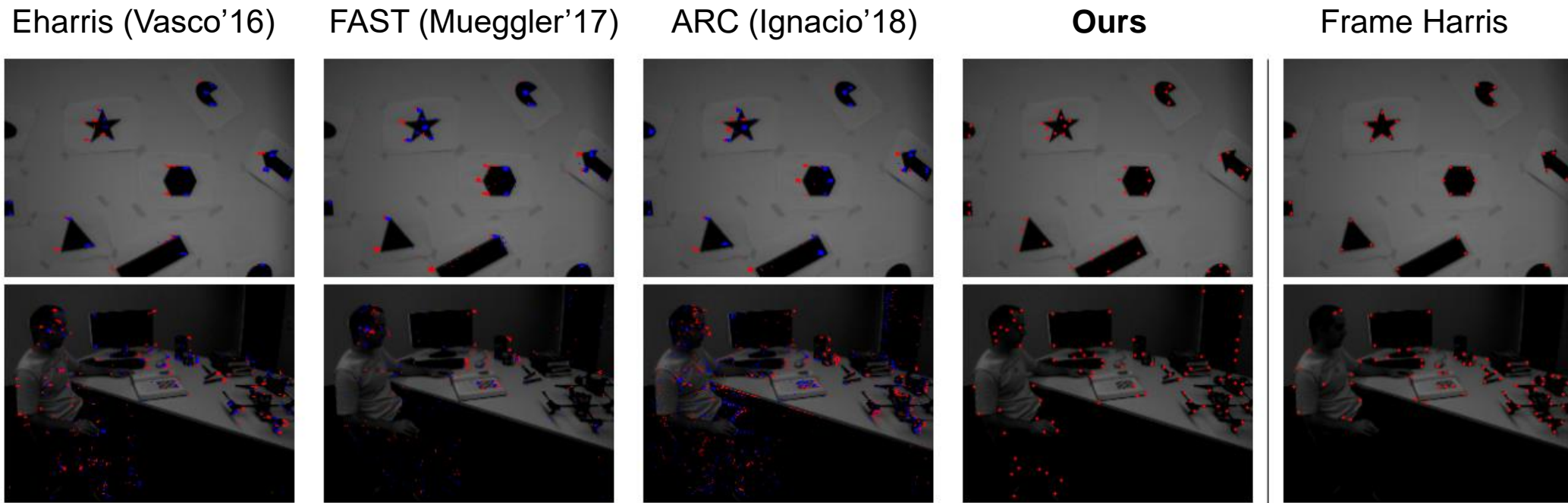
Results

Gradient can be used as input to asynchronous Harris corner detector.



Results

Comparison to state-of-the-art event-based corner detection.



Since we output a continuous-time Harris response state, at any point in time we can apply non-maximum suppression to get clean corners.

Conclusion

- We have introduced a methodology for event-based convolutions.
- Each event is individually convolved, producing a cluster of convolved events.
- Convolved events are fed into an asynchronous image reconstruction algorithm to produce a continuous-time state estimate of the convolved image.
- We introduce an asynchronous Harris corner detector based on gradients produced by our method.