EE5110/EE6110
Special Topics in Automation and Control
Segment B

From Frames to Events: Theory and Applications of Event-based Vision

Jiang Rui, Ph.D., BEng
rui.j@nus.edu.sg
Adjunct Lecturer
Department of Electrical and Computer Engineering

Semester 1, AY2021/2022

Course Structure

- **Lecture 1 – Event-based Vision System**
  - Conventional machine vision systems
  - Event-based sensors and cameras
  - Event data representation and compression
  - Event data simulation and datasets
  - Event data processing

Course Structure

- **Lecture 2 – Algorithms and Applications**
  - Feature detection and tracking
  - Optical flow estimation
  - Pose estimation and SLAM
  - Image reconstruction
  - Motion segmentation
  - Recognition
  - Event-based control

Continuous Assessment (CA) and Research Project (RP)

Important information:

- Project end time and date: All results must be submitted by
  11:59pm, FRIDAY, 1-OCT-2021 (Week 7)
  11:59pm, FRIDAY, 5-NOV-2021 (Week 12)
  to LumiNUS
  (folder: **Segment B-submission-CA/RP**).
- Submission: **Report, codes, demo video** in one zip file with name: **your_NUS_ID.zip**, e.g., A0123456X.zip.
- Please always remain contactable during the semester.
- Format of the report: Use IEEE Templates for Conference Proceedings. **No more than 5 pages for CA, no more than 6 pages for RP** allowed. The template (word/tex) have been uploaded to LumiNUS.

  You may also download it from
  https://www.ieee.org/conferences/publishing/templates.html

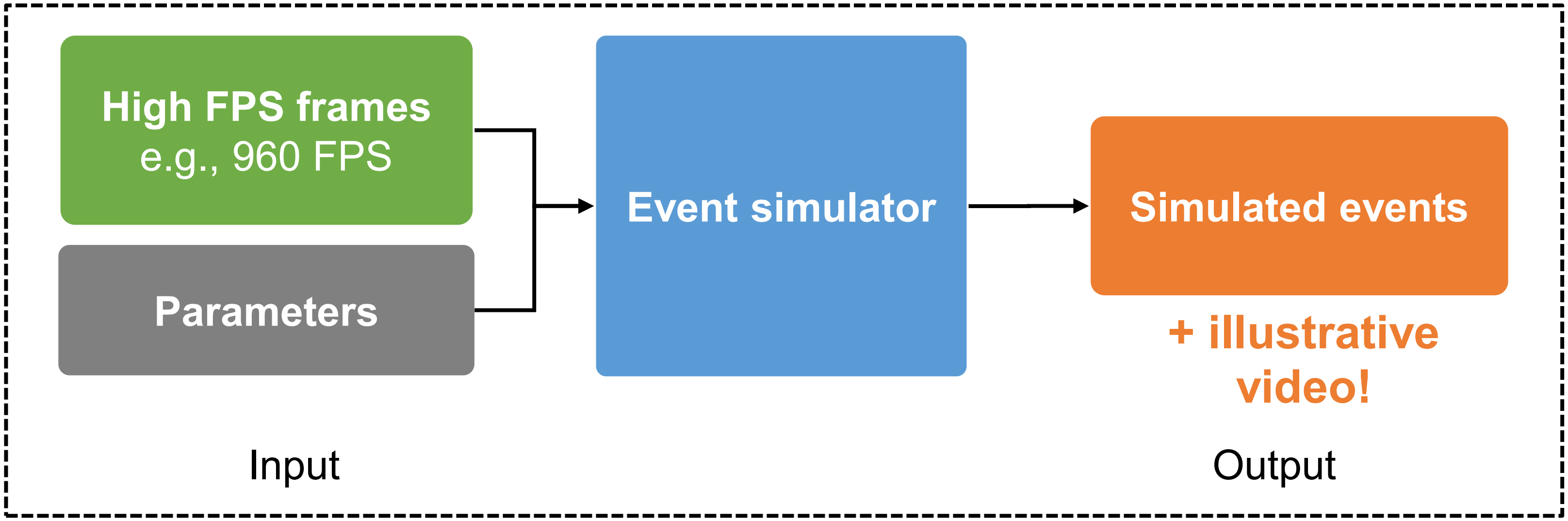
Continuous Assessment (CA) and Research Project (RP)

Important information:

- Written report should contains at least the following:
  1. An abstract that briefly summarizes what you have done and what the results are,
  2. a brief introduction to the project and the problem that you are solving,
  3. derivation of the model (“problem formulation”),
  4. description of the framework and algorithm(s) used
  5. the results,
  6. conclusions and/or a summary,
  7. references (if applicable).

CA: Design an Event Data Simulator

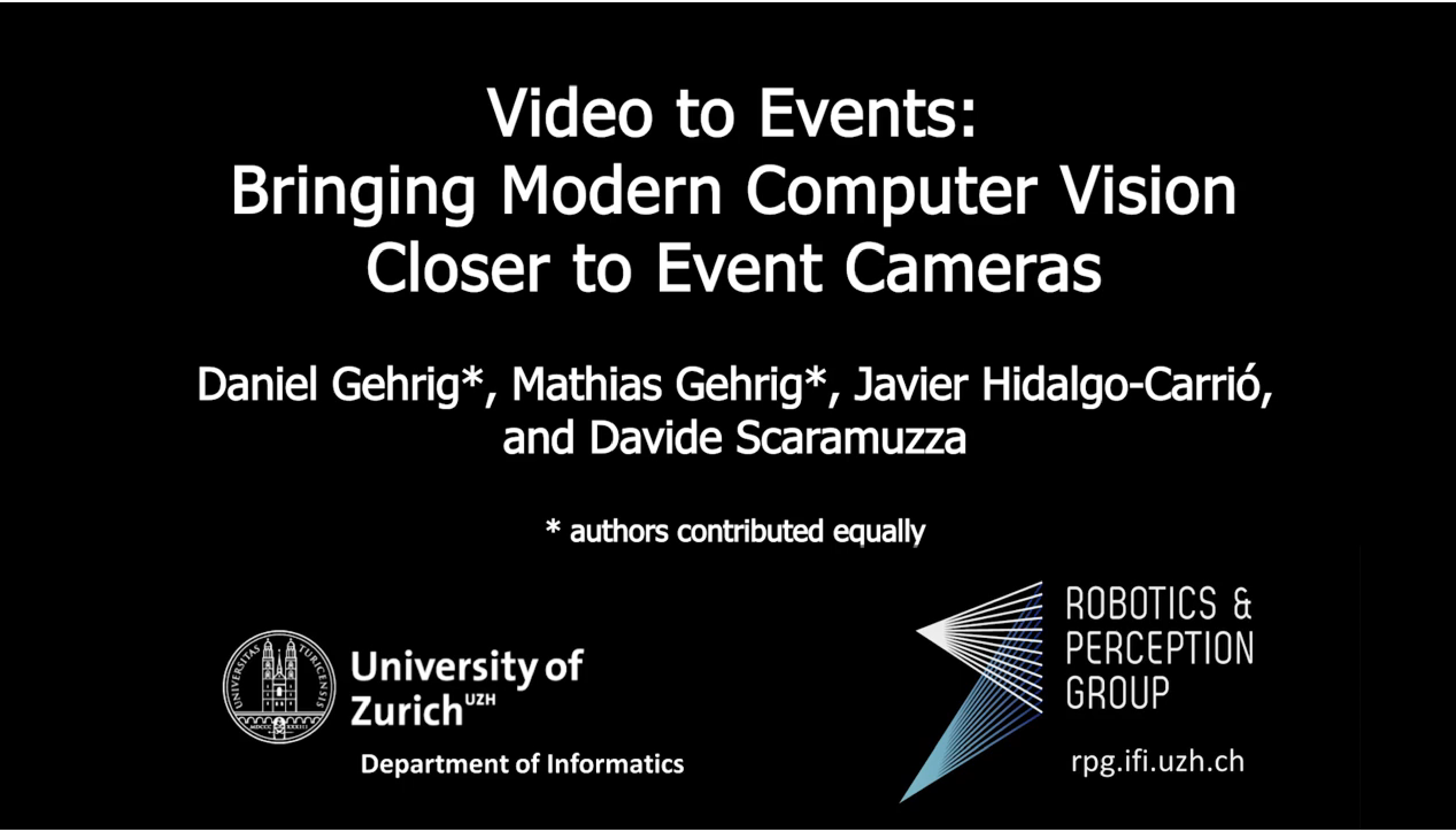
- **Aim:** To design a simulation framework to generate events, given high frame-rate images as inputs. A reference diagram is shown below:



- The principle of event camera and event data simulation will be discussed later in the first lecture.

1610.08336.pdf (arxiv.org)
(4) The Event-Camera Dataset and Simulator - YouTube

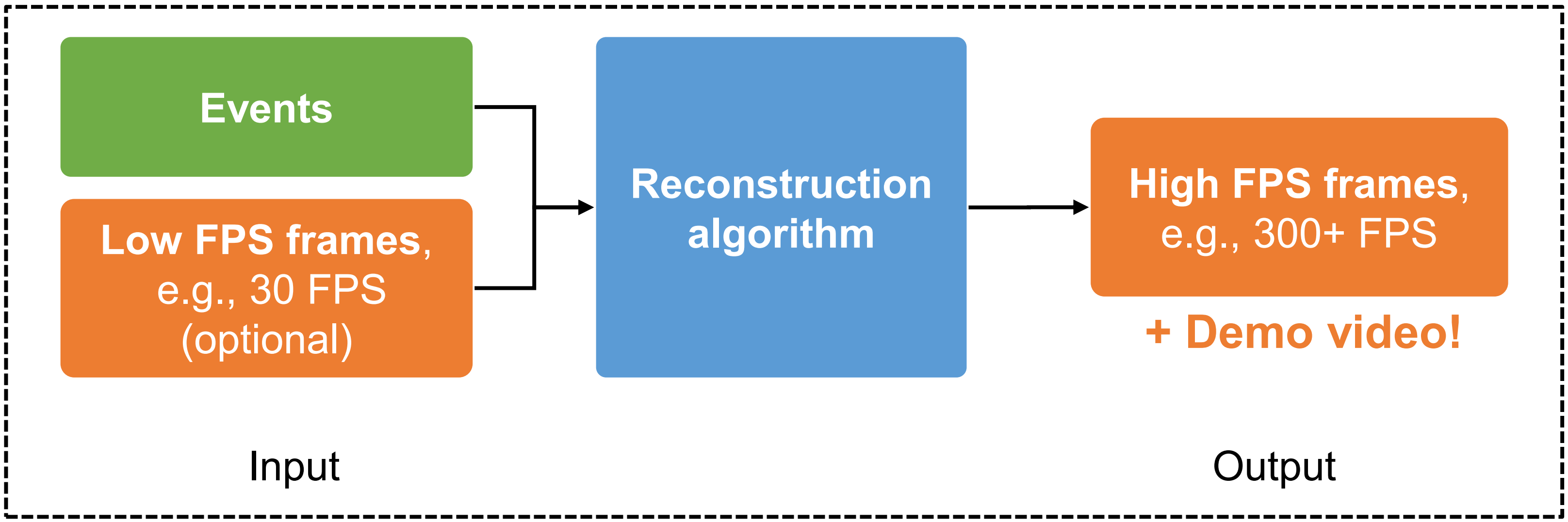
CA: Design an Event Data Simulator



[1912.03095] Video to Events: Recycling Video Datasets for Event Cameras (arxiv.org)
GitHub - uzh-rpg/rpg_vid2e: Open source implementation of CVPR 2020 "Video to Events: Recycling Video Dataset for Event Cameras"

RP: Event-based High Frame-Rate Video Reconstruction

- **Aim:** To design and implement an event-based high FPS video reconstruction framework. A reference diagram is shown below:



- Information hub for event-based vision: [uzh-rpg/event-based_vision_resources (github.com)]
- More information/Q&A session in the 2nd lecture, after introduction of all required backgrounds/knowledges.

RP: Event-based High Frame-Rate Video Reconstruction

https://www.youtube.com/watch?v=eomALySSGVUhttps://www.youtube.com/watch?v=eomALySSGVUhttps://www.youtube.com/watch?v=eomALySSGVU

Learning Objectives

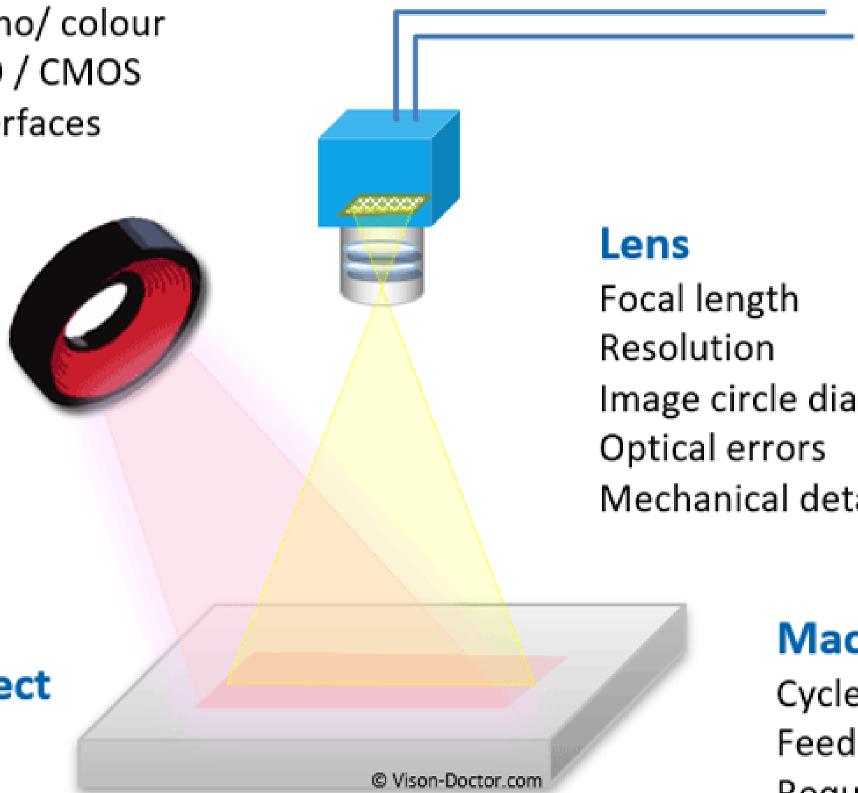
- In this lecture, you are expected to:

1. Describe and explain the basic concepts in both conventional and event-based machine vision systems;
2. Understand the principle, pros and cons, and current status of event cameras;
3. Understand the common procedure of event-based sensing, including data representation and data compression;
4. Understand the mechanism and principles of event data simulation.

Conventional Machine Vision Systems

Camera & sensor
Resolution
Sensor size
Mono/ colour
CCD / CMOS
Interfaces

Machine vision system
Software algorithms
Speed
Interfaces
Programming Interface
HMI



Illumination
Form factor
Working distance
Color of light
Direct/ diffuse light
Intensity
Alignment

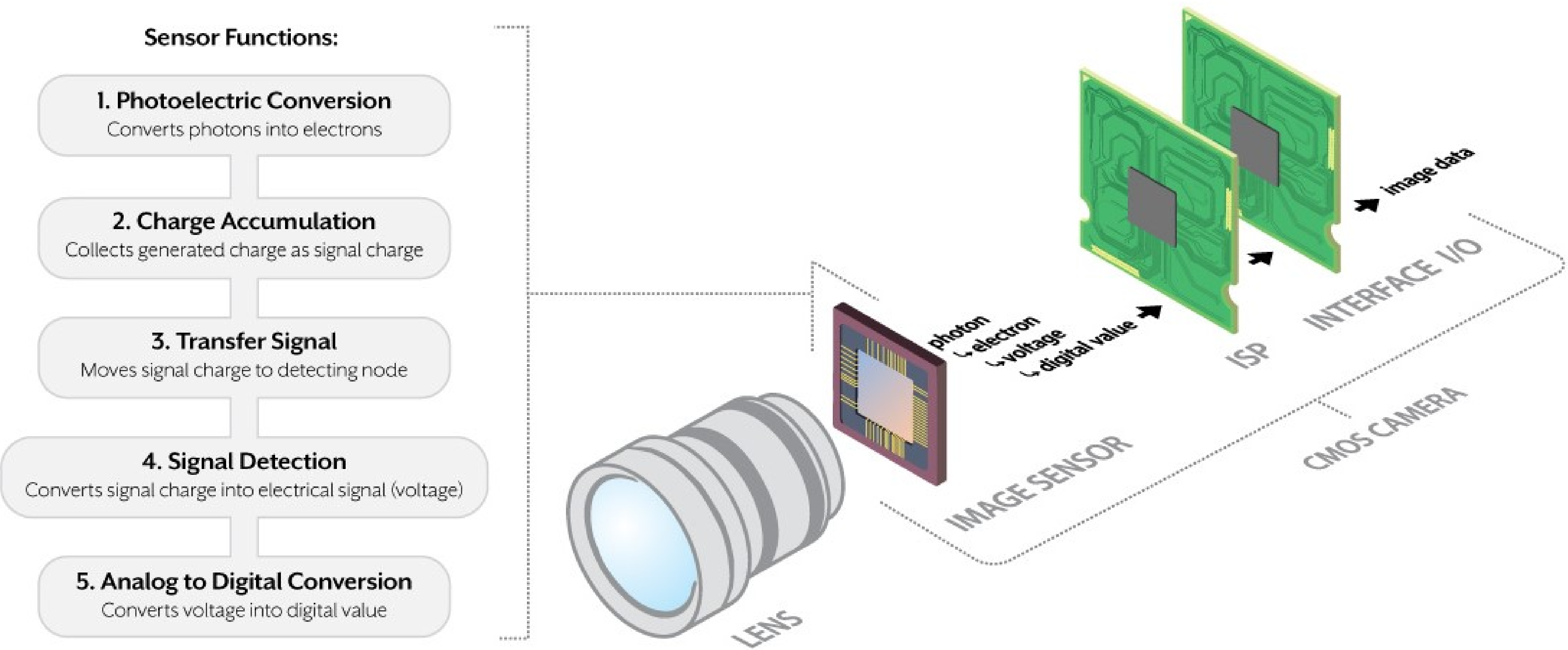
Lens
Focal length
Resolution
Image circle diameter
Optical errors
Mechanical details

Test object
Size
Geometry
Surface
Material
Color

Machine & environment
Cycle time
Feeding system
Required space
Environmental conditions
(Dirt, fog, vibration, daylight)
Interfaces (trigger, digital I/O)

Sensor Functions Inside a Camera

- Typical CMOS Camera Layout



Open Challenges in Computer Vision

The past 60 years of research have been devoted to frame-based cameras ...but they are not good enough!

Latency & Motion blur

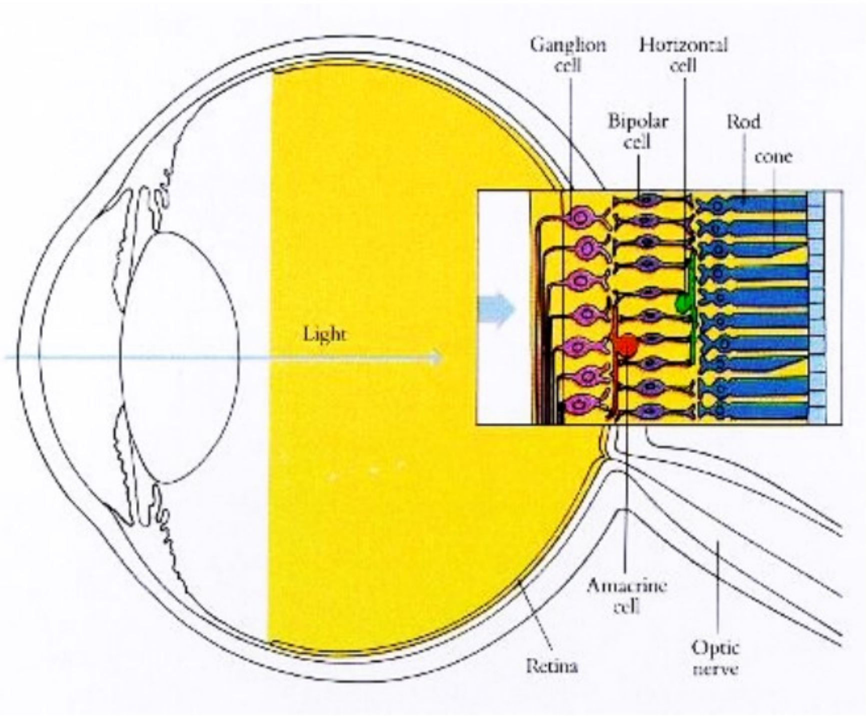
Dynamic Range

Event cameras do not suffer from these problems!

http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf

Inspiration

- Inspired by biological sensor – our eyes



100M photoreceptors

1M output fibers carrying max 100Hz spike rates

180dB (10^9) operating range

>20 different "eyes"

Many TOp/s computing

3mW power consumption

Output is a **sparse**, **asynchronous** stream of **digital spike events**

http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf
Event Cameras Tutorial - Delbruck - v3.0.pdf - Google Drive

What is an Event Camera?

- Novel sensor that measures only **motion in the scene**
- **First commercialized in 2008** by T. Delbruck (UZH&ETH) under the name of Dynamic Vision Sensor (DVS)
- **Low-latency** (~ 1 μs)
- **No motion blur**
- **High dynamic range** (140 dB instead of 60 dB)
- **Ultra-low power** (mean: 1mW vs 1W)

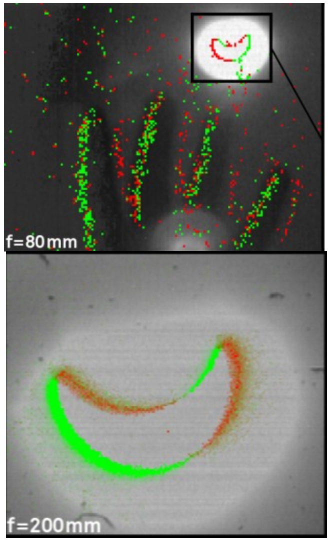
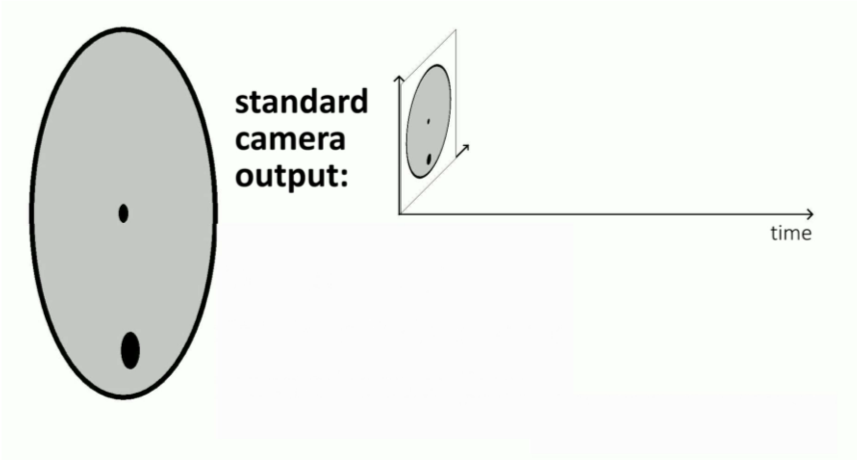


Image of the solar eclipse captured by a DVS

Traditional vision algorithms cannot be used because:
- **Asynchronous** pixels
- **No intensity information** (only binary intensity changes)

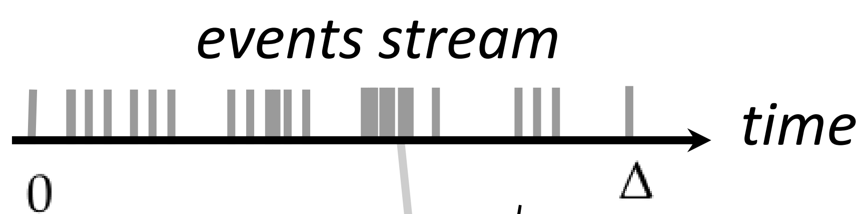


[Lichtsteiner, Posch, Delbruck, A 128x128 120 dB 15μs Latency Asynchronous Temporal Contrast Vision Sensor, 2008
http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf

Traditional Camera vs. Event Camera

- A traditional camera outputs frames at fixed time intervals:

- By contrast, a DVS outputs asynchronous events at microsecond resolution. An **event** is generated each time a single pixel detects an intensity changes value



$$\text{event:} \left\langle t, \langle x, y \rangle, sign\left(\frac{dI(x,y)}{dt}\right) \right\rangle$$

Timestamp (μs)

Pixel coordinates

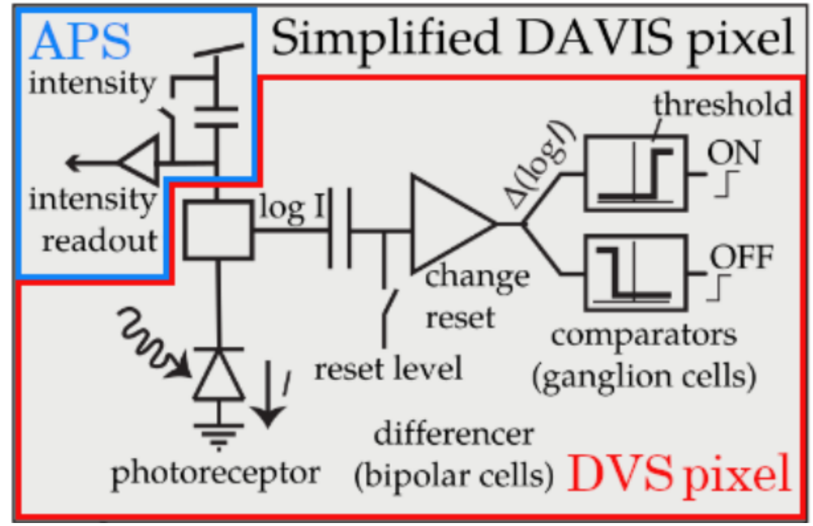
Event polarity (or sign) (-1 *or* 1): *increase or decrease of brightness*

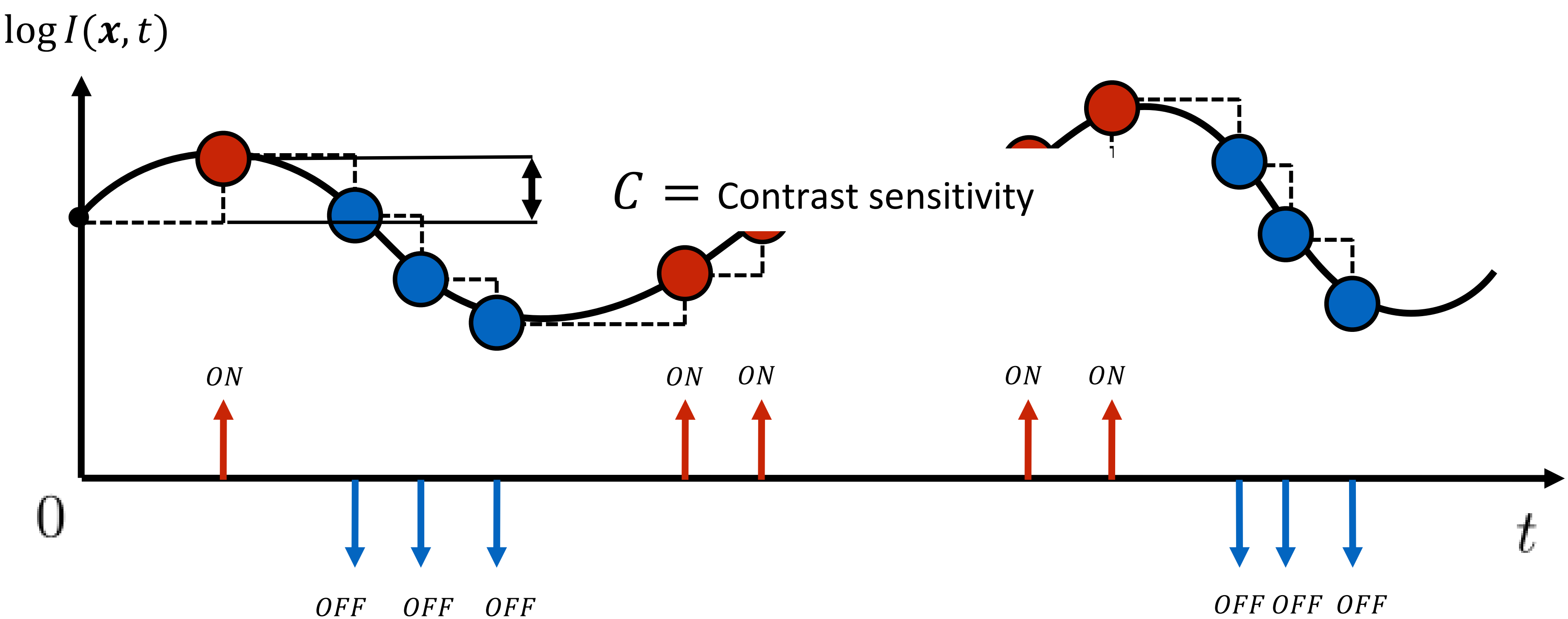
http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf

Generative Event Model

- Consider the intensity at a single pixel…

$$\pm C = \log I(\boldsymbol{x}, t) - \log I(\boldsymbol{x}, t - \Delta t)$$





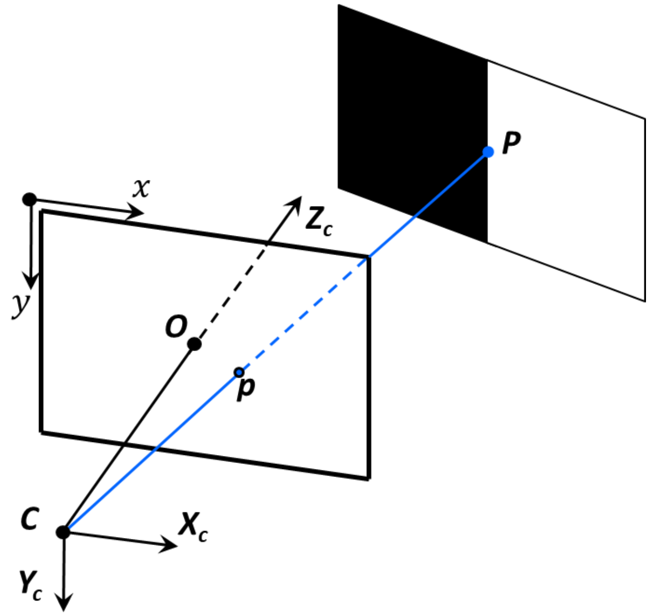
Events are triggered **asynchronously!**

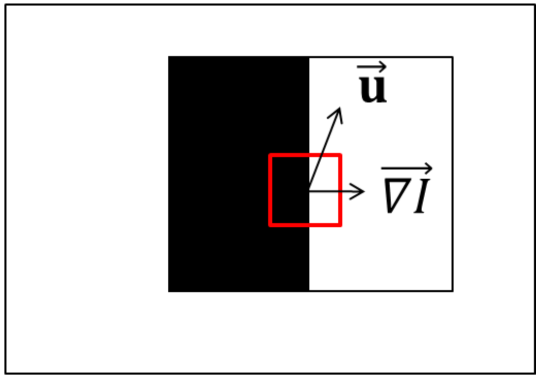
http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf

Generative Event Model: 1st Order Approximation

- Let us define $L(x,y,t)=Log(I(x,y,t))$
- Consider a given pixel $p(x,y)$ with gradient $\nabla L(x,y)$ undergoing the motion $\boldsymbol{u}=(u,v)$ in pixels, induced by a moving 3D point $\mathbf{P}$.
- Then, it can be shown that:

$$\boxed{-\nabla L \cdot \mathbf{u} = C}$$





http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf

Generative Event Model: 1st Order Approximation – Proof

- Recall the **Brightness constancy assumption,** which says: the intensity value of p, **before and after the motion**, must **remain unchanged**.

$$L(x, y, t) = L(x + u, y + v, t + \Delta t)$$

- By replacing the right-hand term by its 1st order approximation at $t+\Delta t$:

$$L(x, y, t) = L(x, y, t + \Delta t) + \frac{\partial L}{\partial x} u + \frac{\partial L}{\partial y} v$$

$$\Rightarrow L(x, y, t + \Delta t) - L(x, y, t) = -\frac{\partial L}{\partial x} u - \frac{\partial L}{\partial y} v$$

$$\Rightarrow \Delta \mathrm{L} = \boxed{C = -\nabla L \cdot \mathbf{u}}$$

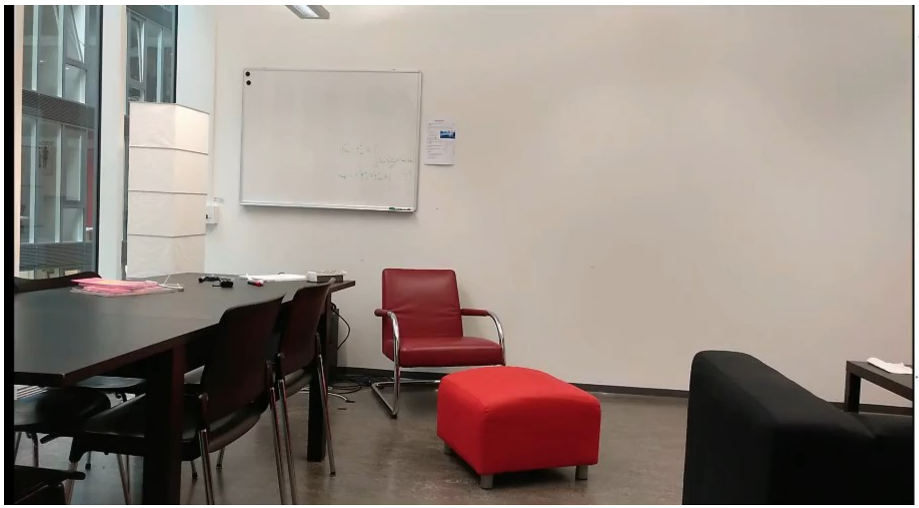
  - This equation describes the **linearized** event generation equation for an event generated by a gradient ∇L that moved by a motion vector $\mathbf{u}$ (optical flow) during a time interval Δt.

http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf

Event Camera Output With/Without Motion

- Without motion, only output is the background noise

Standard Camera



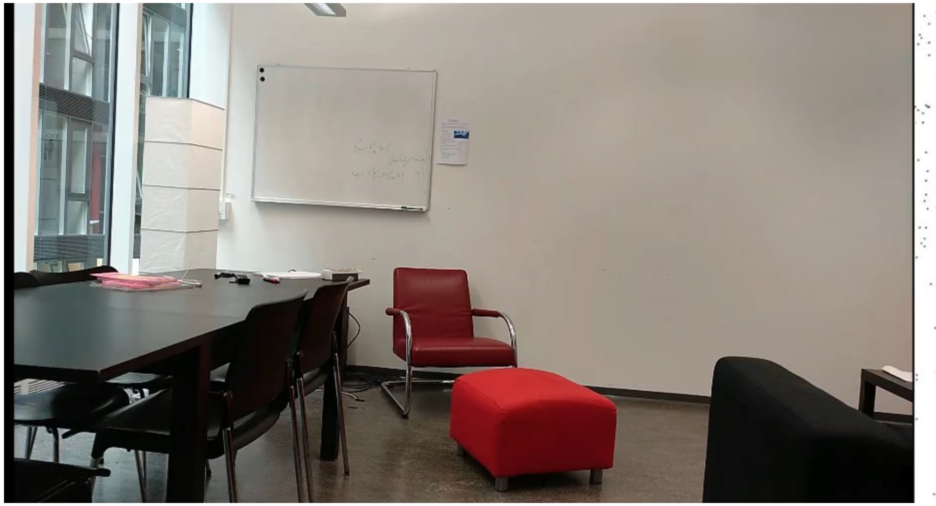
Event Camera (ON, OFF events)

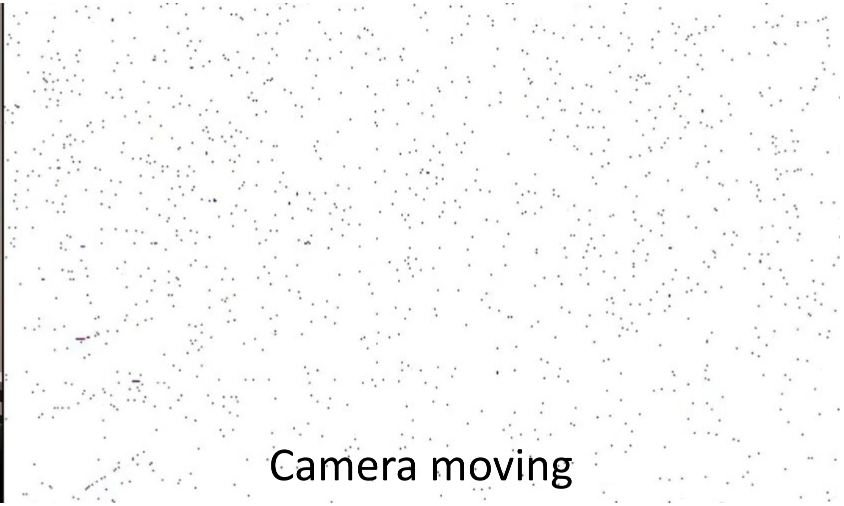
http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf

Event Camera Output With/Without Motion

Standard Camera

Event Camera (ON, OFF events)

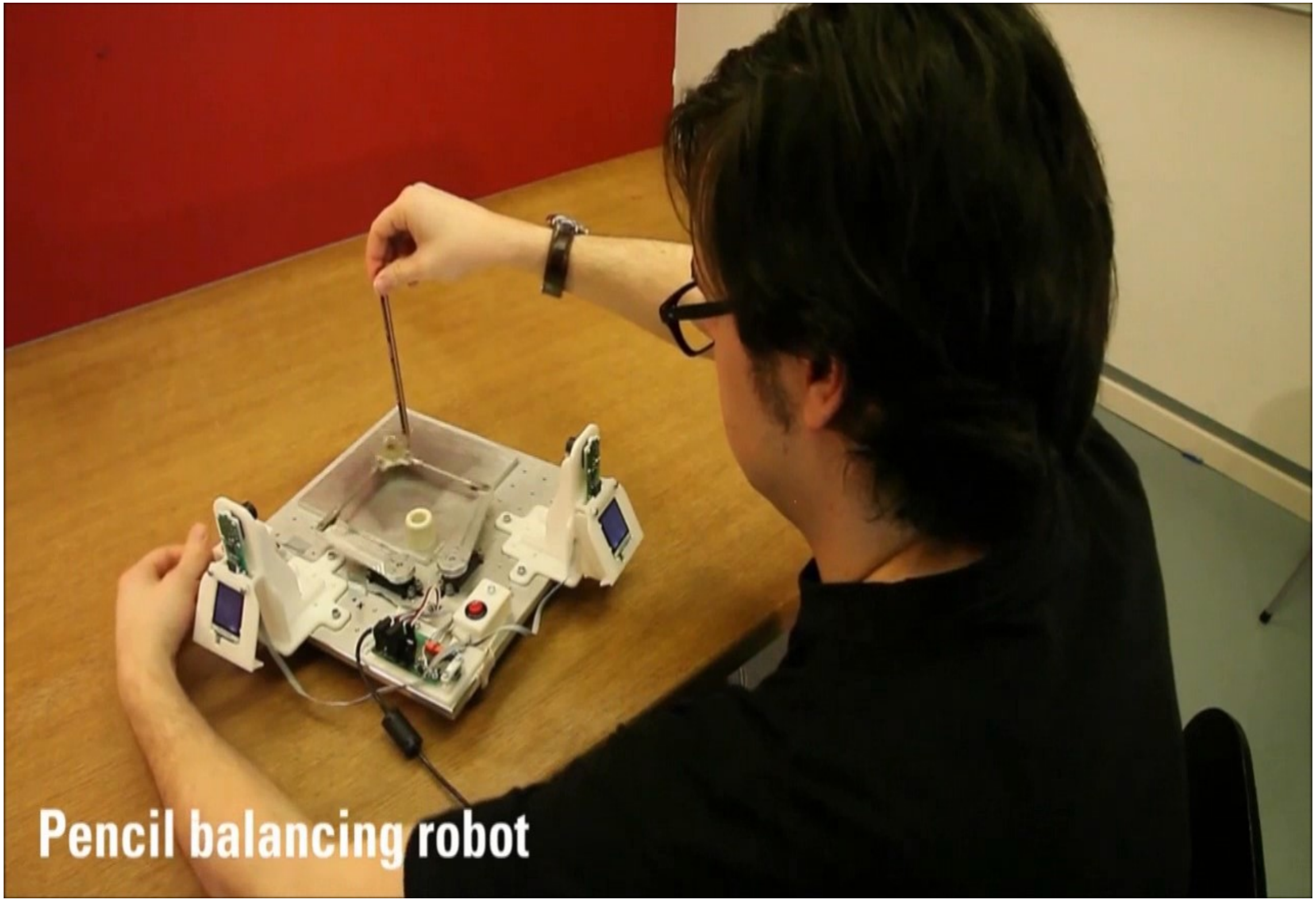




http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf

Examples: Inverted Pendulum

- Low latency, high-speed measurement on pencil angle using event cameras

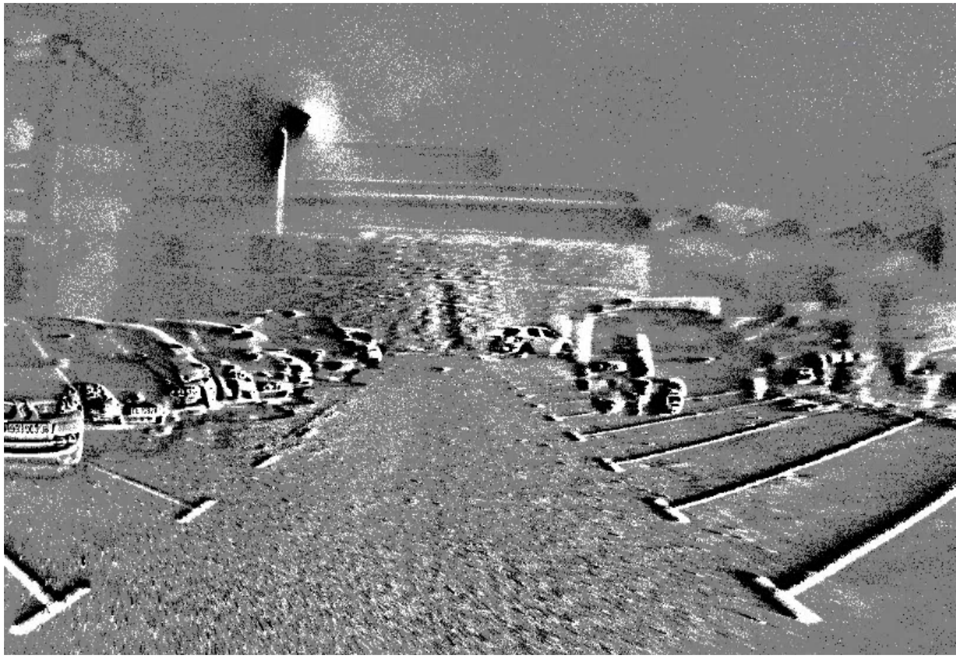


Conradt, Cook, Berner, Lichtsteiner, Douglas, Delbruck, **A pencil balancing robot using a pair of AER dynamic vision sensors**, IEEE International Symposium on Circuits and Systems. 2009

Low-light Sensitivity (Night Drive)

- Excellent night vision ability

GoPro Hero 6



Event Camera by Prophesee
White = Positive events
Black = Negative events

http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf
Video courtesy of Prophesee: https://www.prophesee.ai

High-speed vs Event Cameras

- Can I just use a high-speed camera?



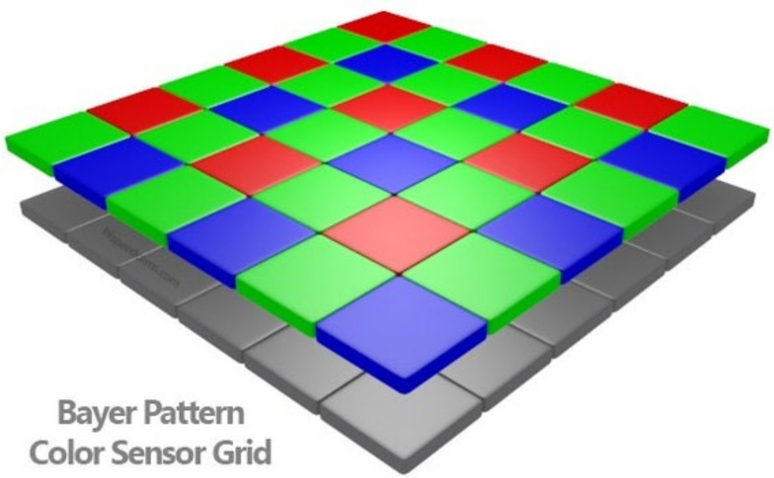
	High speed camera	Standard camera	Event Camera
Max fps or measurement rate	**Up to 1MHz**	100-1,000 fps	**1MHz**
Resolution at max fps	64x16 pixels	**>1Mpxl**	**>1Mpxl**
Bits per pixels (event)	12 bits	8-10 per pixel	**~40 bits/event** {t,(x,y),p)}
Weight	6.2 Kg	**30 g**	**30 g**
Active cooling	yes	**No cooling**	**No cooling**
Data rate	1.5 GB/s	32MB/s	**~1MB/s on average (depends on dynamics)**
Mean power consumption	150 W + external light	1 W	**1 mW**
Dynamic range	n.a.	60 dB	**140 dB**

http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf

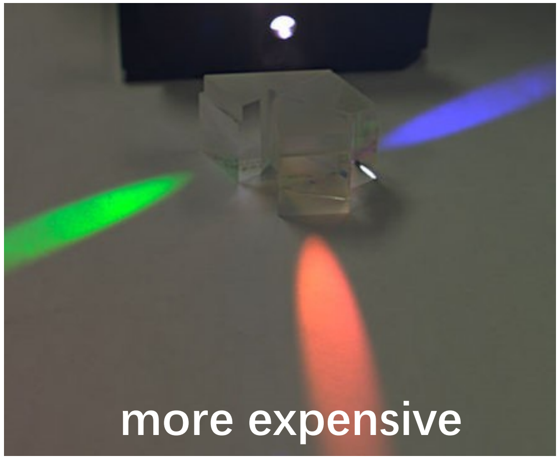
Color Event Cameras

- Two ways to build a color event camera
  - color filter
  - color-splitter prisms
- Few publications of practical color event cameras available
- Simulated data available for research

color filters



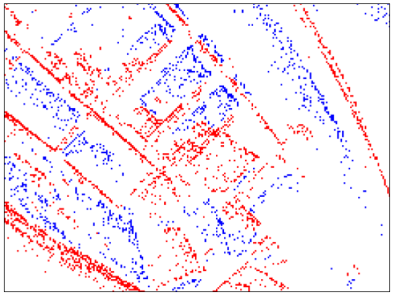
color-splitter prisms

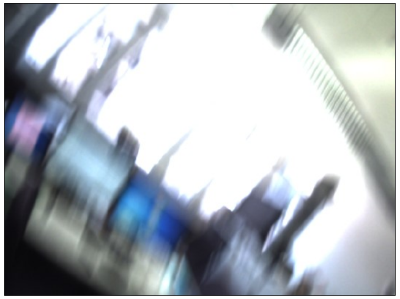


Reference page

Event Camera + Conventional Camera + …

- **Design idea:** utilize the complementary advantages of event and standard cameras, and other sensors.





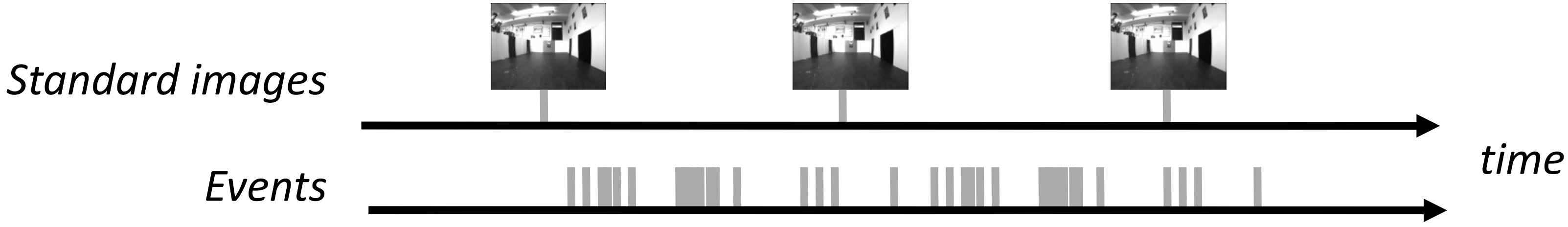
	Event Camera	**Standard Camera**
Update rate	High (asynchronous): 1 MHz	Low (synchronous)
Dynamic Range	High (140 dB)	Low (60 dB)
Motion Blur	No	Yes
Static motion	No (event camera is a high pass filter)	Yes
Absolute intensity	No (reconstructable up to a constant)	Yes
Maturity	< 10 years of research	> 60 years of research!

http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf

DAVIS Sensor: Events + Images + IMU

- Combines an event and a standard camera in the **same pixel array** (→ the same pixel can both trigger events and integrate light intensity).
- It also has an IMU.

Temporal aggregation of events overlaid on a DAVIS frame

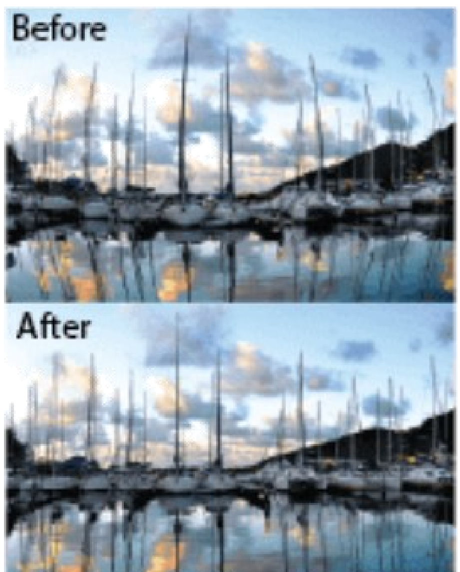


http://rpg.ifi.uzh.ch/docs/scaramuzza/Tutorial_on_Event_Cameras_Scaramuzza.pdf

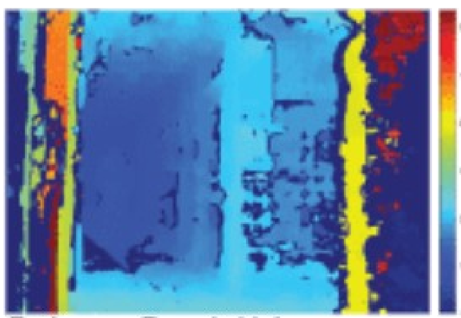
Event Camera Calibration

- Why need camera calibration?
  - To estimate the **parameters** of a lens and image sensor.

- With a calibrated camera, you can:

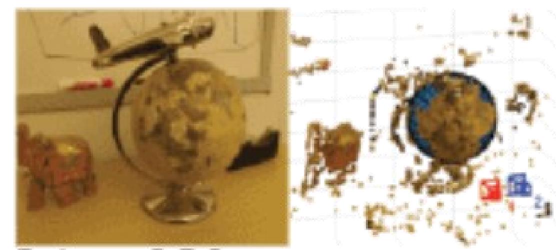


Remove Lens Distortion



Estimate Depth Using a Stereo Camera

Measure Planar Objects

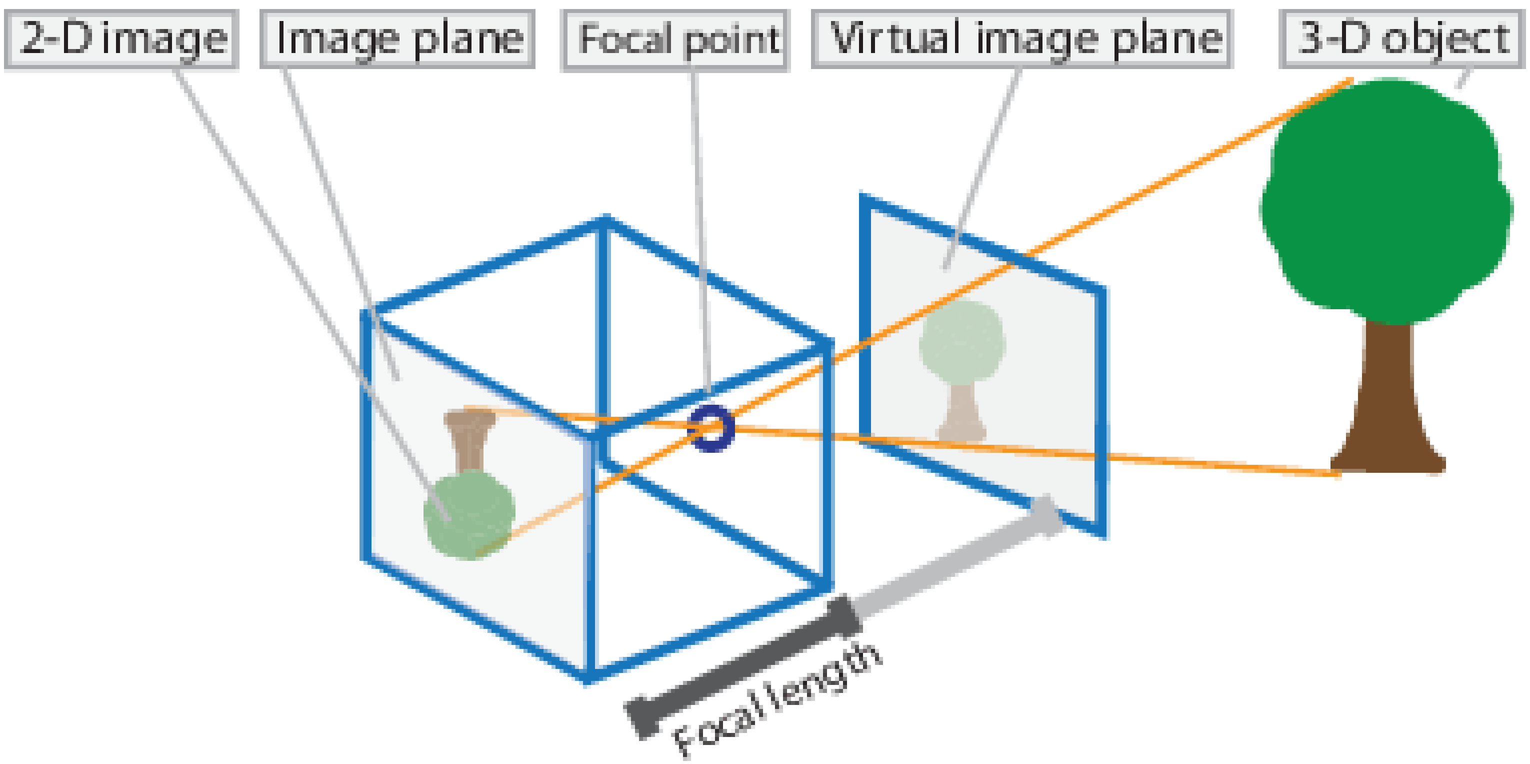


Estimate 3-D Structure from Camera Motion

[2021CVPRW_How_to_Calibrate_Your_Event_Camera.pdf (tub-rip.github.io)](2021CVPRW_How_to_Calibrate_Your_Event_Camera.pdf)
[What Is Camera Calibration? - MATLAB & Simulink (mathworks.com)](mathworks.com)

Camera Models

- Pinhole camera model
  - A simple camera without a lens and with a single small aperture



What Is Camera Calibration? - MATLAB & Simulink (mathworks.com)

Camera Models

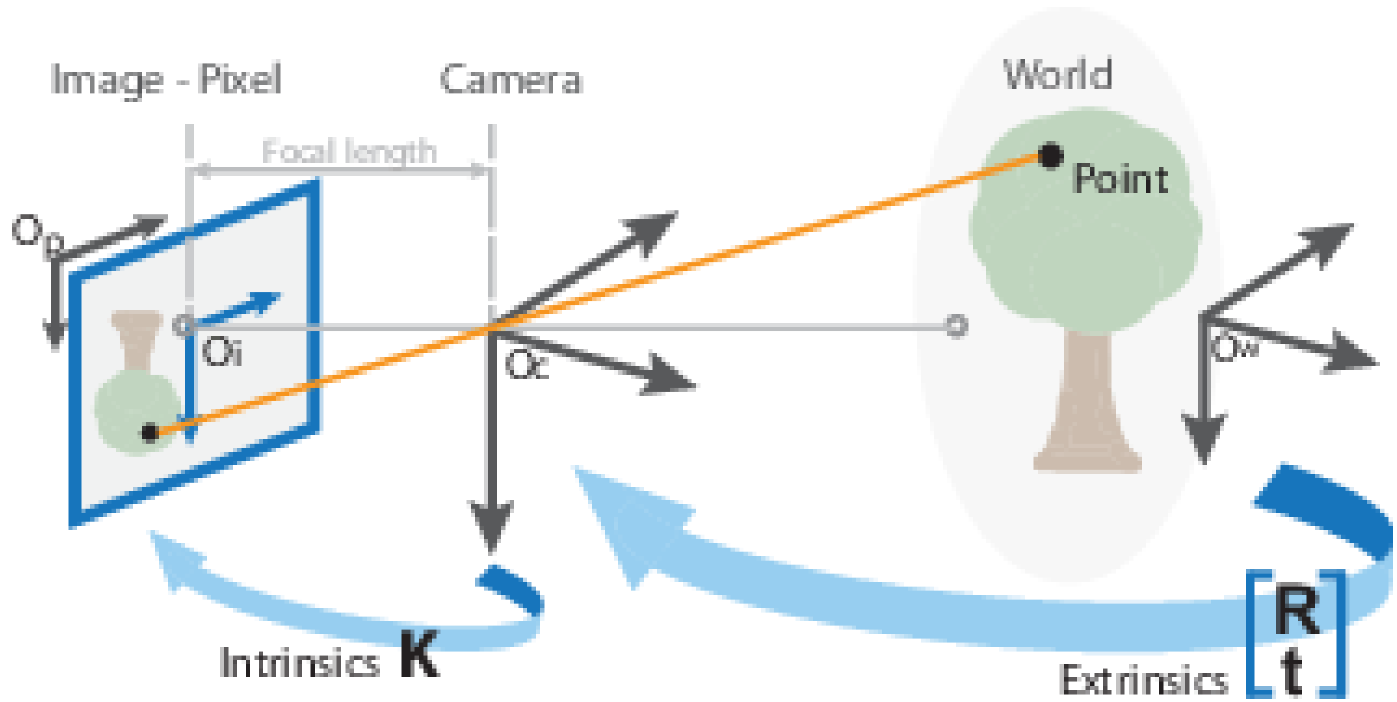
- Pinhole camera model
  - Parameters are included in a 4-by-3 **camera matrix: 3-D → 2-D**

$$\underset{\text{Scale factor}}{w}\ \underbrace{[x\ y\ 1]}_{\text{Image points}} = \underbrace{[X\ Y\ Z\ 1]}_{\text{World points}}\ \boxed{P}$$

$$\underset{\text{Camera matrix}}{\boxed{P}} = \underset{\substack{\text{Extrinsics}\\ \text{Rotation and translation}}}{\begin{bmatrix} R \\ t \end{bmatrix}}\ \underset{\text{Intrinsic matrix}}{K}$$

[What Is Camera Calibration? - MATLAB & Simulink (mathworks.com)]

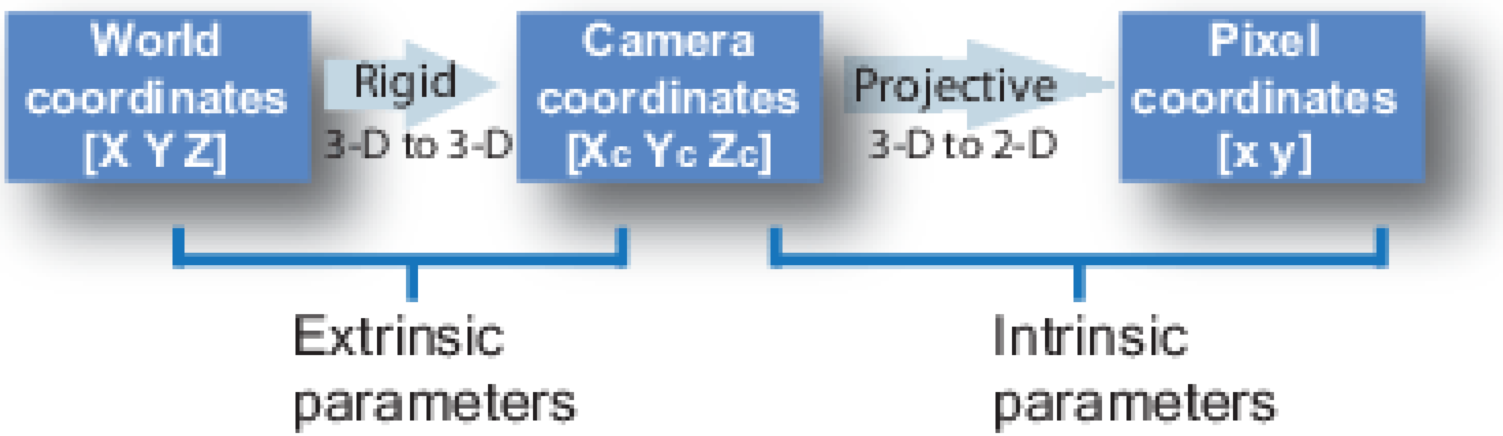
Camera Models



What Is Camera Calibration? - MATLAB & Simulink (mathworks.com)

Camera Models

- Pinhole camera model
  - Extrinsics parameters**: world points → camera coordinates**
  - Intrinsics parameters: **camera coordinates → image plane**



With extrinsics and intrinsics parameters, we could obtain the **camera matrix**!

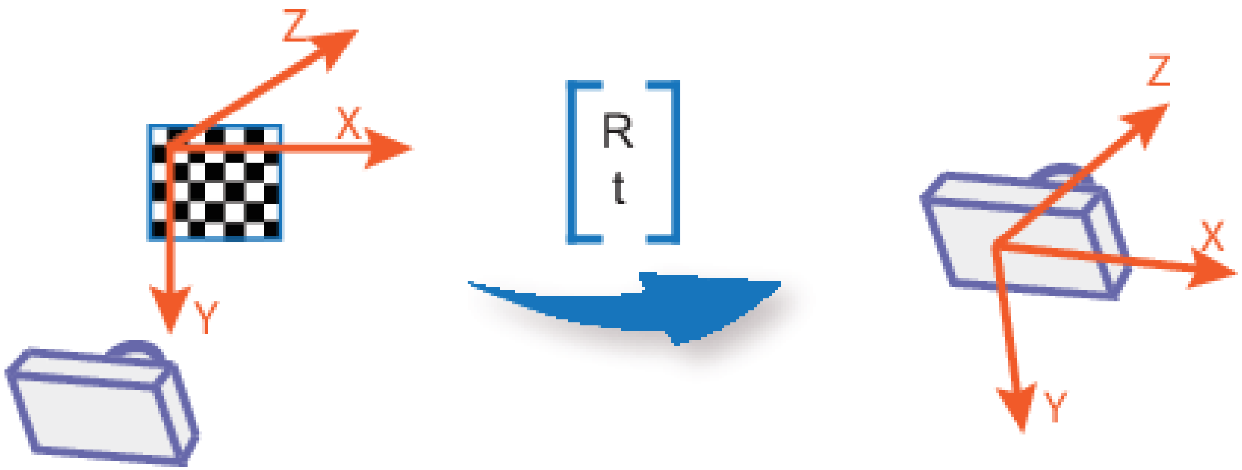
What Is Camera Calibration? - MATLAB & Simulink (mathworks.com)

Camera Models

- Extrinsic parameters
  - consist of a **rotation**, **R**, and a **translation**, **t**.

  R: rotation matrix

  t: translation vector



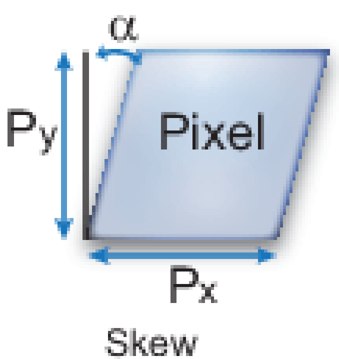
What Is Camera Calibration? - MATLAB & Simulink (mathworks.com)

Camera Models

- Intrinsic parameters
  - include: the **focal length**, the **optical center**, also known as the **principal point**, and the **skew coefficient**

$$\begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$\begin{bmatrix} c_x & c_y \end{bmatrix}$ – Optical center (the principal point), in pixels.
$(f_x,\ f_y)$ – Focal length in pixels. $f_x = F/p_x$ $f_y = F/p_y$ F – Focal length in world units, typically expressed in millimeters. $(p_x,\ p_y)$ – Size of the pixel in world units.
s – Skew coefficient, which is non-zero if the image axes are not perpendicular. $s = f_x \tan\alpha$

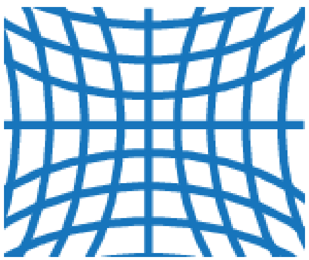


What Is Camera Calibration? - MATLAB & Simulink (mathworks.com)

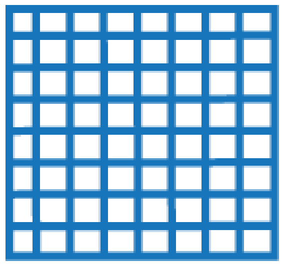
Distortion in Camera Calibration

Lens distortion requires additional correction.

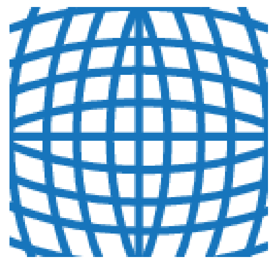
- **Radial Distortion:** Light rays bend more near the edges of a lens than they do at its optical center.



Negative radial distortion "pincushion"



No distortion



Positive radial distortion "barrel"

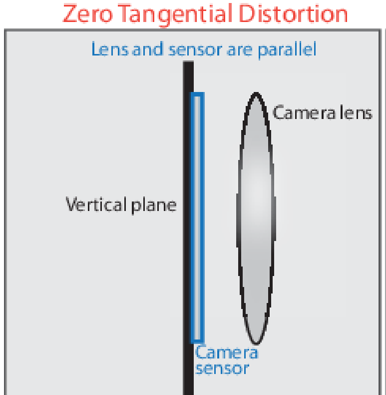
$$x_{\text{distorted}} = x(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6)$$

$$y_{\text{distorted}} = y(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6)$$

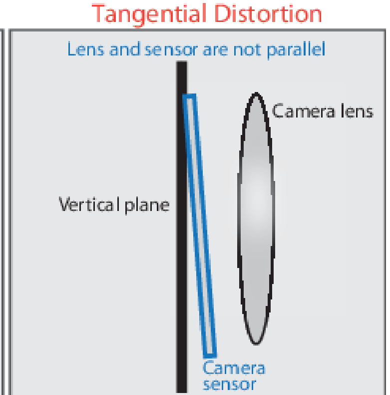
radial distortion coefficients

- **Tangential Distortion**: when the lens and the image plane are not parallel

Zero Tangential Distortion



Tangential Distortion



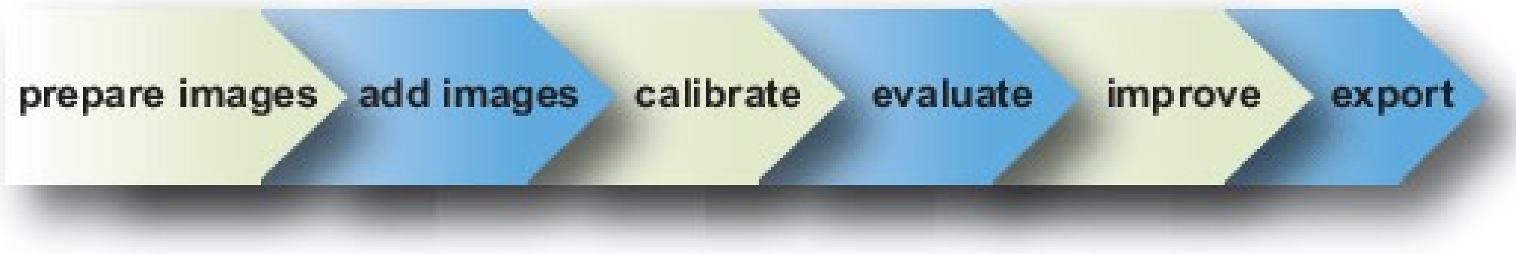
$$x_{\text{distorted}} = x + [2 * p_1 * x * y + p_2 * (r^2 + 2 * x^2)]$$

$$y_{\text{distorted}} = y + [p_1 * (r^2 + 2 * y^2) + 2 * p_2 * x * y]$$

tangential distortion coefficients

What Is Camera Calibration? - MATLAB & Simulink (mathworks.com)

Calibration Process

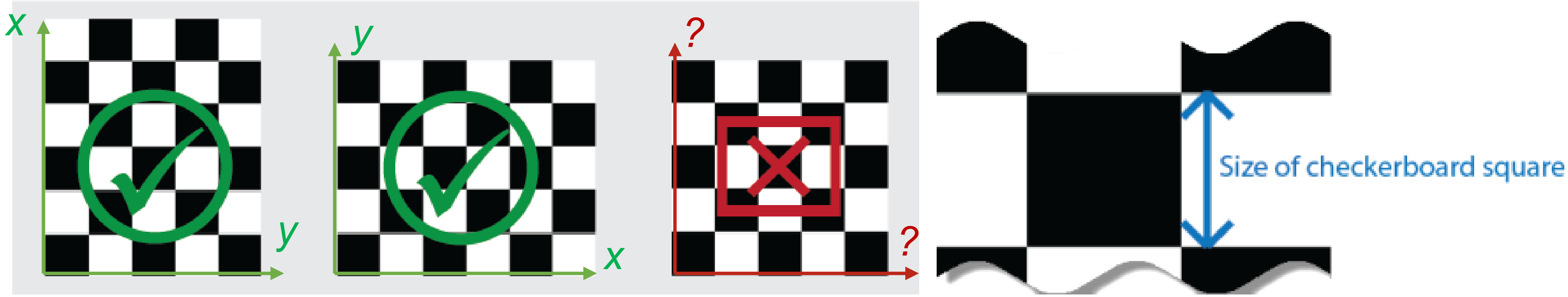


1. Prepare the pattern, camera, and images (by taking pictures)
2. Add images to the calibrator and select standard or fisheye camera model
3. Calibrate the camera and obtain the parameters
4. Evaluate calibration accuracy
5. Adjust parameters to improve accuracy
6. Export the parameters and complete the calibration

[Single Camera Calibrator App - MATLAB & Simulink (mathworks.com)]

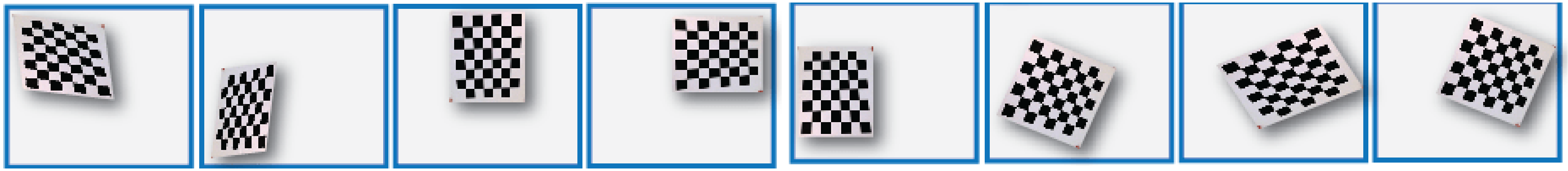
Prepare Images

- Print the checkerboard pattern and attach it to a flat surface.
- Measure the size of the checkerboard square for future use.



Even number of squares along one edge and
an odd number of squares along the other edge

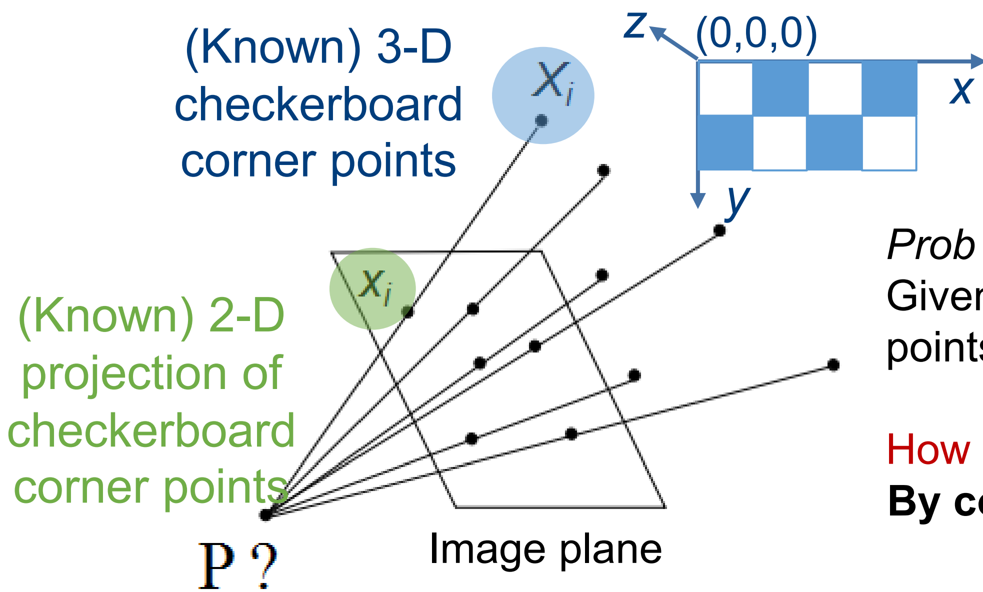
- Capture 10 and 20 images of the calibration pattern. Do not compress the image.



[Single Camera Calibrator App - MATLAB & Simulink (mathworks.com)](mathworks.com)

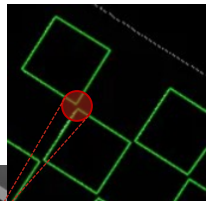
Calibrate

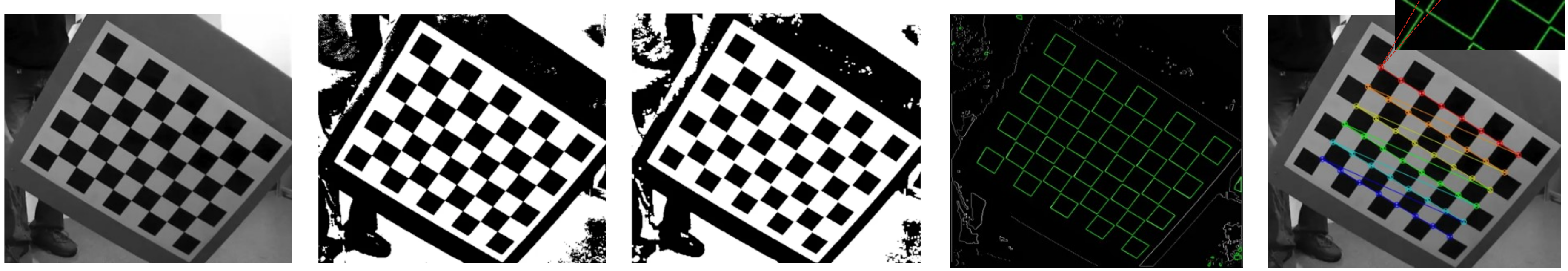
1. Solve for the intrinsics and extrinsics in closed form, assuming that lens distortion is zero.



Prob Formulation:
Given the correspondence between 3-D and 2-D points, find the camera matrix P.

How to find the correspondence?
By corner detection!





raw image | binarization | dilation | quadrilateral detection | result

Calibrate

With the correspondence, we have the **closed-form solution**. Details omitted, please refer to the paper:

Zhang, Zhengyou. "A flexible new technique for camera calibration." *IEEE Transactions on pattern analysis and machine intelligence* 22.11 (2000): 1330-1334.

A Flexible New Technique for Camera Calibration

Zhengyou Zhang

December 2, 1998
(updated on December 14, 1998)
(updated on March 25, 1999)
(updated on Aug. 10, 2002; a typo in Appendix B)
(last updated on Aug. 13, 2008; a typo in Section 3.3)

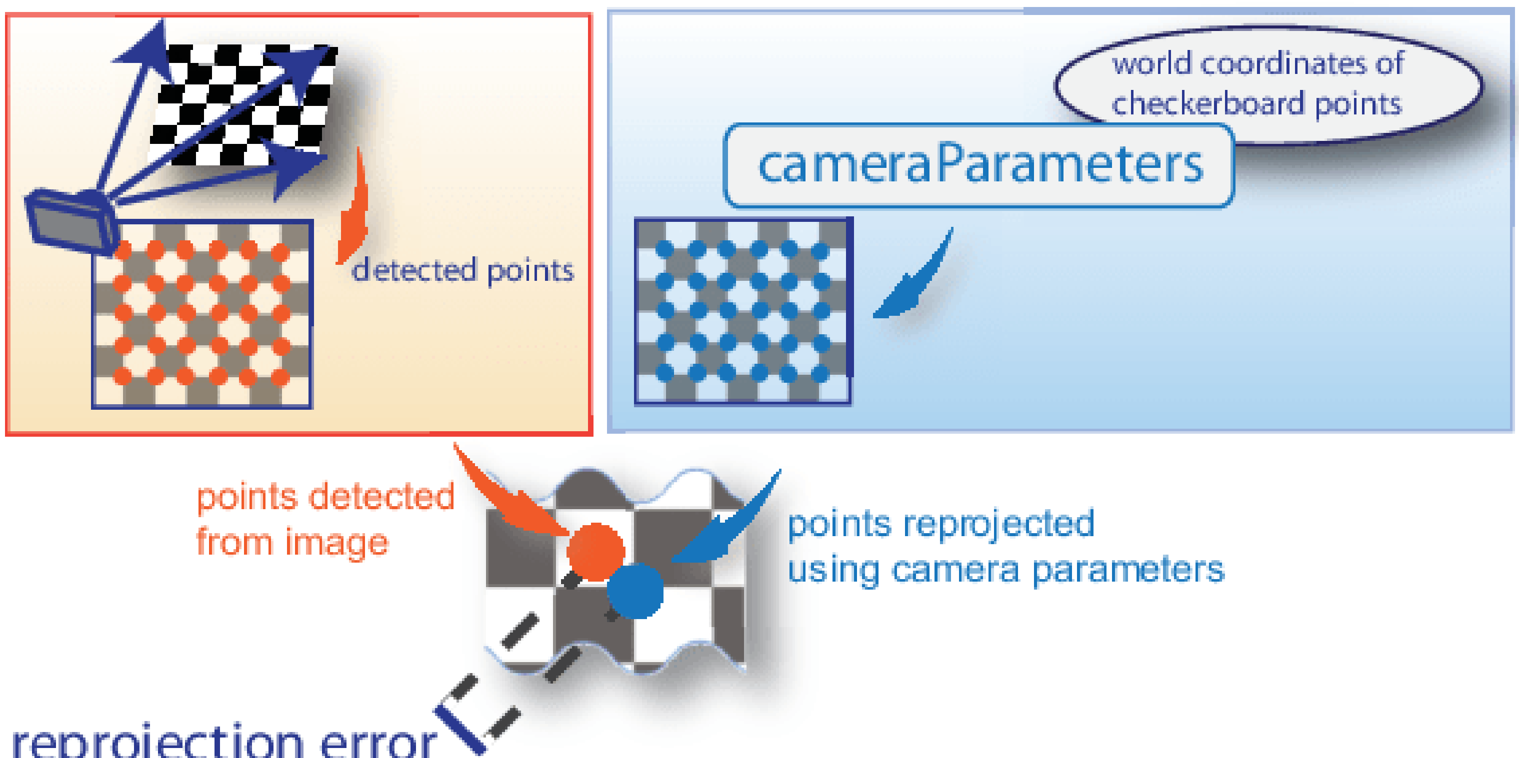
Technical Report
MSR-TR-98-71

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

[tr98-71.pdf (microsoft.com)]

Calibrate

2. Refine above results by minimizing the reprojection error. Estimate all parameters simultaneously, including the distortion coefficients, using **nonlinear least-squares minimization** (Levenberg–Marquardt algorithm).

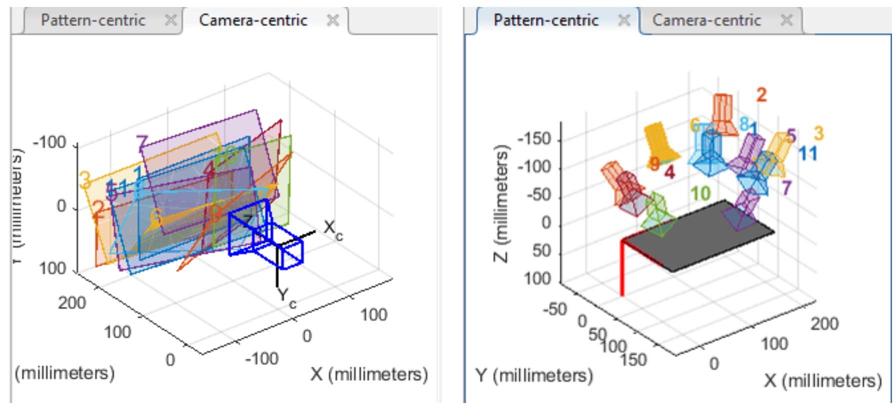


[Single Camera Calibrator App - MATLAB & Simulink (mathworks.com)]

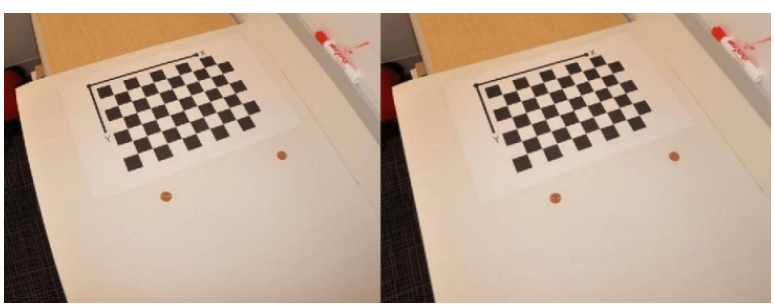
Evaluate Calibration Results

- Check the Reprojection Errors Bar Graph
- Check the Extrinsic Parameter Visualization
- Check undistorted image

Reprojection Errors Bar Graph



Extrinsic Parameter Visualization



Correct undistorted image

Incorrect undistorted image

[Single Camera Calibrator App - MATLAB & Simulink (mathworks.com)]

Improve Calibration

- Add or Remove Images
  - Add images if there is <10 images, the pattern do not cover the whole frame, or the pattern do not have enough variation in orientation w.r.t. the camera.
  - Remove those images with a high mean reprojection error, blurred images, or images with incorrectly detected checkerboard points.
- Change the number of radial distortion coefficients
  - Typically, two coefficients are sufficient for calibration. For severe distortion, such as in wide-angle lenses, you can select 3 coefficients to include k_3.
- Any other possible ways to improve calibration:
  - Compute the skew parameter
  - Compute tangential distortion

A complex camera model may not be always better!

Single Camera Calibrator App - MATLAB & Simulink (mathworks.com)

Event Camera Calibration

- What makes it different for **event camera calibration**?
  - Camera model? Lens distortion?
  - **Event camera output nothing when there is no motion…**

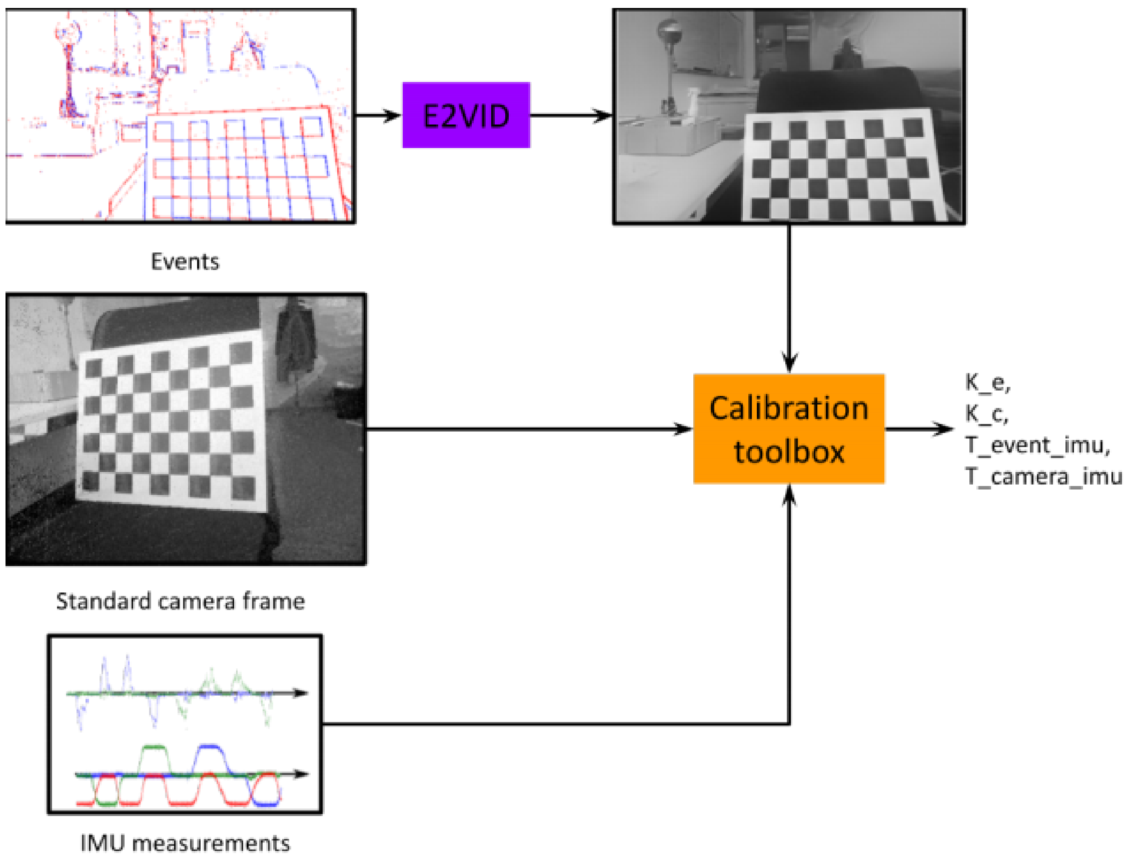
- Ideas
  - Design a calibration pattern that constantly triggers events.
  - Reconstruct events into frames, such that standard calibration patterns can be used.

Designing a Calibration Pattern That Constantly Triggers Events

Event-Based Camera Calibration - YouTube

Calibration by Reconstructing Events into Frames

- Intrinsic calibration for event camera
- Multi-sensor extrinsics calibration (conventional camera, event camera, and IMU)



2021CVPRW_How_to_Calibrate_Your_Event_Camera.pdf (tub-rip.github.io)

Event Camera Manufactures and Products

	Supplier	iniVation			Prophesee				Samsung			CelePixel		Insightness
	Camera model	DVS128	DAVIS240	DAVIS346	ATIS	Gen3 CD	Gen3 ATIS	Gen 4 CD	DVS-Gen2	DVS-Gen3	DVS-Gen4	CeleX-IV	CeleX-V	Rino 3
Sensor specifications	Year, Reference	2008 [2]	2014 [4]	2017	2011 [3]	2017 [67]	2017 [67]	2020 [68]	2017 [5]	2018 [69]	2020 [39]	2017 [70]	2019 [71]	2018 [72]
	Resolution (pixels)	128 × 128	240 × 180	346 × 260	304 × 240	640 × 480	480 × 360	1280 × 720	640 × 480	640 × 480	1280 × 960	768 × 640	1280 × 800	320 × 262
	Latency (μs)	12μs @ 1klux	12μs @ 1klux	20	3	40 - 200	40 - 200	20 - 150	65 - 410	50	150	10	8	125μs @ 10lux
	Dynamic range (dB)	120	120	120	143	> 120	> 120	> 124	90	90	100	90	120	> 100
	Min. contrast sensitivity (%)	17	11	14.3 - 22.5	13	12	12	11	9	15	20	30	10	15
	Power consumption (mW)	23	5 - 14	10 - 170	50 - 175	36 - 95	25 - 87	32 - 84	27 - 50	40	130	-	400	20-70
	Chip size (mm^2)	6.3 × 6	5 × 5	8 × 6	9.9 × 8.2	9.6 × 7.2	9.6 × 7.2	6.22 × 3.5	8 × 5.8	8 × 5.8	8.4 × 7.6	15.5 × 15.8	14.3 × 11.6	5.3 × 5.3
	Pixel size (μm^2)	40 × 40	18.5 × 18.5	18.5 × 18.5	30 × 30	15 × 15	20 × 20	4.86 × 4.86	9 × 9	9 × 9	4.95 × 4.95	18 × 18	9.8 × 9.8	13 × 13
	Fill factor (%)	8.1	22	22	20	25	20	> 77	11	12	22	8.5	8	22
	Supply voltage (V)	3.3	1.8 & 3.3	1.8 & 3.3	1.8 & 3.3	1.8	1.8	1.1 & 2.5	1.2 & 2.8	1.2 & 2.8		1.8 & 3.3	1.2 & 2.5	1.8 & 3.3
	Stationary noise (ev/pix/s) at 25C	0.05	0.1	0.1	-	0.1	0.1	0.1	0.03	0.03		0.15	0.2	0.1
	CMOS technology (nm)	350 2P4M	180 1P6M MIM	180 1P6M MIM	180 1P6M	180 1P6M CIS	180 1P6M CIS	90 BI CIS	90 1P5M BSI	90	65/28	180 1P6M CIS	65 CIS	180 1P6M CIS
	Grayscale output	no	yes	yes	yes	no	yes	no	no	no	no	yes	yes	yes
	Grayscale dynamic range (dB)	NA	55	56.7	130	NA	> 100	NA	NA	NA	NA	90	120	50
	Max. frame rate (fps)	NA	35	40	NA	NA	NA	NA	NA	NA	NA	50	100	30
Camera	Max. Bandwidth (Meps)	1	12	12	-	66	66	1066	300	600	1200	200	140	20
	Interface	USB 2	USB 2	USB 3		USB 3	USB 3	USB 3	USB 2	USB 3	USB 3			USB 2
	IMU output	no	1 kHz	1 kHz	no	1 kHz	1 kHz	no	no	1 kHz	no	no	no	1 kHz

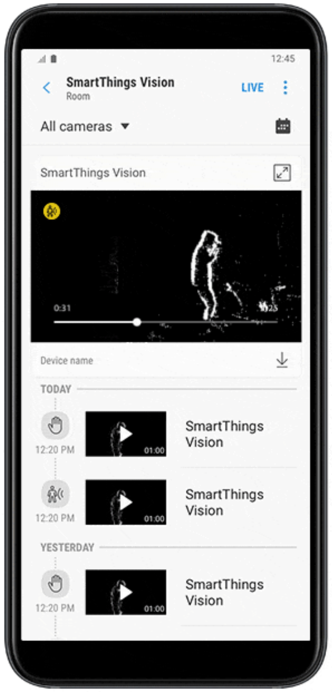
Event-based Vision: A Survey (arxiv.org)

Event Camera Manufactures and Products

- SmartThings Vision by Samsung

"Security that respects your privacy"

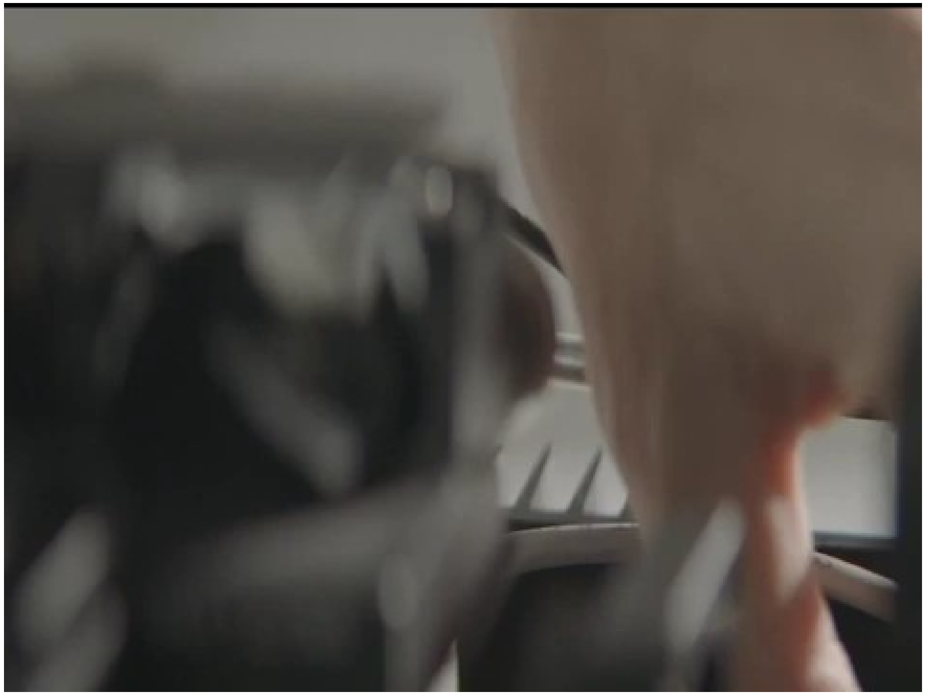
Unexpected person or intruder warning	while helps protects the privacy

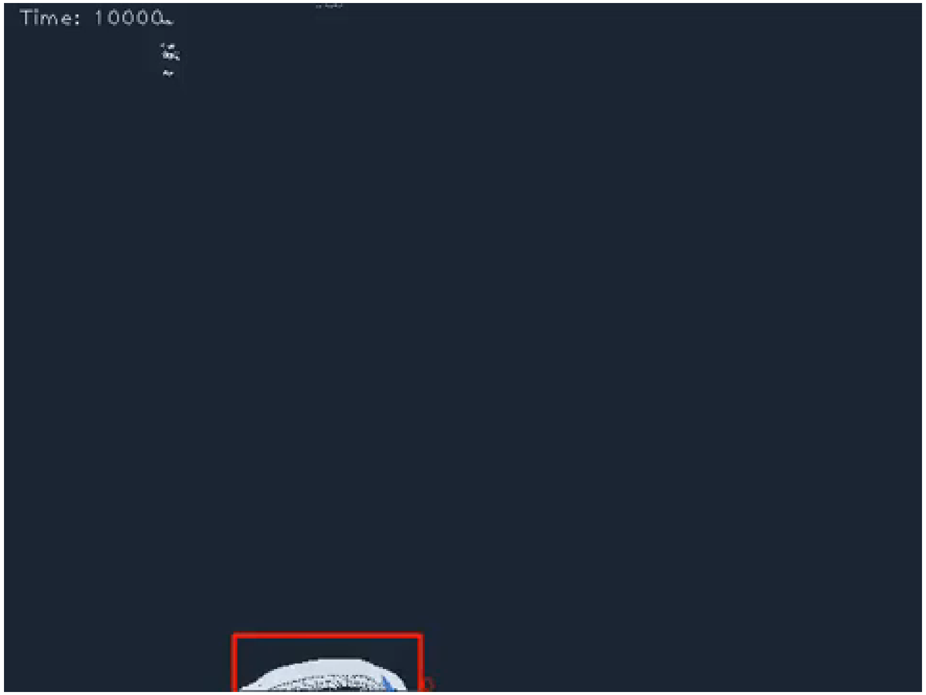


SmartThings Vision | GP-U999GTEEAAC | Samsung Australia

Event Camera Manufactures and Products

- Industrial solutions by Phophesee
  - ULTRA SLOW MOTION - Up to 200,000 fps (time resolution equivalent)
  - OBJECT TRACKING - Continuous tracking in time, no blind spots between frames





Event-Based Vision Applications | PROPHESEE

Event Camera Manufactures and Products

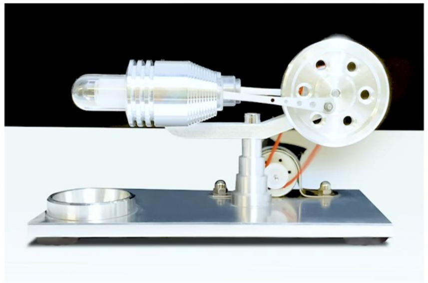
- Industrial solutions by Phophesee
  - OPTICAL FLOW – intrinsic from the sensor, much less power consumption
  - HIGH-SPEED COUNTING – Throughput >1,000 Obj/s Accuracy >99.5% @1,000 Obj/s.

PROPHESEE
METAVISION FOR MACHINES

Event-Based Vision Applications | PROPHESEE

Event Camera Manufactures and Products

- Industrial, IoT, and robotics solutions by Phophesee
  - VIBRATION MONITORING - From 1Hz to kHz range
  - VISUAL ODOMETRY – under fast motion

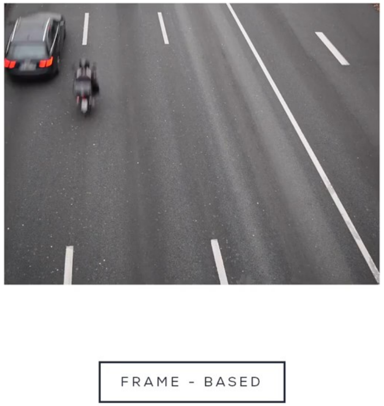


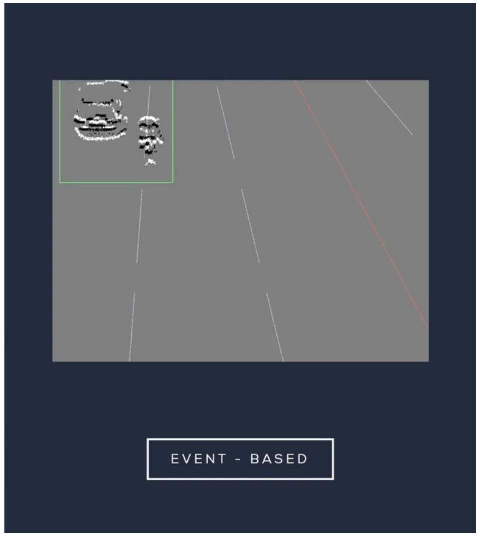
PROPHESEE
METAVISION FOR MACHINES

Event-Based Vision Applications | PROPHESEE

Event Camera Manufactures and Products

- Industrial, IoT, and robotics solutions by Phophesee
  - TRAFFIC DATA ACQUISITION – vehicle speed measurement
  - GESTURE DETECTION – for human-machine interaction





Event-Based Vision Applications | PROPHESEE

Summary – Event-based Sensors and Cameras

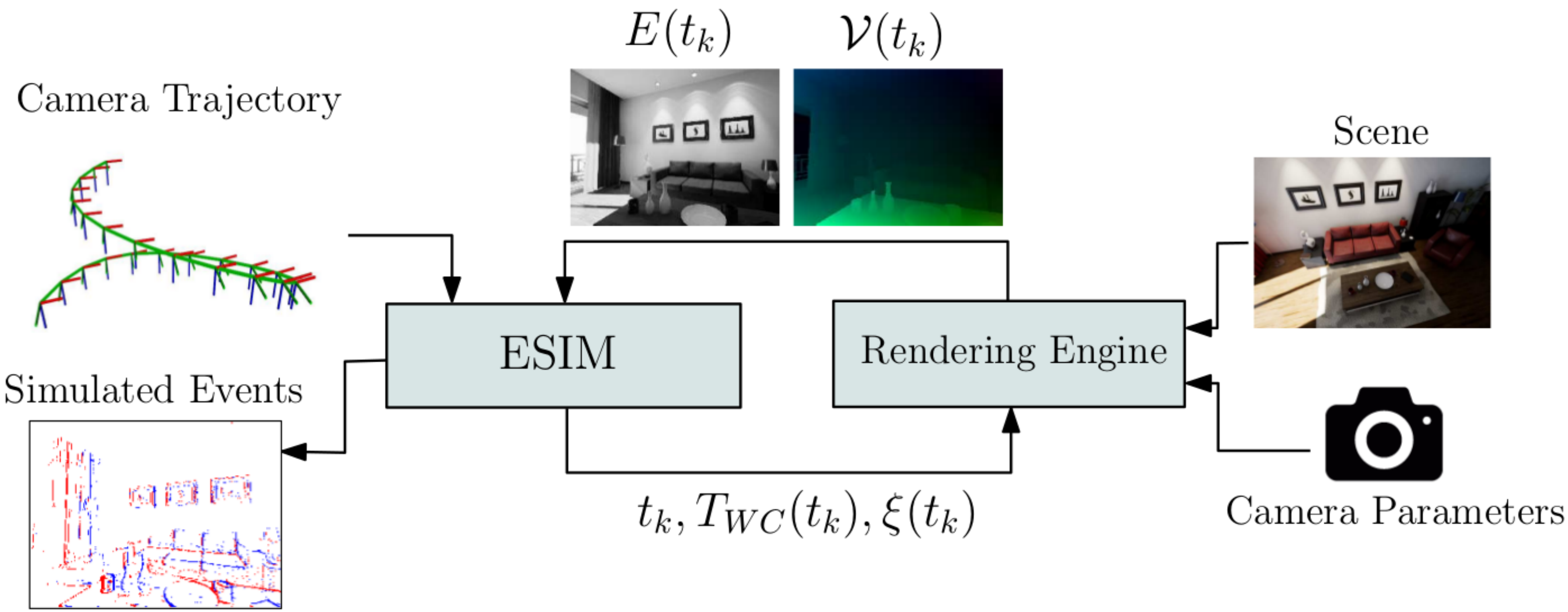
- Event camera principles
- Event camera calibration
- Event camera manufactures, products, and applications

Event Data Simulation

- Why need event data simulation?
  - Not everyone has an event camera!
  - To configure parameters in event generation process
    (To freely control the event generation:
    noise level, resolution, triggering threshold, ...)
  - To help model the event sensor model mathematically.

Event Data Simulation

- Event data simulators
  - [ESIM: uzh-rpg/rpg_esim: ESIM: an Open Event Camera Simulator (github.com)]



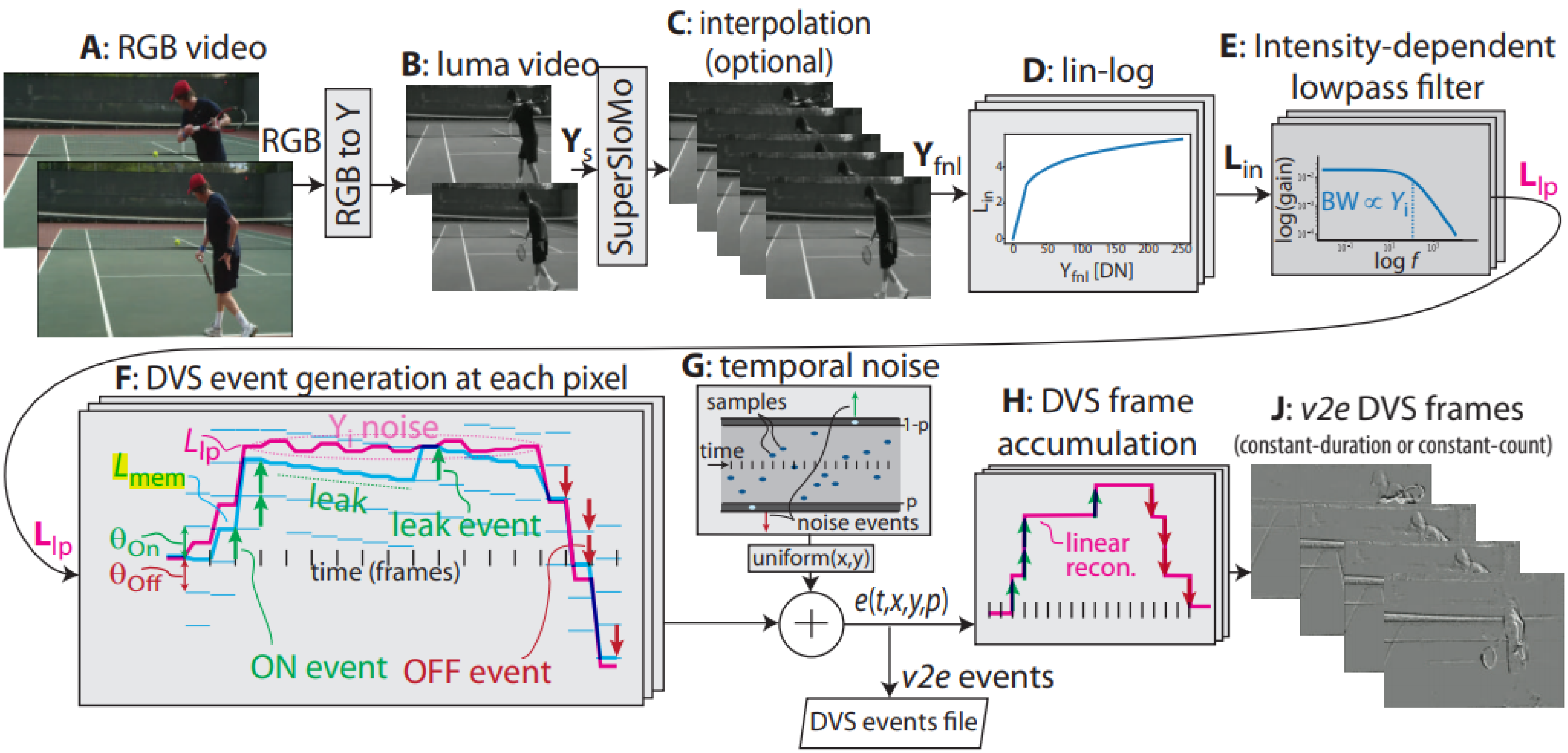
ESIM (uzh.ch)

Event Data Simulation

ESIM (uzh.ch)

Event Data Simulation

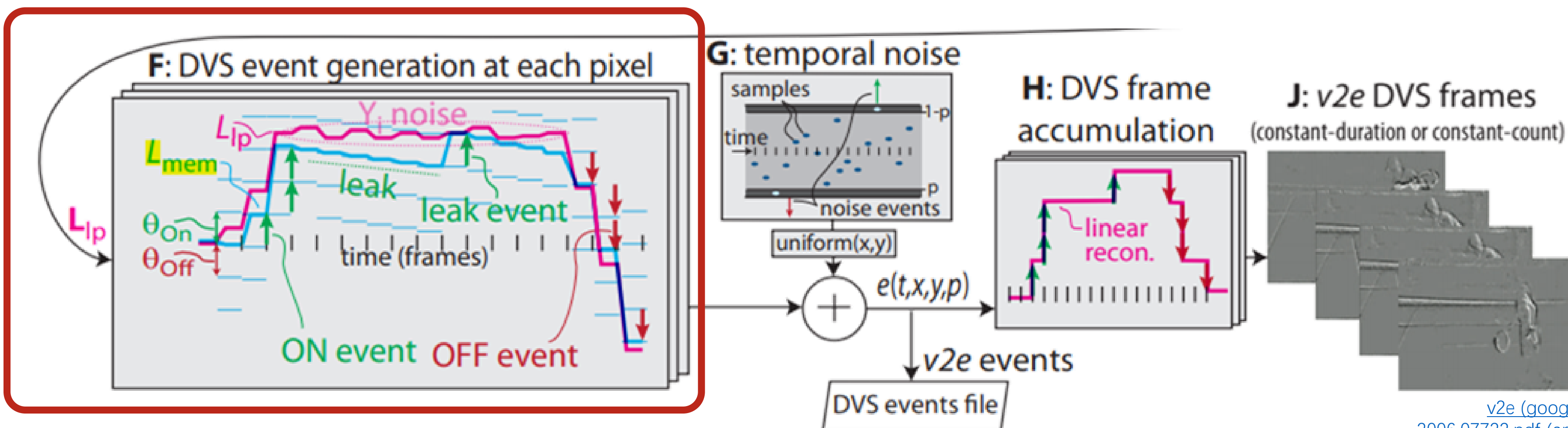
- Event data simulators
  - v2e (google.com)



v2e (google.com)
2006.07722.pdf (arxiv.org)

Event Data Simulation

- Noises in Event generation model
  - **Threshold mismatch:** Typical value θ = 0.3.
    Threshold varies with a Gaussian distribution $\sigma_\theta \approx 0.03$.
  - **Hot pixels:** DVS sensors always have some 'hot pixels', which continuously fire events at a high rate even in the absence of input.
  - **Leak noise events:** DVS pixels emit spontaneous ON events called leak events with typical rates ≈ 0.1 Hz.



v2e (google.com)
2006.07722.pdf (arxiv.org)

Event Data Simulation

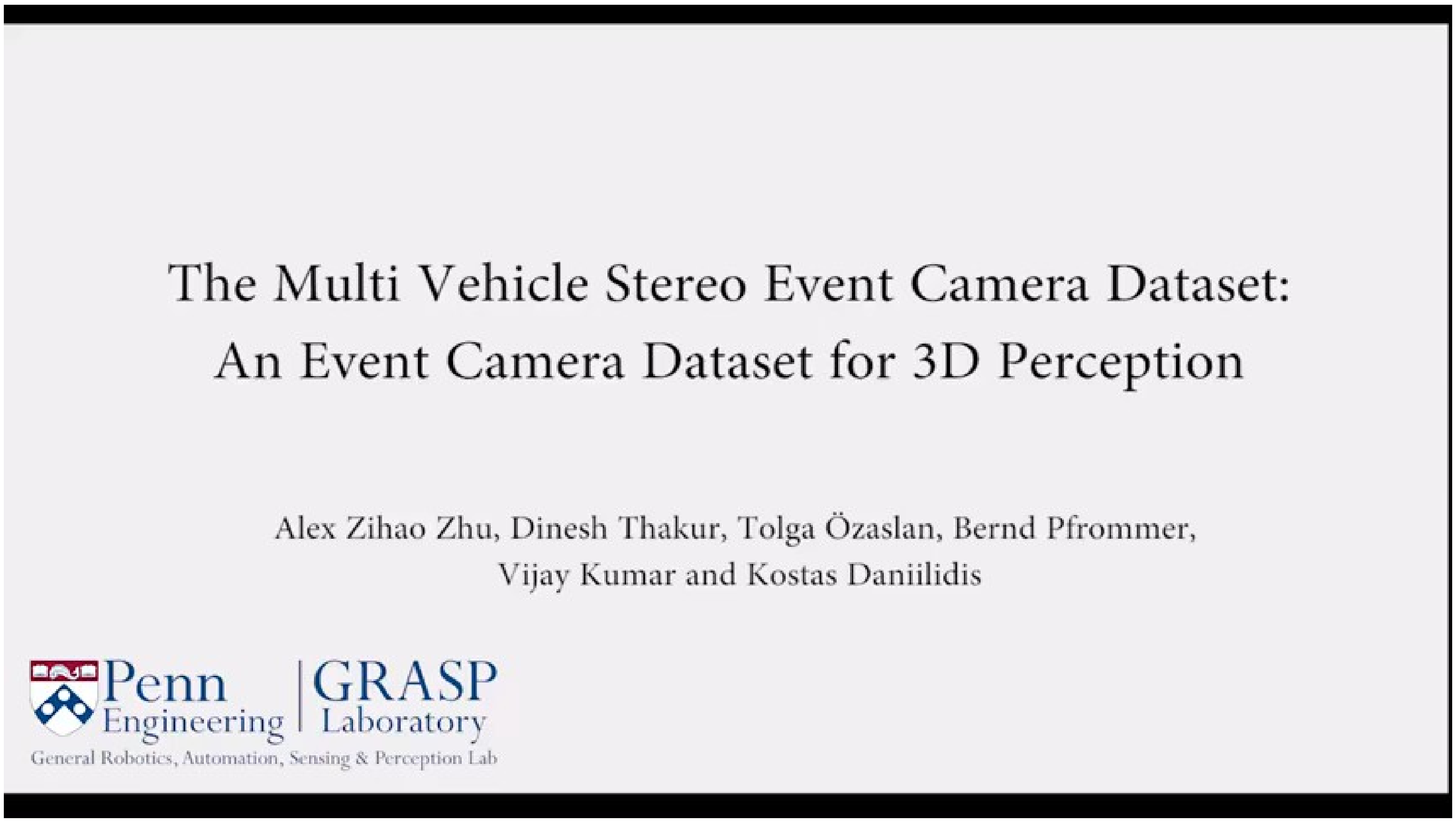
- v2e vs ESIM

ESIM	v2e
Complex front-end: • Can simulate **camera motion**	Simpler front-end: • Only process movies **without modelling 3D environments**
Programmed in **C++**	Programmed in **Python**
Basic pixel model: • Contrast thresholds with Gaussian noise **for the whole sensor**	More realistic pixel model: • **Pixel to pixel** Gaussian temporal contrast threshold variation • Finite**, intensity-dependent photoreceptor bandwidth** • **Leak events** (intensity-dependent background activity noise) • **Intensity-dependent temporal noise**

v2e (google.com)

Event Datasets

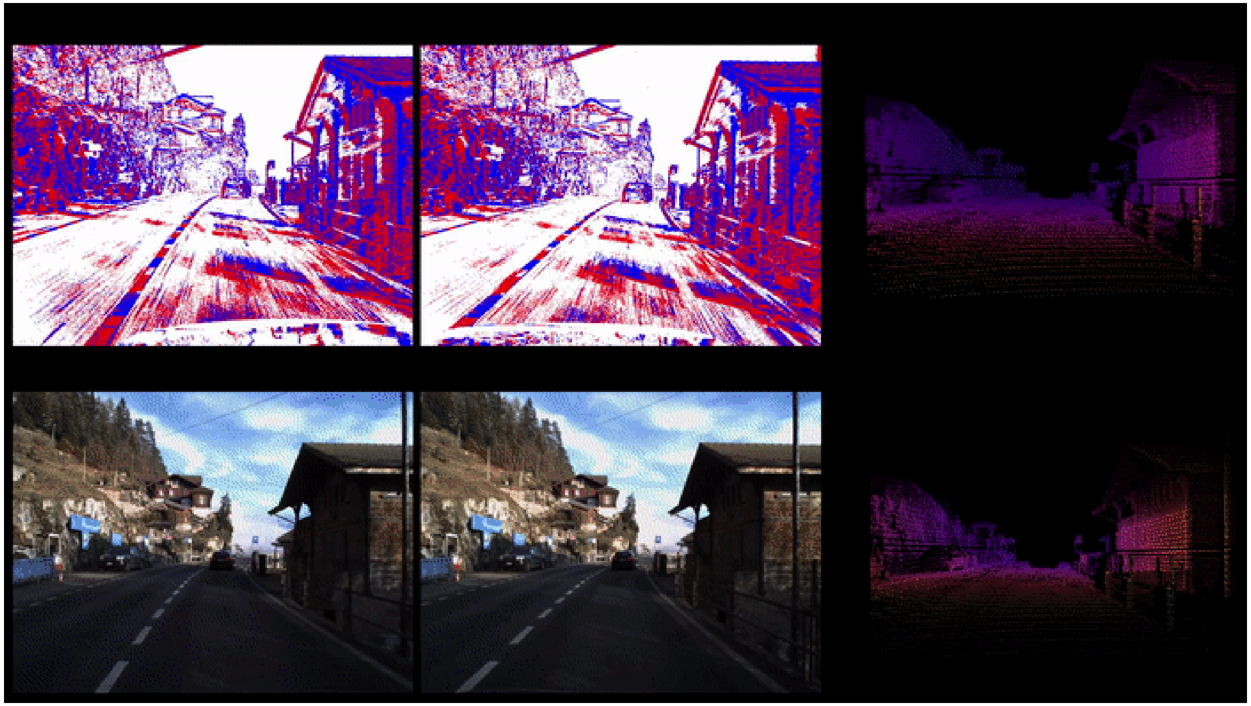
- The Multi Vehicle Stereo Event Camera Dataset (MVSEC)
  - Good for **optical flow estimation, pose estimation, and SLAM**



[Multi Vehicle Stereo Event Camera Dataset (daniilidis-group.github.io)](https://daniilidis-group.github.io)

Event Datasets

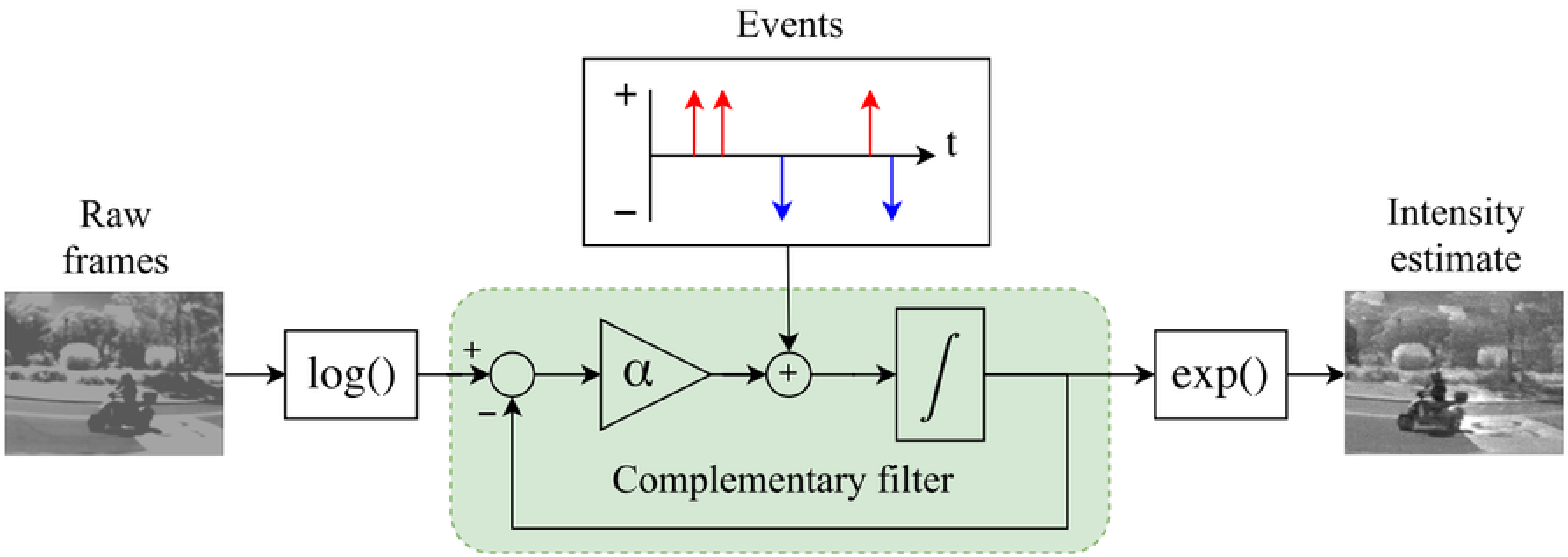
- A Stereo Event Camera Dataset for Driving Scenarios (DSEC Dataset)
  - **Stereo camera dataset in driving scenarios**
  - 2 monochrome event cameras + 2 global shutter color cameras
  - Lidar data and RTK GPS measurements



[DSEC – A Stereo Event Camera Dataset for Driving Scenarios (uzh.ch)]

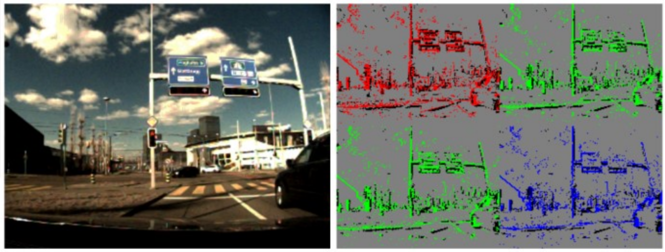
Event Datasets

- DVS Image Reconstruction Dataset
  - Together with paper "*Continuous-time Intensity Estimation Using Event Cameras*"
  - A good start to do **image reconstruction** using both events and frames
  - Details of the paper will be discussed in the future.



[Continuous-time Intensity Estimation Using Event Cameras - Cedric Scheerlinck]
[dvs_image_reconstruction_dataset - Google Drive]

Event Datasets



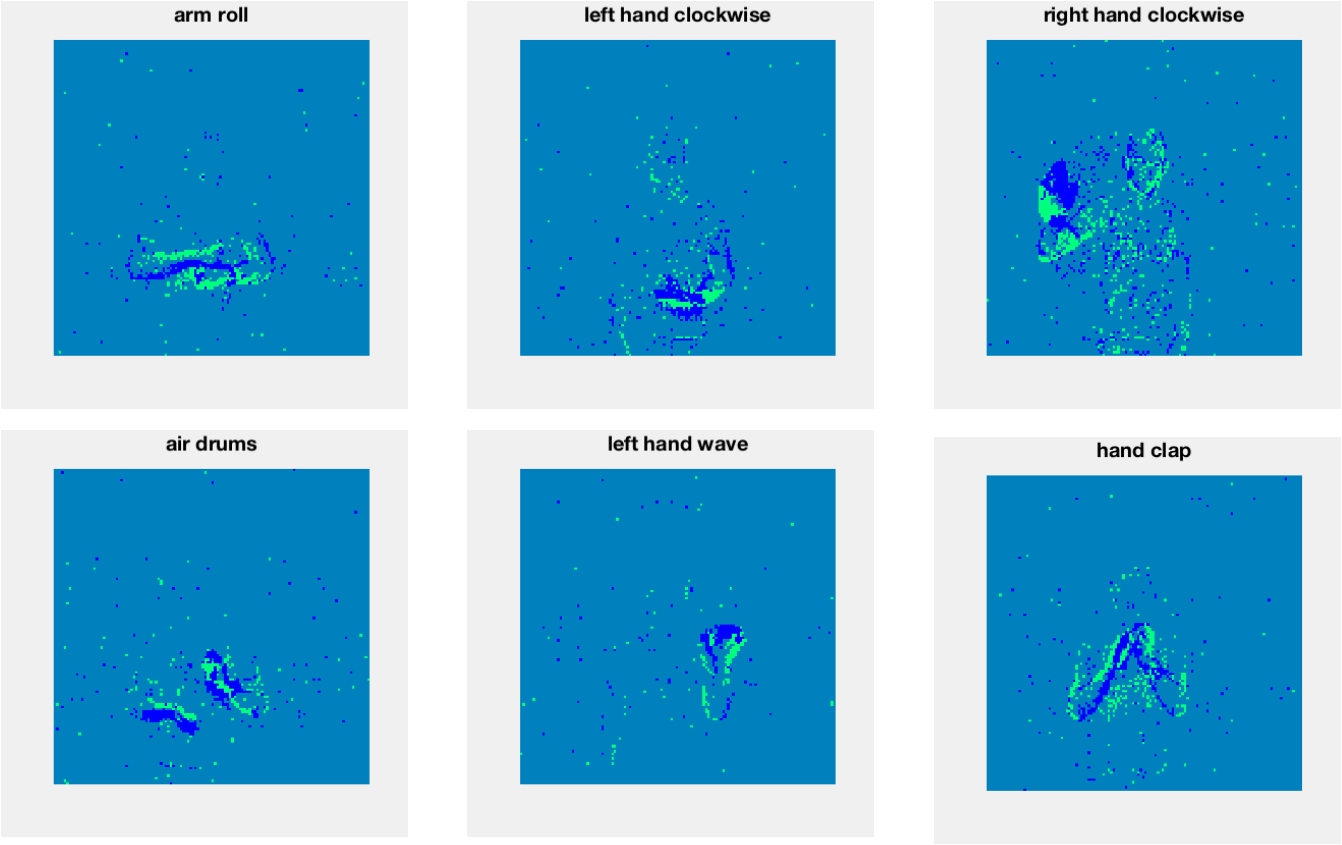
DAVIS frame Events

- CED: Color Event Camera Dataset
  - Featuring 50 minutes of footage with both color frames and color events from the Color-DAVIS346.

[CED: Color Event Camera Dataset (uzh.ch)](rpg.ifi.uzh.ch)
rpg.ifi.uzh.ch/docs/CVPRW19_Scheerlinck.pdf

Event Datasets

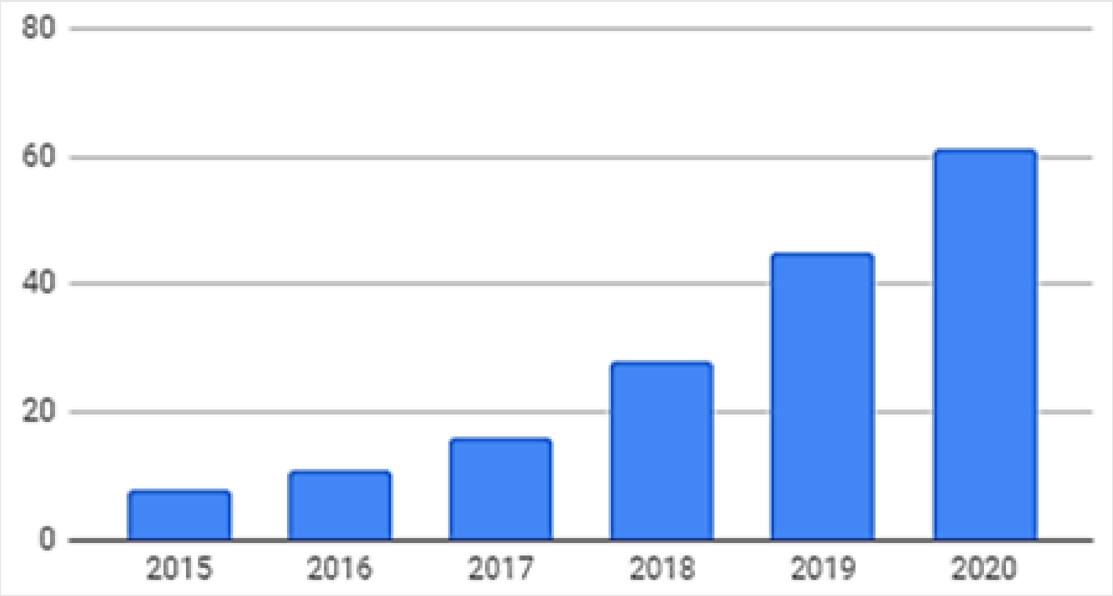
- DVS128 Gesture Dataset
  - 11 hand gestures from 29 subjects under 3 illumination conditions
  - Can be used to build the **gesture recognition system**



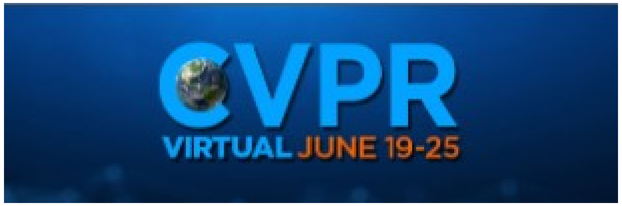
DVS128 Gesture Dataset - IBM Research

Fast-Growing Community

- Statistics: Papers in computer vision and robotics venues



- [CVPR 2021 Workshop on Event-based Vision (tub-rip.github.io)]
  - Video/slides are free available online!



Guillermo Gallego - Event-based Vision (google.com)

Summary – Event Data Simulation and Datasets

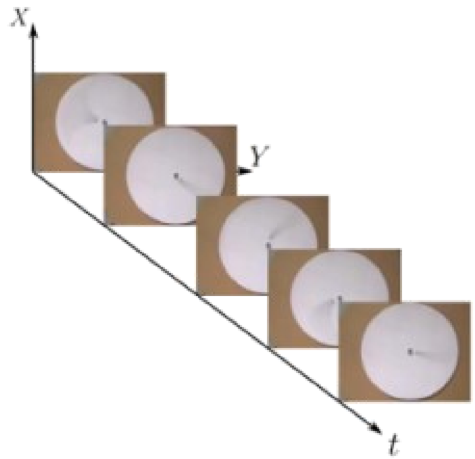
- The reasons for event data simulation and event datasets
- Commonly-used event simulator and datasets
- Fast-growing community in event-based vision research

Event Representations

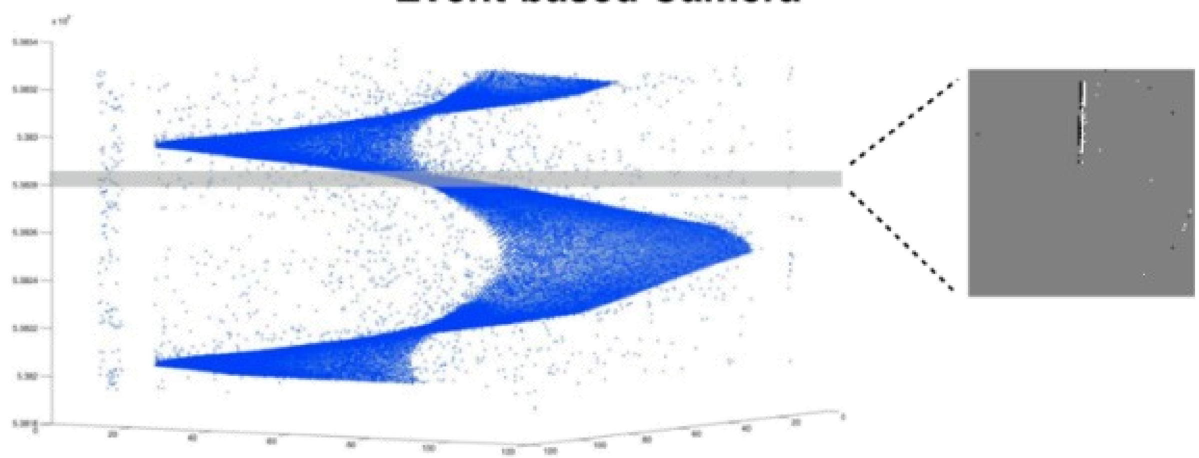
- Ways of event representations
  - Raw data – event “point cloud”
  - Event frames
  - Event tensors
- Influence on data rate
  - Compute the data rate for different representations

Raw Data – Event “Point Cloud”

Frame-based Camera



Event-based Camera

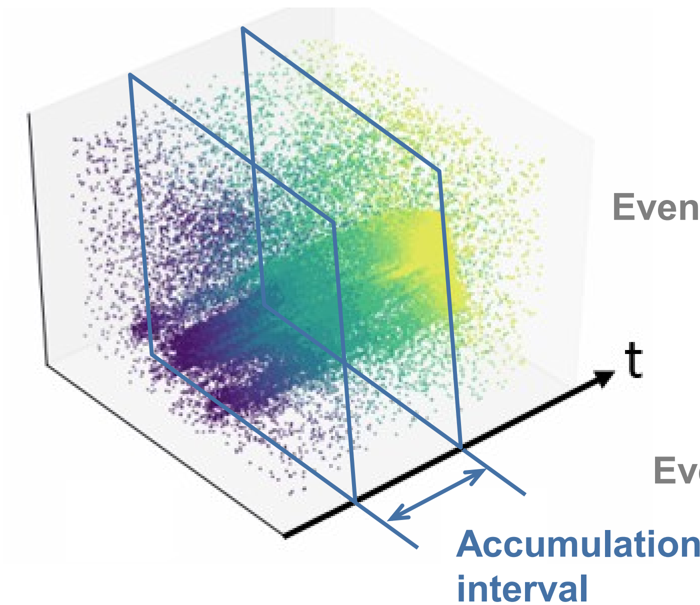


+ All information retained
- Requires algorithm that can accept point cloud as input

Event-Based Visual Flow Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozz, [untitled (neuromorphic-vision.com)](neuromorphic-vision.com)

Event Frames

Events



Event count frame

4	5	5	6	2
5	4	2	1	2
5	4	6	3	3
0	6	6	5	3
1	0	2	0	0

The number of triggering times

Event binary frame

0	1	0	1	0
0	0	0	1	0
0	0	0	0	0
0	1	1	1	0
0	0	0	0	1

If the pixel has been triggered

Event gray frame

0	0	64	220	48
0	58	99	0	5
32	16	1	168	32
2	0	18	38	147
0	24	200	45	0

The absolute intensity of the pixel

Event flow frame

9	6	64	220	48
0	0	99	23	222
32	16	100	168	32
36	48	18	38	147
59	24	200	45	45

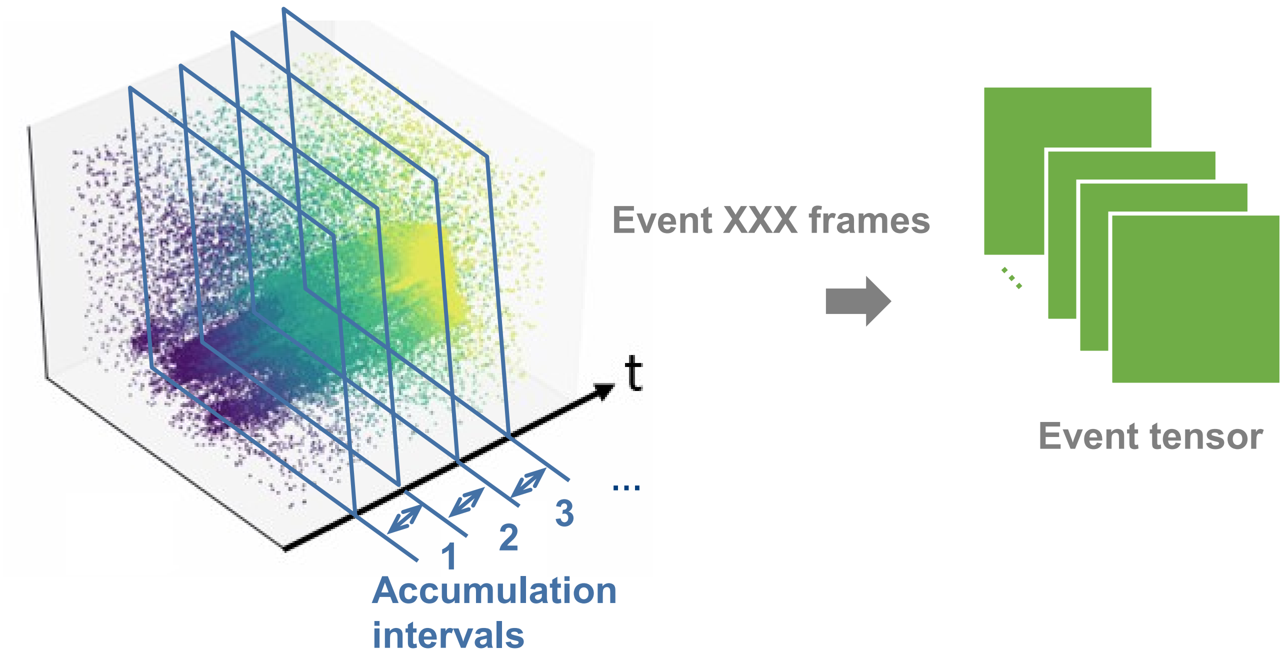
The triggering time of the pixel

+ Easy to process by off-the-shelf machine vision algorithms
- Information loss
- Only contains events in accumulation interval

Different types of event frames

Event Tensors

- Adding another dimension to event frames, representing long-time events



+ Easy to process by off-the-shelf machine vision algorithms (especially for deep learning)
+ Represents events in longer time interval without loss too much information
- Requires experiments to determine the number and length of accumulation intervals for best performance, depending on different application

Paper: End-to-End Learning of Representations for Asynchronous Event-Based Data

- Event tensors in literatures

Representation	Dimensions	Description	Characteristics
Event frame [53]	$H \times W$	Image of event polarities	Discards temporal and polarity information
Event count image [36, 69]	$2 \times H \times W$	Image of event counts	Discards time stamps
Surface of Active Events (SAE) [7, 69]	$2 \times H \times W$	Image of most recent time stamp	Discards earlier time stamps
Voxel grid [70]	$B \times H \times W$	Voxel grid summing event polarities	Discards event polarity
Histogram of Time Surfaces (HATS) [59]	$2 \times H \times W$	Histogram of average time surfaces	Discards temporal information
Event Spike Tensor (EST, our work)	$2 \times B \times H \times W$	Sample event point-set into a grid	Discards the least amount of information

Table 1. Comparison of grid-based event representations used in prior work on event-based deep learning. H and W denote the image height and width dimensions, respectively, and B the number of temporal bins.

Is it possible to learn the event representation from data?

- We first need to define the "**general form**" of event representation, and then keep some terms fixed and train the other terms.
- In other words, define the following **f**, given input raw events x:

$$y = \mathbf{f}(w, x)$$

where y is the event representation.

flexible parameters to train

1904.08245.pdf (arxiv.org)

Paper: End-to-End Learning of Representations for Asynchronous Event-Based Data

- Raw event field representation for event $e_k = \{x_k, y_k, t_k\}$

$$S_{\pm}(x,y,t) = \sum_{e_k \in \mathcal{E}_{\pm}} \delta(x - x_k, y - y_k)\delta(t - t_k)$$

- We further define the event measurement field by adding the "**f**" term:

$$S_{\pm}(x,y,t) = \sum_{e_k \in \mathcal{E}_{\pm}} f_{\pm}(x,y,t)\delta(x - x_k, y - y_k)\delta(t - t_k)$$

where f can be either:

- event polarity $f_{\pm}(x,y,t) = \pm 1$
- event count $f_{\pm}(x,y,t) = 1$
- normalized timestamp $f_{\pm}(x,y,t) = \frac{t-t_0}{\Delta t}$

Above is still ill-defined due to the use of Dirac pulses.

We may add a "smoothing" kernel.

1904.08245.pdf (arxiv.org)

Paper: End-to-End Learning of Representations for Asynchronous Event-Based Data

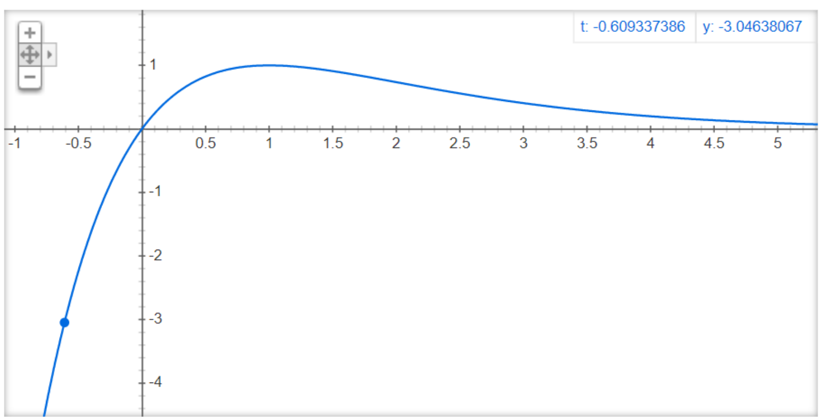
- Convolution with kernel "**k**"

$$(k * S_\pm)(x, y, t) = \sum_{e_k \in \mathcal{E}_\pm} f_\pm(x_k, y_k, t_k) k(x - x_k, y - y_k, t - t_k)$$

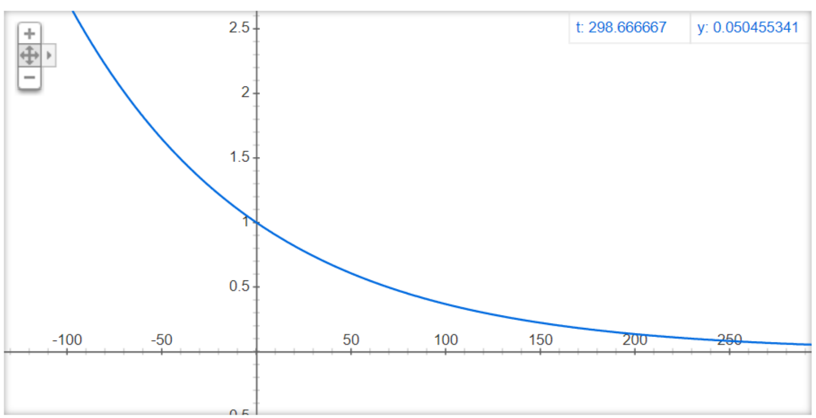
  - alpha-kernel $k(x, y, t) = \delta(x, y) \frac{et}{\tau} \exp(-t/\tau)$
  - exponential kernel $k(x, y, t) = \delta(x, y) \frac{1}{\tau} \exp(-t/\tau)$

Kernels are task-dependent with no general agreement on optimal one.

Graph for e*t/1*exp((-t)/1)



Graph for 1*exp((-t)/100)



1904.08245.pdf (arxiv.org)

Paper: End-to-End Learning of Representations for Asynchronous Event-Based Data

- Sampling the convolved signal **at regular intervals** to generate grid representations

$$(k * S_\pm)(x,y,t) = \sum_{e_k \in \mathcal{E}_\pm} f_\pm(x_k, y_k, t_k) k(x - x_k, y - y_k, t - t_k)$$

Event Spike Tensor (EST)

$$S_\pm[x_l, y_m, t_n] = (k * S_\pm)(x_l, y_m t_n) = \sum_{e_k \in \mathcal{E}_\pm} f_\pm(x_k, y_k, t_k) k(x_l - x_k, y_m - y_k, t_n - t_k)$$

How to find the best representation from data perspective?

Replace the kernel function with a **multilayer perceptron** with two hidden layers each with 30 units.

1904.08245.pdf (arxiv.org)

Paper: End-to-End Learning of Representations for Asynchronous Event-Based Data

- Results: Average end-point error (AEE) and % of outliers evaluation on the MVSEC dataset.

Representation	Measurement	Kernel	*indoor_flying1*		*indoor_flying2*		*indoor_flying3*	
			AEE	% Outlier	AEE	% Outlier	AEE	% Outlier
Two-Channel Image [36]	count	trilinear	1.21	4.49	2.03	22.8	1.84	17.7
EV-FlowNet [69]	-	trilinear	1.03	2.20	1.72	15.1	1.53	11.9
Voxel Grid [70]	polarity	trilinear	0.96	1.47	1.65	14.6	1.45	11.4
Event Frame	time stamps	trilinear	1.17	2.44	1.93	18.9	1.74	15.5
Two-Channel Image	time stamps	trilinear	1.17	1.5	1.97	14.9	1.78	11.7
Voxel Grid	time stamps	trilinear	0.98	1.20	1.70	14.3	1.5	12.0
EST (Ours)	time stamps	trilinear	1.00	1.35	1.71	11.4	1.51	8.29
EST (Ours)	time stamps	alpha	1.03	1.34	1.52	11.7	1.41	8.32
EST (Ours)	time stamps	exponential	**0.96**	1.27	1.58	10.5	**1.40**	9.44
EST (Ours)	time stamps	learnt	0.97	**0.91**	**1.38**	**8.20**	1.43	**6.47**

AEE: The EST outperforms the state-of-the-art by a large margin (12%).
% Outlier: Reduced by an average of 49%.

1904.08245.pdf (arxiv.org)

Paper: End-to-End Learning of Representations for Asynchronous Event-Based Data

- Adding another dimension to event frames, representing long-time events

1904.08245.pdf (arxiv.org)

Influence of Event Representation on Data Rate

- Suppose each pixel triggers **1 time per second** on average;
  The resolution of the sensor is **1 Mega Pixels**.
  Event (x, y, p, t) encoding is **(uint10, uint10, binary, float32).**
- Calculate and compare the data rate for:
  - event point cloud
  - 30 fps event binary frames (binary)
  - 30 fps event count frames (uint8)

Influence of Event Representation on Data Rate

Solution:

Note:

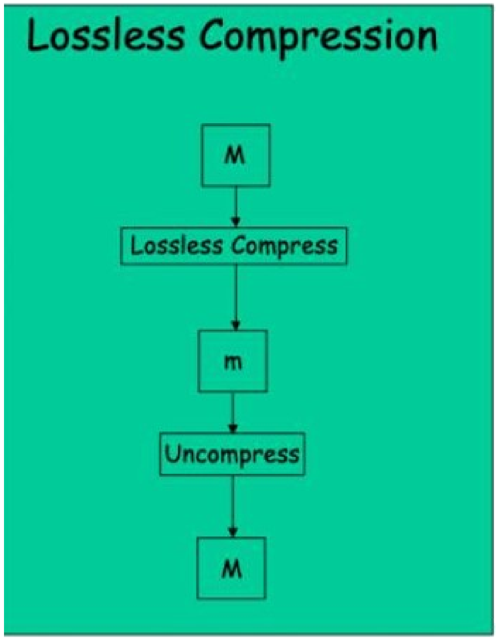
- The data rate for event frames **does not change** w.r.t. event rate, with a **fixed event frame rate**.
  → Data rate more stable.
- The data rate depends on particular scenarios. E.g., what if each pixel triggers 10 times per second?
  → The event frame representation will have obviously smaller data rate!

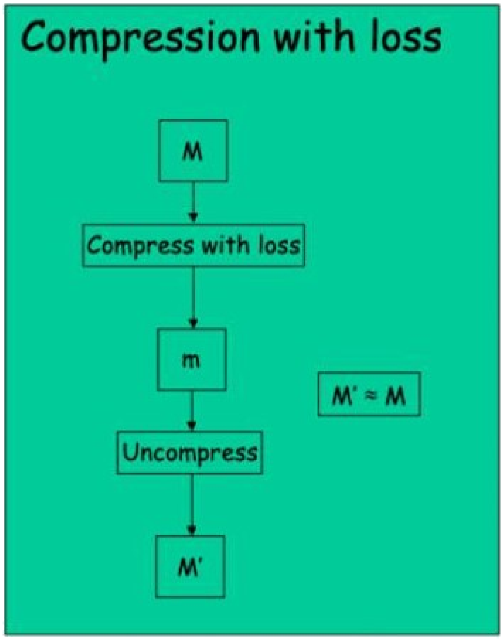
Event Data Compression

- Some people say, “Event data rate is already much less then conventional camera.” Why is data compression still required?
  - Above statement is not always true. In practice, the event data rate varies and depends on the scenario.
  - It is always better to use less bandwidth/space to transmit/store data!
- Two main streams for event data compression:
  - Lossy compression

  → Event frames, event tensors are lossy compression.
  - **Lossless compression**

  → The main research direction for event data compression.



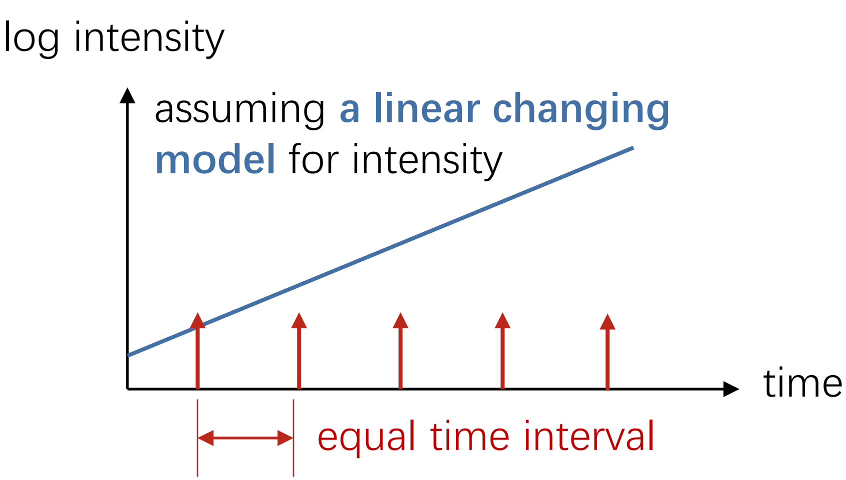


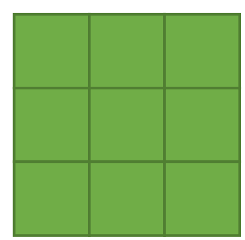
https://www.thebroadcastbridge.com/content/entry/16741/lossless-compression-rewriting-data-in-a-more-efficient-way

Event Data Compression

- Ideas
  - General event data compression approaches can still be applied.
  - DVS-specific compression approaches. These approaches utilizes: temporal redundancy, and spatial redundancy of events.

- Temporal redundancy (left) and spatial redundancy (right)





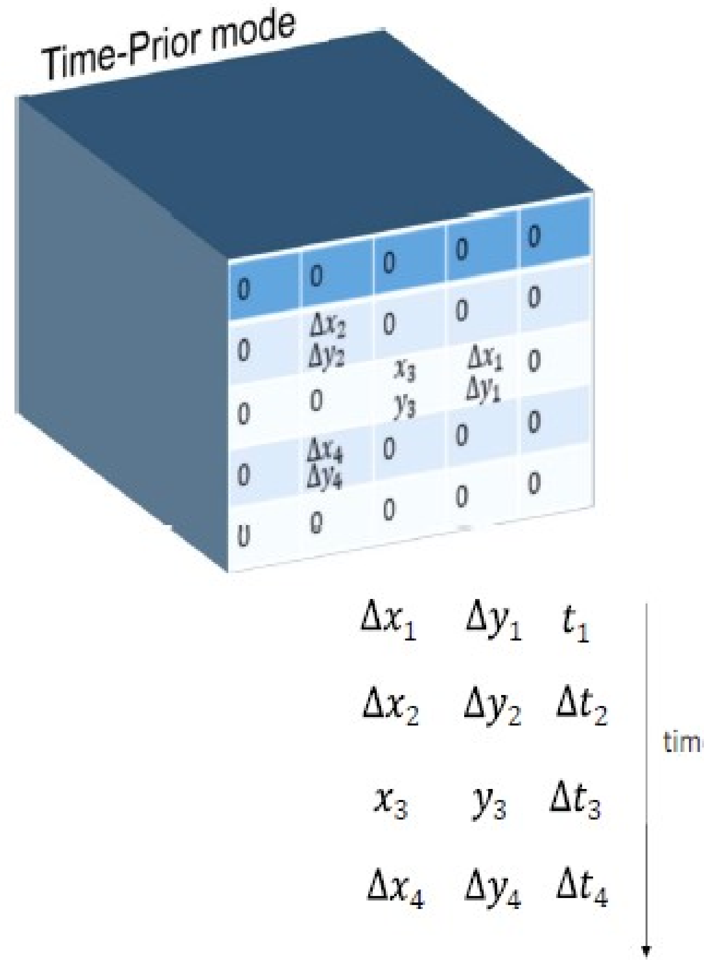
assuming **adjacent pixels receive almost the same luminance concurrently**

Two modes based on above redundancies are available for event compression.

https://www.researchgate.net/publication/341585734_Lossless_Compression_of_Data_From_Static_and_Mobile_Dynamic_Vision_Sensors-Performance_and_Trade-Offs

Time-Prior Mode

- Exploits spatial redundancy
  - Finds a centre point (x3, y3 in the fig).
  - Projects all other events w.r.t the centre point.
  - We call this “**delta coding**” on **x** and **y**.



We could use less bits to represent Δx and Δy, based on the spatial redundancy.

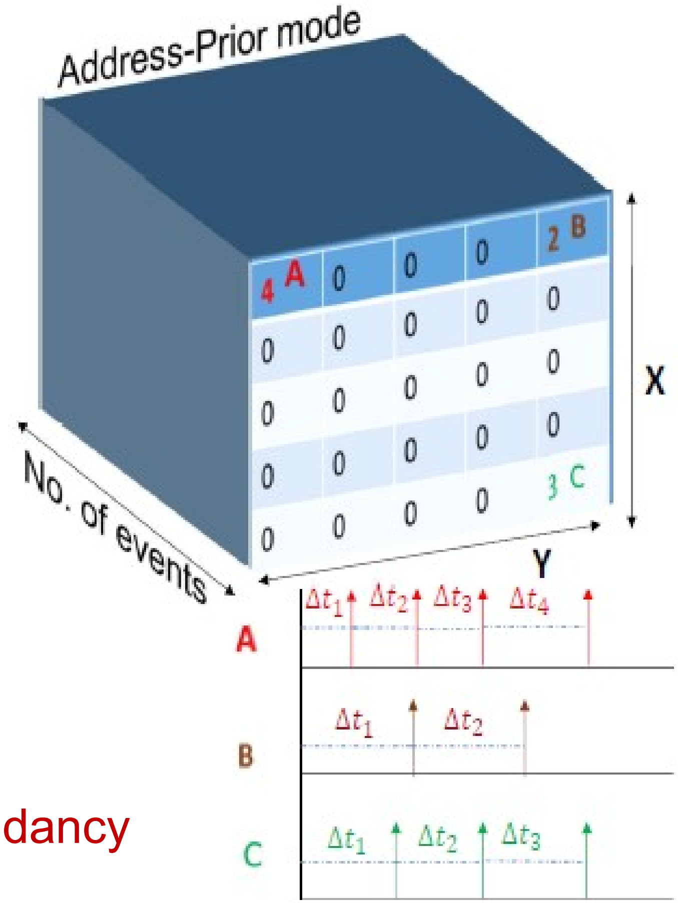
→ **Efficient compression on address field.**

Drawback: Not suitable for events in a large area.

https://www.researchgate.net/publication/341585734_Lossless_Compression_of_Data_From_Static_and_Mobile_Dynamic_Vision_Sensors-Performance_and_Trade-Offs

Address-Prior Mode

- Exploits temporal redundancy
  - The number of events at each pixel is recorded, as shown in the fig.
  - Similarly, we do "**delta coding**" on **time**.



We could use less bits to represent Δt, based on the temporal redundancy.

→ **Efficient compression on timestamp field.**

Open question: can we utilize the temporal redundancy and spatial redundancy simultaneously?

https://www.researchgate.net/publication/341585734_Lossless_Compression_of_Data_From_Static_and_Mobile_Dynamic_Vision_Sensors-Performance_and_Trade-Offs

Spike Event Polarity

- Polarity can be compressed individually based on a entropy coder.

Assumption: If the polarity of the previous event is one (or zero) on a pixel, there is a high probability that the **next event polarity will be one (or zero) on the same pixel.**

  - Idea of entropy coding: information with larger probability is coded using a shorter binary string.

Information	Probability	Binary string
S_1	1/2	0
S_2	1/4	10
S_3	1/8	110
S_4	1/16	1110
S_5	1/64	111100
S_6	1/64	111101
S_7	1/64	111110
S_8	1/64	111111

3.6 ENTROPY CODING - Audio Signal Processing and Coding [Book] (oreilly.com)

Summary – Event Data Processing

- Event data representations
- Event data compression

References

1. Event-based Vision: A Survey (arxiv.org)
2. CIS 849 -- Autonomous Robot Vision (udel.edu)
3. Course Detail (imaging.org)