Tutorial on Event-based Vision for High-Speed Robotics

Davide Scaramuzza

Robotics and Perception Group
http://rpg.ifi.uzh.ch
University of Zurich
Current Research

Visual & Inertial State Estimation and Mapping
[T-RO’08, IJCV’11, PAMI’13, RSS’15]

Autonomous Navigation of Flying Robots
[AURO’12, RAM’14, JFR’15a-b]

Collaboration of Aerial and Ground Robots
[IROS’13, SSRR’14]

Event-based Vision for Agile Flight
[IROS’3, ICRA’14-15, RSS’15]
Outline

- Motivation
- Event-based Cameras: DVS and DAVIS
  - Generative model
  - Calibration
  - Visualization
  - Life-time estimation
  - Pose estimation
The Progress of Autonomous Robotics

Past

Kuka KR240

AGV

Autonomous Ground Vehicles

Present

2000

Perception Improvements

KIVA’s Robotics Warehouse

Mars rovers

Future?

Google Car

iCub

UPenn’s Swarm of Quadcopters
A Comparison between Off-board and On-board sensing

**Off-board sensors**

- **VICON-controlled quadcopter**
  - Mueller, Lupashin, D’Andrea

**Onboard sensors**

- **VISION-controlled quadcopter**
  - Fontana, Faessler, Scaramuzza
Open Problems and Challenges with Micro Helicopters

Current flight maneuvers achieved with onboard cameras are still slow compared with those attainable with Motion Capture Systems.
How fast can we go with an onboard camera?

Let’s assume that we have perfect perception

Can we achieve the same flight performances attainable with motion capture systems or go even faster?
To go faster, we need faster sensors!

- At the current state, the agility of a robot is limited by the latency and temporal discretization of its sensing pipeline [Censi & Scaramuzza, ICRA’14]

- Currently, the average robot-vision algorithms have latencies of 50-200 ms. This puts a hard bound on the agility of the platform. [Censi & Scaramuzza, ICRA’14]

To go faster, we need faster sensors!

- At the current state, the agility of a robot is limited by the latency and temporal discretization of its sensing pipeline.

- Currently, the average robot-vision algorithms have latencies of 50-200 ms. This puts a hard bound on the agility of the platform.

- Can we create low-latency, low-discretization perception architectures?

  Yes...

  ...if we use a camera where pixels do not spike all at the same time

  ...in a way as we humans do..
Human Vision System

- Retina is ~1000mm²
- 130 million photoreceptors
  - 120 mil. rods and 10 mil. cones for color sampling
  - 1.7 million axons
Human Vision System
Dynamic Vision Sensor (DVS)

**Event-based camera** developed by Tobi Delbruck’s group (ETH & UZH).

- **Temporal resolution:** 1 μs
- **High dynamic range:** 120 dB
- **Low transmission bandwidth:** ~200Kb/s
- **Low power:** 20 mW
- **Cost:** 2,500 EUR

Image of the solar eclipse (March’15) captured by a DVS (courtesy of Sim Bamford by INILabs)

DARPA project Synapse: 1M neuron, brain-inspired processor: IBM TrueNorth

Camera vs DVS

- A traditional camera outputs frames at **fixed time intervals**:

  ![Diagram of frame and next frame]

  

- By contrast, a **DVS** outputs **asynchronous events** at **microsecond resolution**.

  An event is generated each time a single pixel changes value.

  ![Diagram of events stream]

  

  event: \[ (t, (x, y), \text{sign} \left( \frac{d}{dt} \log(I_t(x, y)) \right) \) \]

  sign (+1 or -1)

Camera vs Dynamic Vision Sensor

Video: [http://youtu.be/LauQ6LWTkxM](http://youtu.be/LauQ6LWTkxM)
If you intend to use this video in your presentations, please credit the authors of the paper below, plus the paper.

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS'14]
DVS Operating Principle [Lichtsteiner, ISCAS’09]

Events are generated any time a single pixel sees a change in brightness larger than $C$

$$\Delta \log I \geq C$$

The intensity signal at the event time can be reconstructed by integration of $\pm C$

[Cook et al., IJCNN’11] [Kim et al., BMVC’15]

Dynamic Vision Sensor (DVS)

Advantages

1. **low-latency** (~1 micro-second)
2. **high-dynamic range** (120 dB instead 60 dB)
3. Very **low bandwidth** (only intensity changes are transmitted): ~200Kb/s
4. Low storage capacity, processing time, and power

Disadvantages

1. Requires totally **new vision algorithms**
2. **No intensity information** (only binary intensity changes)
3. **Very low image resolution**: 128x128 pixels

## High-speed cameras vs DVS

<table>
<thead>
<tr>
<th></th>
<th>Photron Fastcam SA5</th>
<th>Matrix Vision Bluefox</th>
<th>DVS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max fps or measurement rate</strong></td>
<td>1MHz</td>
<td>90 Hz</td>
<td>1MHz</td>
</tr>
<tr>
<td><strong>Resolution at max fps</strong></td>
<td>64x16 pixels</td>
<td>752x480 pixels</td>
<td>128x128 pixels</td>
</tr>
<tr>
<td><strong>Bits per pixels</strong></td>
<td>12 bits</td>
<td>8-10</td>
<td>1 bits</td>
</tr>
<tr>
<td><strong>Weight</strong></td>
<td>6.2 Kg</td>
<td>30 g</td>
<td>30 g</td>
</tr>
<tr>
<td><strong>Active cooling</strong></td>
<td>yes</td>
<td>No cooling</td>
<td>No cooling</td>
</tr>
<tr>
<td><strong>Data rate</strong></td>
<td>1.5 GB/s</td>
<td>32MB/s</td>
<td>~200KB/s on average</td>
</tr>
<tr>
<td><strong>Power consumption</strong></td>
<td>150 W + Ilighting</td>
<td>1.4 W</td>
<td>20 mW</td>
</tr>
<tr>
<td><strong>Dynamic range</strong></td>
<td>n.a.</td>
<td>60 dB</td>
<td>120 dB</td>
</tr>
</tbody>
</table>
Related Work (1/2)

- **Event-based Tracking**
  - Conradt et al., ISCAS’09
  - Drazen, 2011
  - Mueller et al., ROBIO’11
  - Censi et al., IROS’13
  - Delbruck & Lang, Front. Neuros.’13
  - Lagorce et al., T-NNLS’14

- **Event-based Optic Flow**
  - Cook et al, IJCNN’11
  - Benosman, T-NNLS’14

- **Event-based ICP**
  - Ni et al., T-RO’12

Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor [Delbruck & Lang, Frontiers in Neuroscience, 2013]

Asynchronous Event-Based Multikernel Algorithm for High-Speed Visual Features Tracking [Lagorce et al., TNNLS’14]
Related Work (1/2)

Related Work (2/2)

- **Event-based 6DoF Localization**
  - Weikersdorfer et al., ROBIO’12
  - Mueggler et al., IROS’14

- **Event-based Rotation estimation**
  - Cook et al, IJCNN’11
  - Kim et al, BMVC’15

- **Event-based Visual Odometry**
  - Censi & Scaramuzza, ICRA’14

- **Event-based SLAM**
  - Weikersdorfer et al., ICVS’13

- **Event-based 3D Reconstruction**
  - Carneiro’13

---

Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, [Mueggler et al., IROS’14]

Simultaneous Localization and Mapping for Event-Based Vision Systems [Weikersdorfer et al., ICVS’13]
Related Work: Event-based Tracking

- **Collision avoidance**
  - Guo, ICM’11
  - Clady, FNS’14
  - Mueggler, ECMR’13

- **Estimating absolute intensities**
  - Cook et al, IJCNN’11
  - Kim et al, BMVC’15

- **HDR panorama & Mosaicing**
  - Kim et al, BMVC’15
  - Belbachir, CVPRW’14, Schraml, CVPR’15

Towards Evasive Maneuvers with Quadrotors using Dynamic Vision Sensors [Mueggler et al., ECMR’15]

Simultaneous Mosaicing and Tracking with an Event Camera [Kim et al., BMVC’15]

Interacting Maps for Fast Visual Interpretation [Cook et al., IJCNN’11]
Live Demos
A Simple Use Case:
Active LED marker Tracking

[Censi, Brandli, Delbruck, Scaramuzza, Low-latency localization by Active LED Markers tracking using a Dynamic Vision Sensor, IROS’13]
Low-latency Active LED Tracking [IROS’13]

- Active LED blinking a high frequency (>1 KHz).
- A DVS can detect the LED position and discriminate frequency
- Advantages:
  - simple
  - low latency
  - robust to interferences

Blinking LEDs with different frequency act as uniquely identifiable markers

Time slice = blinking period × 2

[Censi, Brandli, Delbruck, Scaramuzza, Low-latency localization by Active LED Markers tracking using a Dynamic Vision Sensor, IROS’13]
Low-latency Active LED Tracking [IROS’13]

- Robust to the camera motion

**no LEDs, with motion**

**with LEDs, no motion**

**LEDs + motion**

[Censi, Brandli, Delbruck, Scaramuzza, Low-latency localization by Active LED Markers tracking using a Dynamic Vision Sensor, IROS’13]
Results: Flip

[Censi, Brandli, Delbruck, Scaramuzza, Low-latency localization by Active LED Markers tracking using a Dynamic Vision Sensor, IROS’13]
Calibration
[IROS’14]

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]
Calibration of a DVS [IROS’14]

- Standard **pinhole camera model** still valid (same optics)
- Standard passive calibration patterns **cannot be used**
  - need to move the camera → inaccurate corner detection

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]
Calibration of a DVS [IROS’14]

- Standard **pinhole camera model** still valid (same optics)
- Standard passive calibration patterns **cannot be used**
  - need to move the camera → inaccurate corner detection
- **Blinking patterns** (computer screen, LEDs)
- ROS DVS driver + intrinsic and extrinsic stereo calibration **open source**: https://github.com/uzh-rpg/rpg_dvs_ros

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]
Calibration of a DVS [IROS’14]

How to adjust the focus?

- Use screen blicking pattern such as concentric, logarithmically-spaced, B&W squares

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]
Event-based Vision
Why is Event-based Vision challenging?

- **DVS output is a sequence of asynchronous events** rather than a standard image => A new *paradigm shift* is needed to deal with these data.

- Naive solution: **accumulate events** occurred over a certain time interval and adapt «standard» CV algorithms.
  - Drawback: it *increases latency*

- Instead, we want **each single event** to be used as it comes!
  1. **Lifetime**: for how long is an event active?
  2. **How to do asynchronous**, event-based estimation?
Life-time Estimation

[ICRA’15]

How do we Visualize the Event Stream?

**Naive solution:** accumulate all events occurred in a time interval $\Delta t$
How do we Visualize the Event Stream?

**Naive solution:** accumulate all events occurred in a time interval $\Delta t$

1 video frame = 33 ms (real time)
How do we Visualize the Event Stream?

**Naive solution:** accumulate all events occurred in a time interval $\Delta t$

1 video frame $= 1 \text{ ms}$
How do we Visualize the Event Stream?

**Naive solution:** accumulate all events occurred in a time interval $\Delta t$

1 video frame = 0.5 ms
How do we Visualize the Event Stream? [ICRA’15]

**Naive solution:** accumulate all events occurred in a time interval $\Delta t$

- Large integration time causes **motion blur**
- Small integration time causes **sparsity**

Event Lifetime [ICRA’15]

- Naive method:

- Lifetime (in red): time needed to trigger an event at adjacent pixel [Mueggler’15]

The event lifetime allows determining all events that are active at a specific time. This allows using standard CV algorithms on an event-based fashion.

Surface of Active Events [Benosman, NNL’14]

- Event $e = <x, y, p, t>$
- Surface of Active Events $\Sigma_e(x, y) = t$
  - similar to an elevation map

Benosman, Clercq, Lagorce, Sio-Hoi Ieng, Event-based Visual Flow, IEEE Neural Networks and Learning, 2014
Lifetime estimation of Events [ICRA’15]

- Event velocity on image plane is related to the gradient in the surface of active events:

$$\nabla \Sigma_e(p) = (v_x^{-1}(p), v_y^{-1}(p))^\top$$

- Lifetime of the event:

$$\tau(p) = \|\nabla \Sigma_e(p)\| = \sqrt{v_x^{-2} + v_y^{-2}}$$

Lifetime estimation: Results with a Stripe Pattern [ICRA’15]

- DVS moving on a model train with constant velocity
- Patterns at 0.1m, 0.2m and 5m away from DVS, respectively

\[ \Delta t = 1\text{ms} \]
\[ \Delta t = 30\text{ms} \]

After lifetime estimation

Lifetime estimation: Results with a Stripe Pattern [ICRA’15]

- DVS moving on a model train with constant velocity
- Patterns at 0.1m, 0.2m and 5m away from DVS, respectively

Lifetime estimation: Results from a Drone’s flip [ICRA’15]

- Quadrotor equipped with DVS and standard camera
- Flips with rotational speeds of 1200 deg/s

Flip:

- Quadrotor equipped with DVS and standard camera
- Flips with rotational speeds of 1200 deg/s

Flip:

- $\Delta t = 1\text{ms}$
- $\Delta t = 30\text{ms}$
- After lifetime estimation

Asynchronous, Event-based Vision
[ICRA’14]

[Censi & Scaramuzza, Low Latency, Event-based Visual Odometry, ICRA’14]
Asynchronous, Event-based Vision

- The *event lifetime* is a useful tool to leverage all the events active at a specific time instant
  - Drawback: it *increases latency*

- Instead, we want *each single event* to be used *as it comes*!
  - It allows pose estimation at unprecedented speed, *up to 1MHz*!

- Problem
  - DVS output is a sequence of *asynchronous events* rather than a standard image
  - Thus, a *new paradigm shift* is needed to deal with its data

DVS mounted on a quadrotor AR Drone

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]. Featured on IEEE Spectrum
Application Experiment: Quadrotor Flip (1,200 deg/s)

Video: http://youtu.be/LauQ6LWTkxM

If you intend to use this video, please credit the authors of the paper below, plus the paper.

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]. Featured on IEEE Spectrum
Application Experiment: Quadrotor Flip (1,200 deg/s)

Video: http://youtu.be/LauQ6LWTkxM
If you intend to use this video, please credit the authors of the paper below, plus the paper.

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]. Featured on IEEE Spectrum
Events per time

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]. Featured on IEEE Spectrum
Camera and DVS renderings

Peak Angular Speed: 1,200 deg/s

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]. Featured on IEEE Spectrum
Pose Estimation

- Standard camera: pose at each frame
- DVS: a single event does not provide enough information
  - Need at least 3 events

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]. Featured on IEEE Spectrum
Event-based Tracking Algorithm

- Buffer of $n$ events per side
- When a new event (star) arrives, it replaces the closest event in the buffer (red triangle)
- Reprojection error minimization to estimate new quadrotor pose
- Repeated for every event

$$P^* = \arg \min_P \sum_{l=1}^{4} \sum_{i=1}^{N} \| d(\pi(L_l, P), e_{l,i}) \|^2,$$

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS'14]. Featured on IEEE Spectrum
These errors are comparable with those of a frame-based camera with the same resolution of the DVS and infinite frame-rate!

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]. **Featured on IEEE Spectrum**
Event-based 6DoF Pose-Estimation Results [IROS’14]

- Successful tracking of 24/25 flips up to 1,200 deg/s
- Mean position error: 10.8cm (standard deviation: 7.8cm)
- Mean orientation error: 5.1° (standard deviation: 2.4°)
- Camera resolution is only 128x128 pixels

[Mueggler, Huber, Scaramuzza, Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers, IROS’14]. Featured on IEEE Spectrum
Event-based Pose Estimation from a Photometric Depth Map

[Censi & Scaramuzza, Low Latency, Event-based Visual Odometry, ICRA’14]
Drawbacks of a DVS

- Currently, only the *sign* of the derivative can be measured, but not its magnitude

- **Idea:** Combine a standard camera with a DVS
DAVIS: Dynamic and Active-pixel Vision Sensor [Brandli’14]

Combines the event-driven activity output of the DVS with conventional static frame output of CMOS active-pixel sensors.

Inter-frame, Event-based Pose Estimation [ICRA’14]

- Idea: reduce the problem to “localization” with respect to the previous CMOS frame; assume known depth map

- Solution: Use Bayesian localization
  - Prob. Measurement Model \( p(e_{t,u,v}) \propto |\langle \nabla I, \dot{u} \Delta t \rangle| \)
  - Motion model: we use a constant velocity \((v, \omega)\) model \( \dot{u} = \frac{v}{d} \times p + \omega \)

[Censi & Scaramuzza, «Low Latency, Event-based Visual Odometry», ICRA’14], Featured on MIT News
DVS Operating Principle  [Lichtsteiner, ISCAS’09]

Events are generated any time a single pixel sees a change in brightness larger than $C$

$$\Delta \log I \geq C$$

$L = \log I(t)$

Generative Model [Gallego’15] [Censi’14]

Events are generated any time a single pixel sees a change in brightness larger than $C$ in a time interval $\Delta t$

$$|\Delta \log I| \geq C$$

$$\Delta \log I \approx \frac{\partial \log I}{\partial t} \Delta t$$

If $I(u, t)$ is the intensity function measured by the DVS at a pixel $u = (u, v)$ at time $t$, from the constant-brightness constraint, we have

$$\frac{\partial I}{\partial u} u + \frac{\partial I}{\partial v} v + \frac{\partial I}{\partial t} \Delta t = 0 \Rightarrow \frac{\partial I}{\partial t} + \langle \nabla_u I, \dot{u} \rangle = 0$$

$$|\Delta \log I| \approx |\langle \nabla_u \log(I), \dot{u} \Delta t \rangle| \geq C$$


[Censi & Scaramuzza, Low Latency, Event-based Visual Odometry, ICRA’14]
Generative Model [Censi & Scaramuzza, ICRA’14]

Intuitively, the generative model tells us that the probability that an event is generated depends on the scalar product between the gradient $\nabla I$ and the apparent motion $\mathbf{u}\Delta t$.

\[ |\langle \nabla I, \mathbf{u}\Delta t \rangle| \]


[Censi & Scaramuzza, Low Latency, Event-based Visual Odometry, ICRA’14]
Event-based Pose Estimation, 1D Example (pure rotation)

\[ y_0(\theta) \]

\[ |\nabla y_0(\theta)| \]

\[ \theta \]

\[ \text{pixel} \]

\[ t = 0 \]

\[ \text{time} \]

\[ p(\omega \mid \text{events}) \]

\[ \hat{\omega} \]

\[ \omega = 0 \]

estimated velocity
Event-based 6DoF Pose Estimation Results

RED: observed events; 
GREEN, BLUE: reprojected events (ON, OFF)

Estimated 6DoF pose

[Censi & Scaramuzza, Low Latency, Event-based Visual Odometry, ICRA’14]
Continuous-Time Trajectory Estimation for Event-based Vision Sensors

[Mueggler, Gallego, Scaramuzza, Continuous-Time Trajectory Estimation for Event-based Vision Sensors, RSS’15]
Continuous-Time Trajectories

- Estimate trajectory instead of poses:
  - $T_1, T_2, T_3, ... \rightarrow T(t)$

- **Spline Fusion** [Lovegrove, BMVC'13/IJCV'15]
  - Visual-inertial fusion with rolling-shutter cameras
  - Trajectory is represented with B-splines
  - Cumulative basis functions on SE(3), free from singularities:

\[
T_{w,s}(u(t)) = T_{w,i-1} \prod_{j=1}^{3} \exp \left( \hat{B}_j(u(t)) \Omega_{i+j-1} \right)
\]

Mueggler, Gallego, Scaramuzza, *Continuous-Time Trajectory Estimation for Event-based Vision Sensors*, RSS'15
Continuous-Time Trajectories

- Advantages of continuous-time trajectories
  - **Pose** is well-defined at *any time*
  - Can handle asynchronous, high-frequency data naturally
  - Local support: each event only influences a few control poses

Mueggler, Gallego, Scaramuzza, *Continuous-Time Trajectory Estimation for Event-based Vision Sensors*, RSS’15
Optimization

- Find control poses such that reprojection error of all events is minimized:

$$\{T^*_{w,i}\} = \arg \min_T \sum_k d^2(z_k, l_j(x(t_k)))$$

- Few control poses are needed: 1 control pose per $10^4$ events

Mueggler, Gallego, Scaramuzza, *Continuous-Time Trajectory Estimation for Event-based Vision Sensors*, RSS'15
6DoF Experiments

[IROS’14]: filter

Batch optimization

Ground Truth (Vicon)
Conclusions

➢ **DVS:** revolutionary sensor for robotics:

  - **low-latency** (~1 micro-second)
    - Can enable pose estimation at unprecedented speed
    - Event-based, low-latency control
  
  - **high-dynamic range** (120 dB instead 60 dB)
    - Can enable HDR reconstructions with challenging lighting variations

➢ Very **low bandwidth** (only intensity changes are transmitted)
  - Suitable for hardware implementations

➢ Generative model can be used for filtering-based SLAM solutions

➢ Currently very low resolution (128x128); however soon overcome

➢ Suitable for continuous-time batch optimizations
  - The pose can be evaluated at any time!
A two-level sensing pipeline for future high-speed mobile robotics:

- Standard cameras: Localization and Mapping
- DVS + IMU: agile behavior (evasive maneuver, target tracking, fast re-localization)

Currently working on different problems

- Event-based state-estimation [ICRA’14, IROS’14, RSS’15]
- Tracking [IROS’13, ICRA’15, ECMR’15]
- Collision avoidance [ECMR’15]
Software

- From INILabs
  - DVS software for Windows and Linux (lot of utilities for LED, line, blob tracking, and even processing)
    - [sourceforge.net/p/jaer/wiki/jAER%20Installation/](http://sourceforge.net/p/jaer/wiki/jAER%20Installation/)
    - [sourceforge.net/p/jaer/wiki/jAER%20USB%20Driver%20Install/](http://sourceforge.net/p/jaer/wiki/jAER%20USB%20Driver%20Install/)

- From my lab
  - ROS DVS driver
  - Calibration tools for both intrinsic and stereo calibration:
    - [github.com/uzh-rpg/rpg_dvs_ros](https://github.com/uzh-rpg/rpg_dvs_ros)
References for the Sensors

➢ DVS

➢ DAVIS

➢ BOOK

Shih-Chii Liu  Tobi Delbruck  Christian Braendli  Minhao Yang
Algorithms seen in this tutorial

- **LED Marker Tracking**

- **Probabilistic model and event-based Bayesian localization**

- **Lifetime estimation**

- **Optimization-based localization**
  - E. Mueggler, B. Huber, D. Scaramuzza: *Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers*. IROS’14

- **Collision avoidance**

- **Batch 6DoF localization**
Cognitive Neuromorphic Engineering Workshop

- Every year in Capo Caccia, Sardinia, Italy
- 2 weeks
- 12 working hours a day
- Fully hands-on
Questions?

Wrong believes about DVSes:

- “it’s just another optical-flow sensor”
  - A DVS is not an optical flow sensor! Optic flow is the velocity of a pixel (two components); a DVS pixel only triggers \( \pm 1 \)s if brightness changes

- “A DVS is a camera with a very-high frame rate”
  - There are no frames!
  - A DVS is much faster, consumes less power, has a lower data rate, is much smaller

- “It is of no use because if the scene is very cluttered, all pixels spike”
  - True. Indeed, an event camera is more suitable, for robotics, for scenes with sparse edges
Thanks! Questions?

Funding

SWISS NATIONAL SCIENCE FOUNDATION
erc
Google Faculty Research Awards
Robotics & Perception Group
rpg.ifi.uzh.ch

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederazione svizra
Kommission für Technologie und Innovation KTI

Davide Scaramuzza - University of Zurich – Robotics and Perception Group