Event-based cameras (neuromorphic sensor)

Rodrigo Verschae
Institute of Engineering Sciences
Universidad de O'Higgins
rodrigo@verschae.org

Universidad de O'Higgins

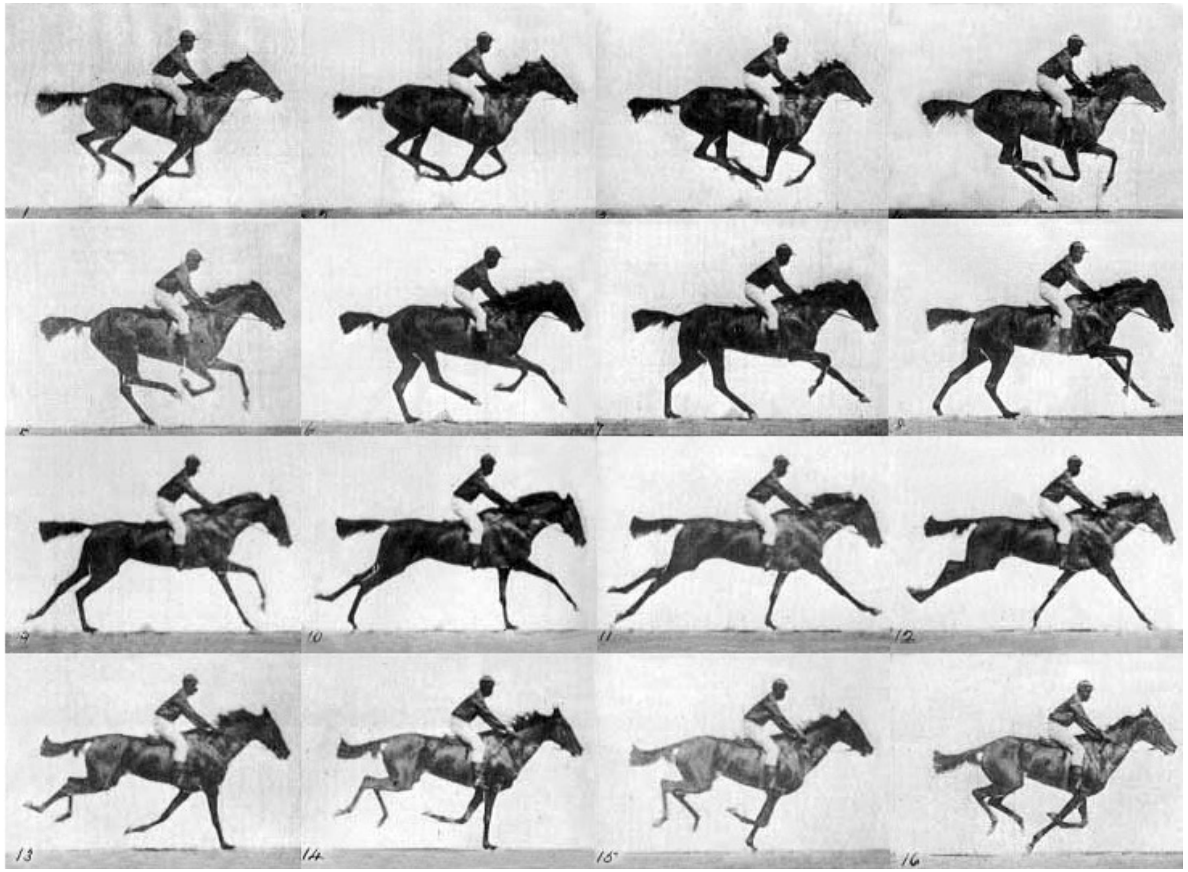
Challenges of traditional “frame-based” cameras

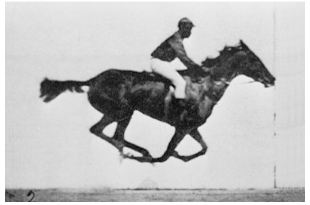
Latency

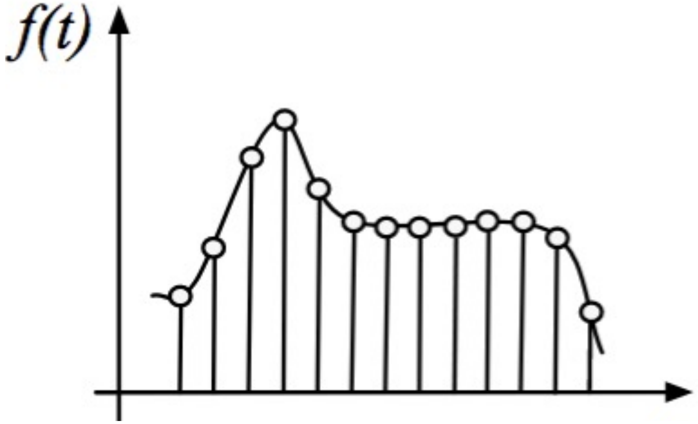
Motion Blur

Dynamic Range

“First” video







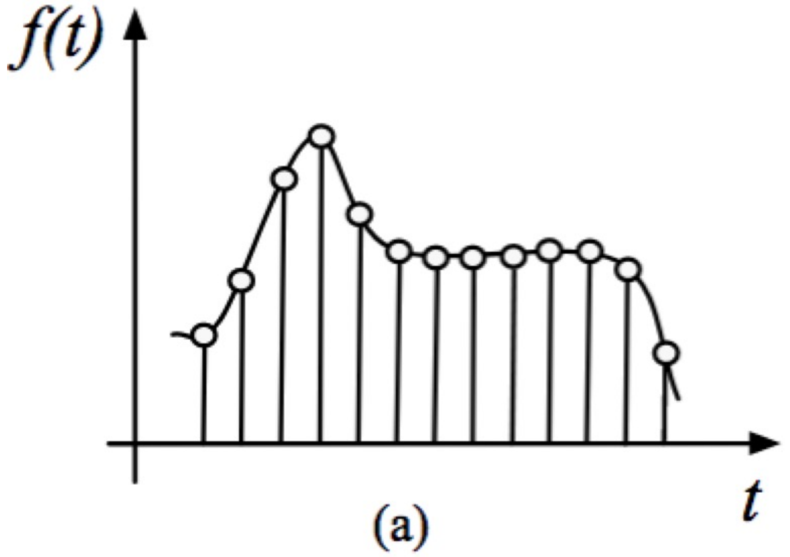
[Horse running by Eadweard-Muybridge, 1878]

https://en.wikipedia.org/wiki/The_Horse_in_Motion

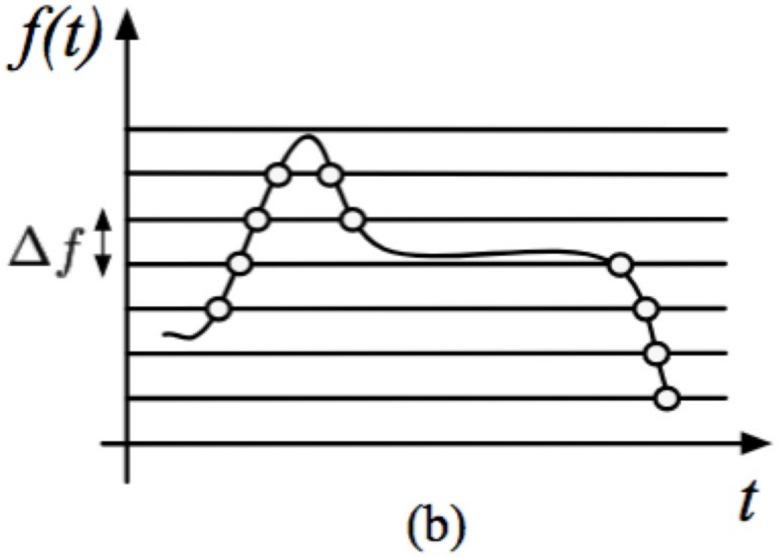
Image Sequence obtained with (standard, frame-based) cameras

Ways to encode/sample information

Time-driven

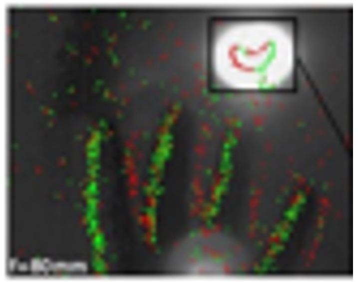


Data-driven



Images from [Clercq 2011].

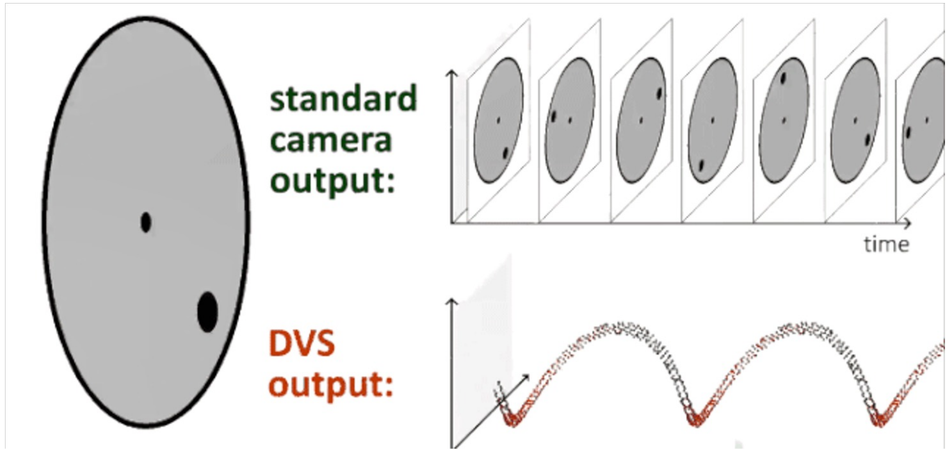
What is an event-based camera?



- Novel sensor that **measures asynchronous pixel intensity change in scenes**
- This sensor generates a **tuple**: *position (x,y), time (t) and binary change of intensity (polarity - p)*
- **First commercialised** in 2008 by T. Delbruck (UZH and ETH)

Features

- Low latency (~ 1 μs)
- No blur motion
- High dynamic range (140 dB instead of 60 dB)
- Ultra-low power (avg: 1mW instead of 1W)
- 1MHz



Event-based Cameras

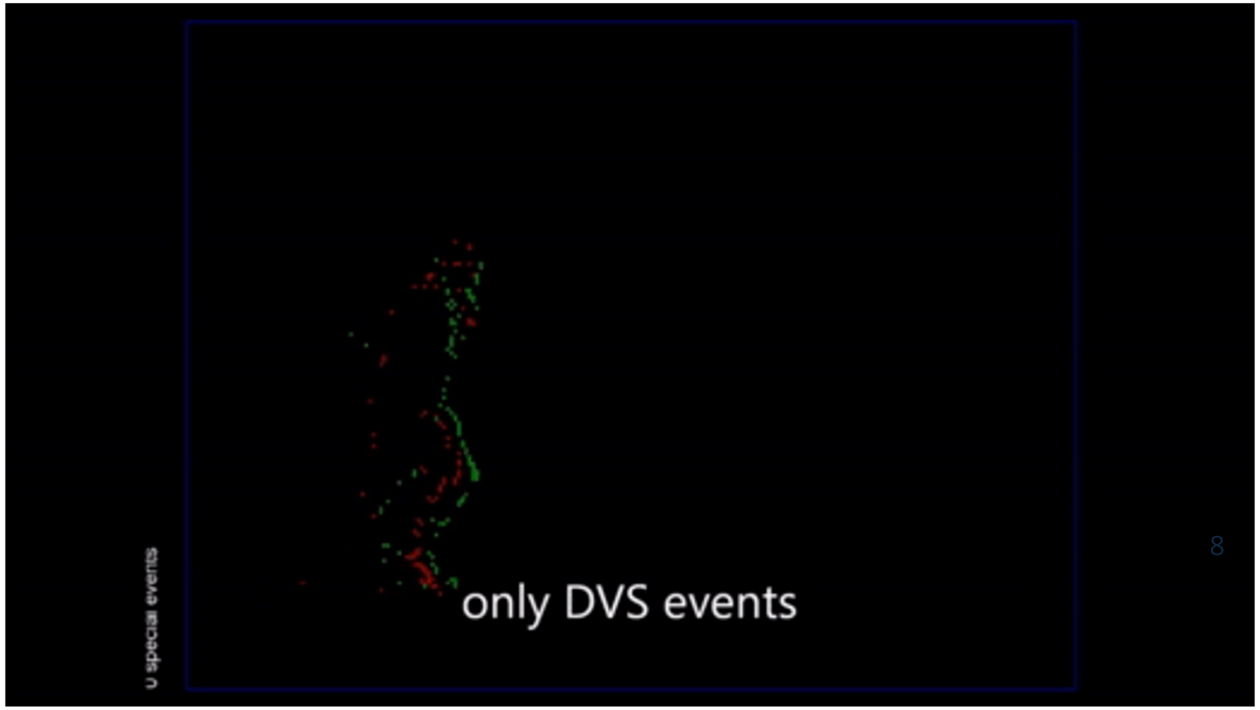
- Bio-inspired sensors
- Asynchronously and independently measure brightness changes in each pixel.
- Advantages of event-based cameras:
  - High temporal resolution (~1μs)
  - Low latency (~10μs)
  - High Dynamic Range (>120 dB vs. 60 dB)
  - Low Power Consumption
- Dynamic Vision Sensor (DVS)
  - Commercially available since 2008

(a) image (b) events

Event-based Cameras: example

Video by Tobi Delbruck. From https://inivation.com/developer/videos/

Event-based Cameras: example



Video by Tobi Delbruck. From https://invivation.com/developer/videos/

Event-based Cameras: example

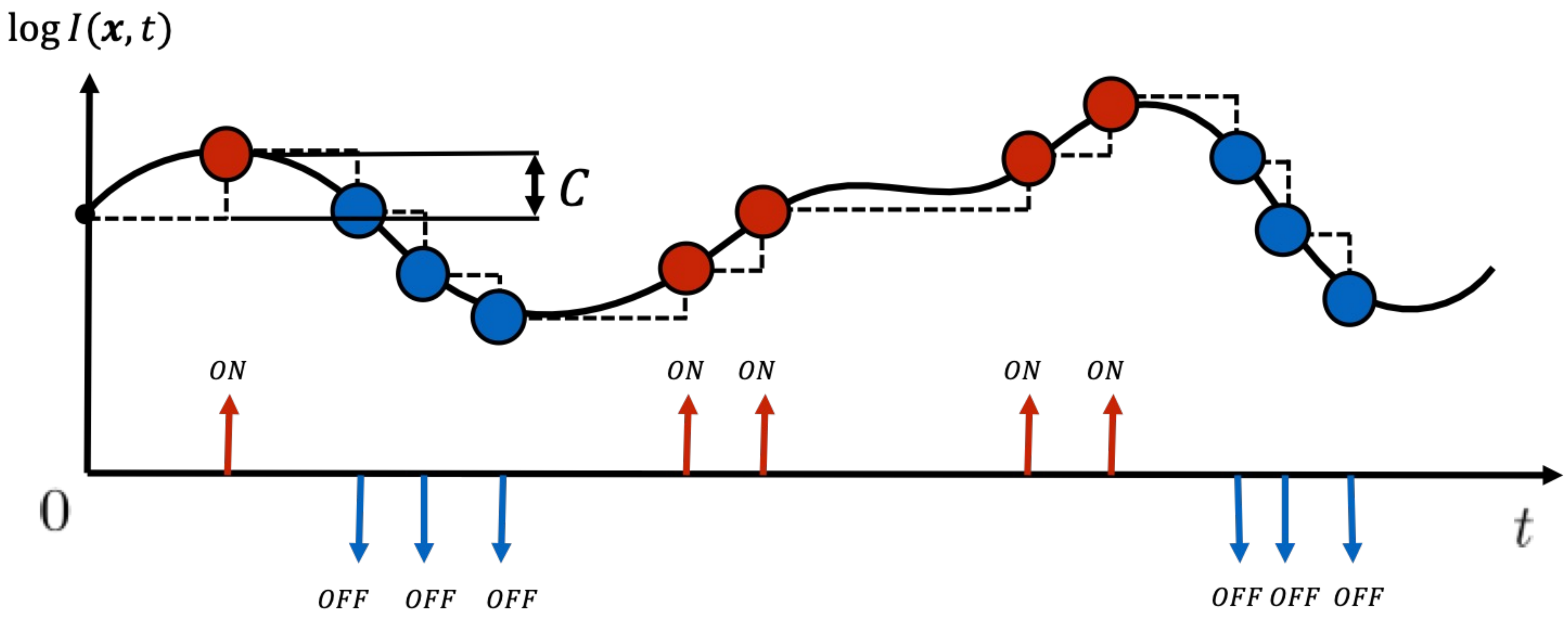
Video by Tobi Delbruck. From https://inivation.com/developer/videos/

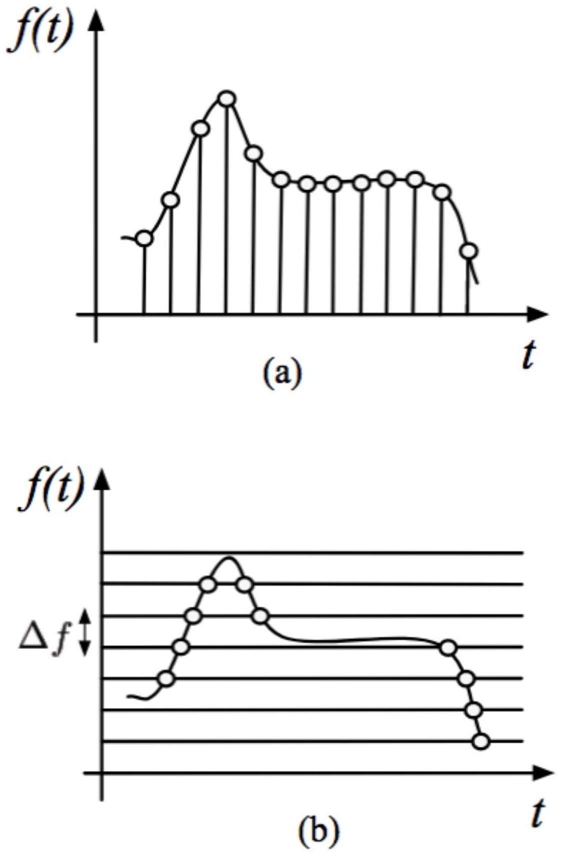
Did you notice the blinking?

https://www.youtube.com/watch?v=fLhbYARLBbk

Event Generation Model

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



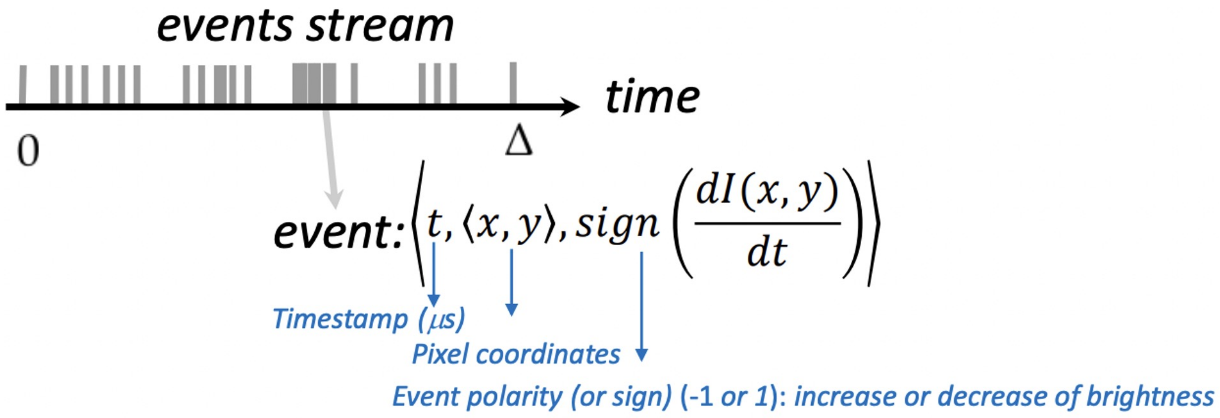


[Gallego et al, Event-based Vision: A Survey, 2019]

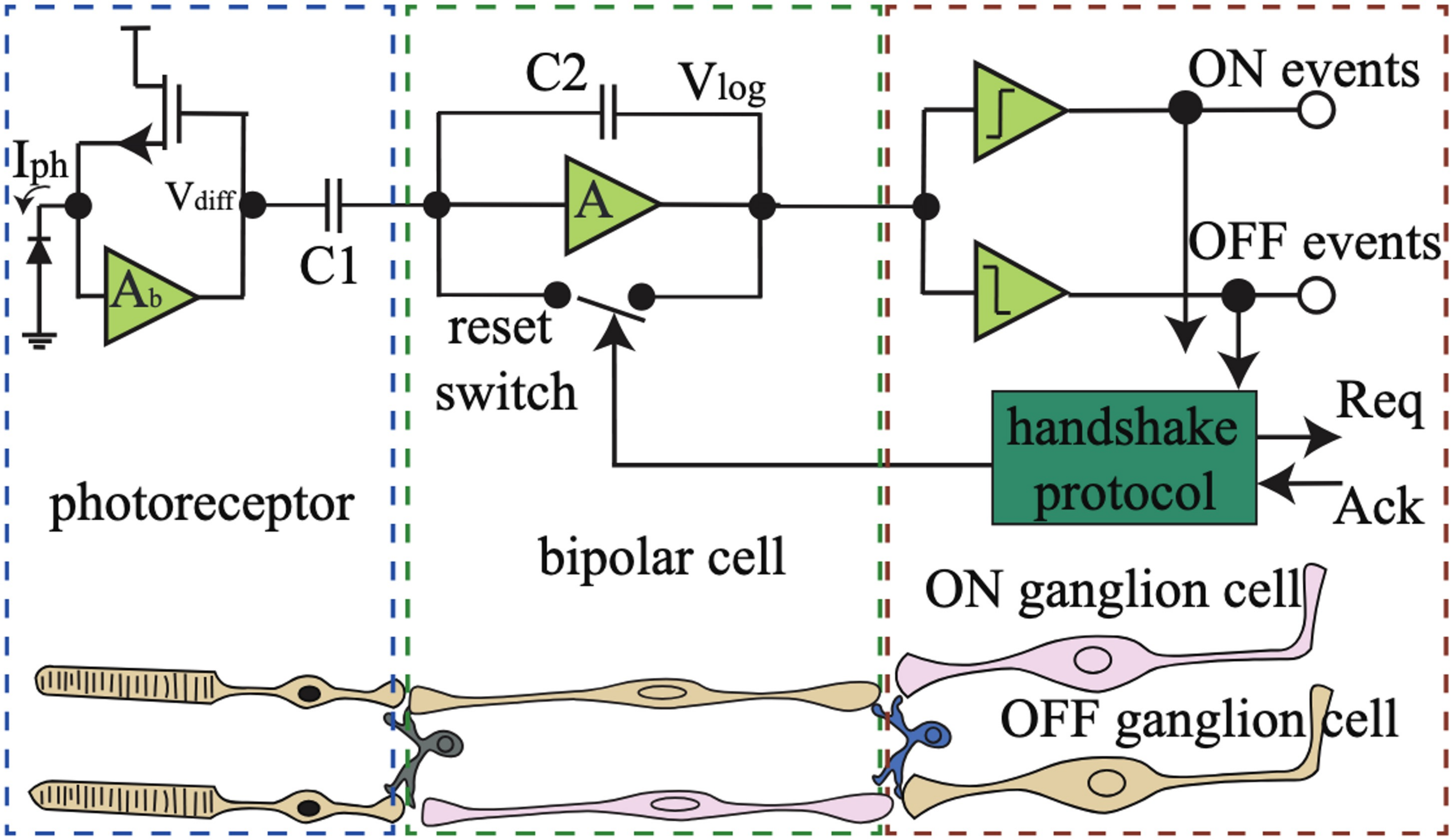
Frame-based camera vs Event-based camera

- A **conventional frame-based camera** produces frames at **fixed time intervals**:

- In contrast, an **event-based camera** produces **asynchronous events** with a ***resolution of microseconds***. An event is generated each time a single pixel detects an intensity change value:



Three-layer model of a human retina and its DVS pixel circuit

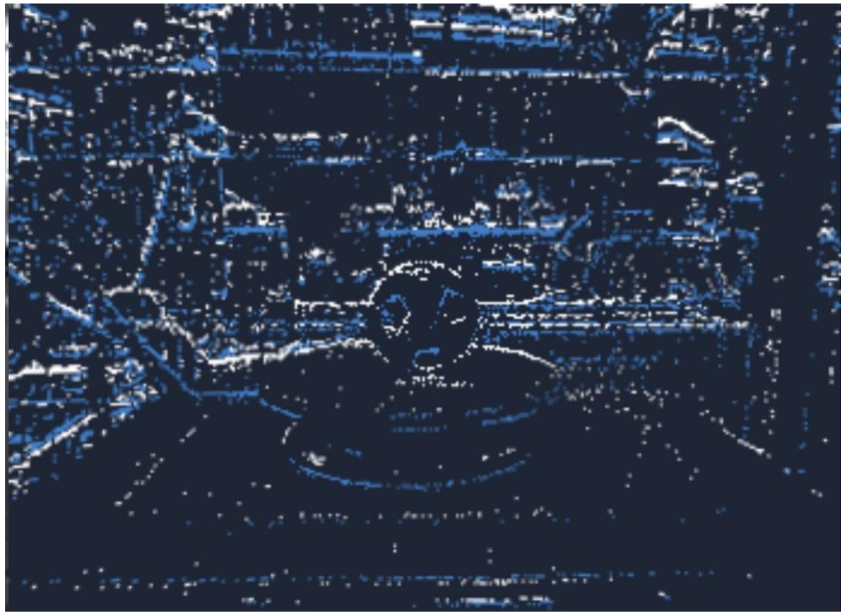


Event-based cameras: What triggers the events?

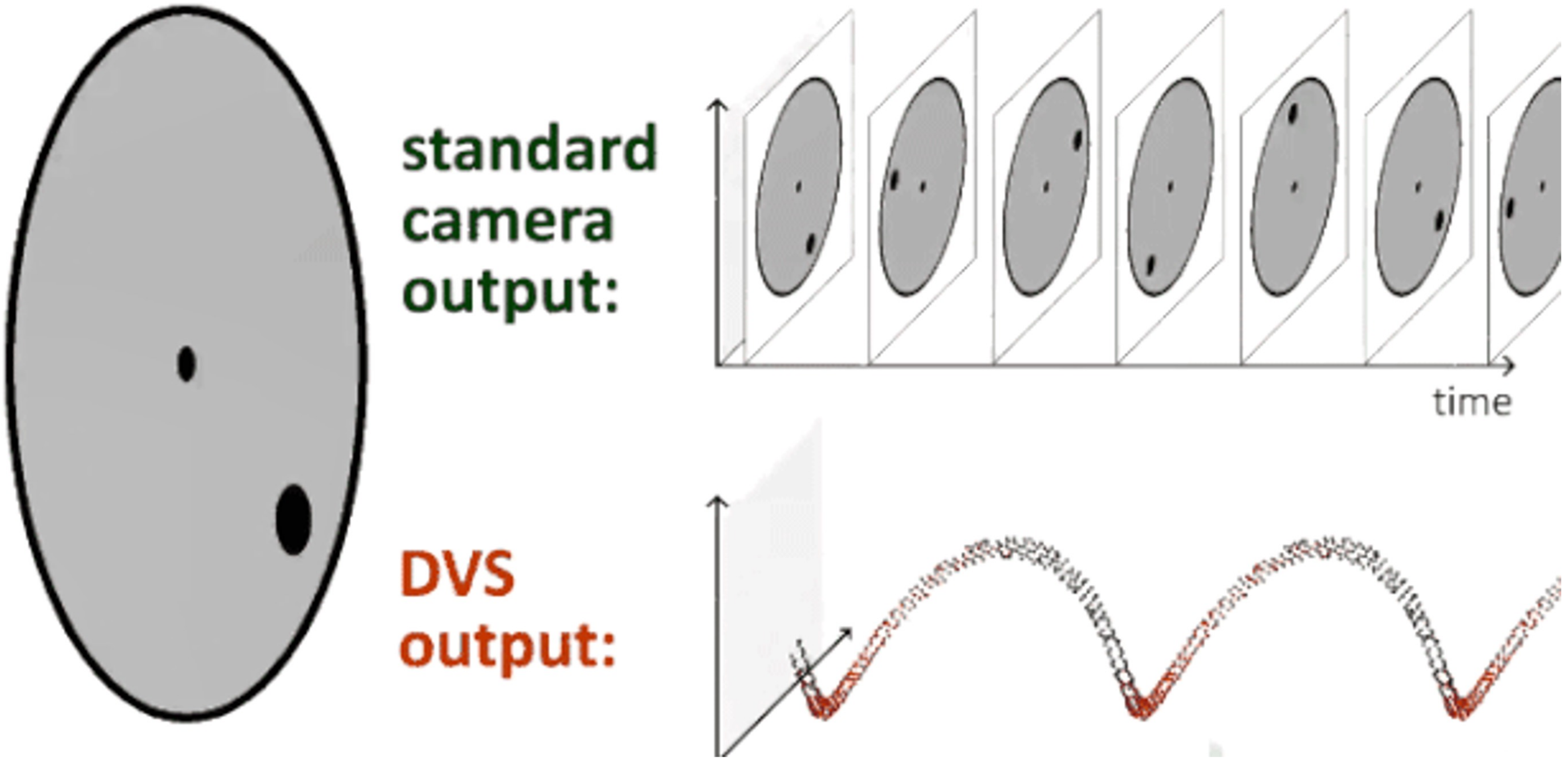
- Events are caused by moving edges
- When the camera moves, events are triggered *"everywhere"*

Frame-based camera

Event-based camera (ON-OFF events)

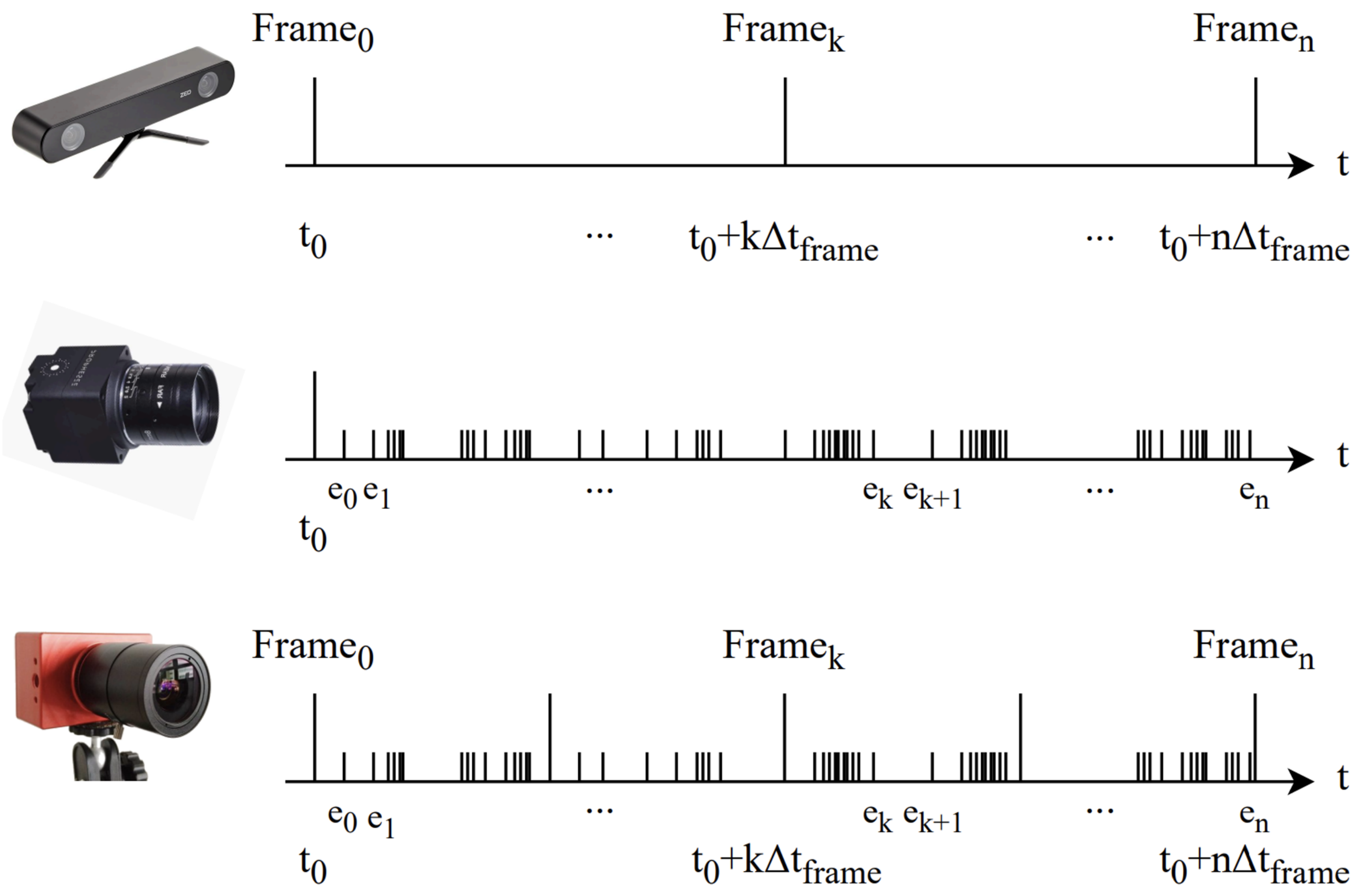


Frame-based camera vs Event-based camera



Gallego, G.: Event-based Robot Vision. In: 2020 TU Berlin, Germany.

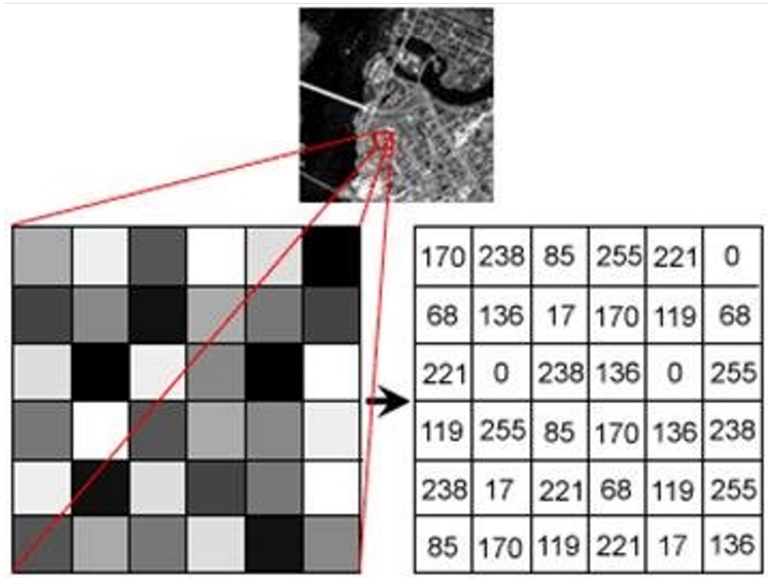
Frame-based camera vs Event-based camera



I. Bugueno-Cordova, R. Verschae, J. Ruiz-del-Solar "Moving6DPoSe: A Multi-Camera Database for Real-Time 6D Pose Estimation and Segmentation of Moving Objects," submitted to IEEE Robotics and Automation Letters

Data from camera:

Standard camera



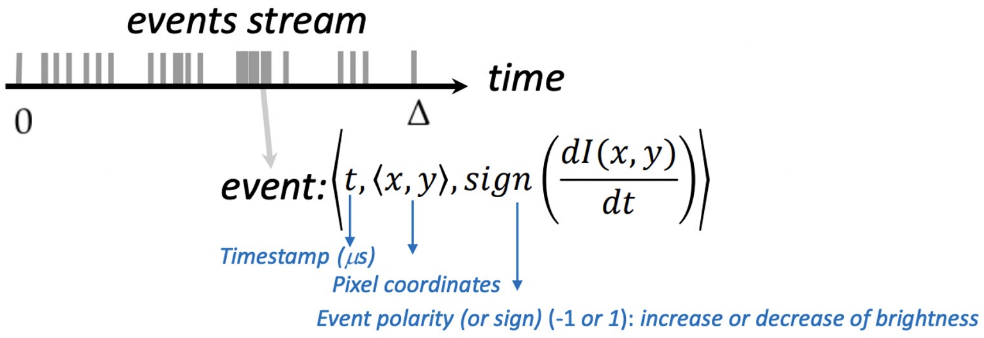
Frame / matrix

Event-based Camera

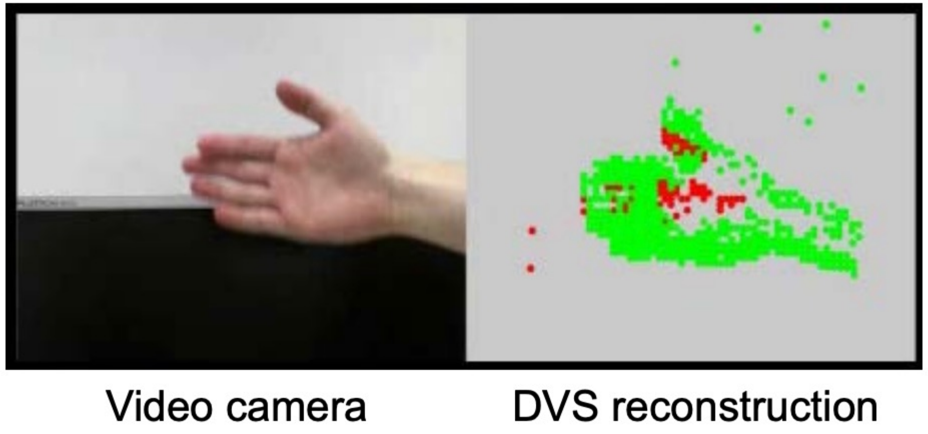
(x)	(y)	(p)	(time)
:	:	:	:
50	112	1	13437757
43	114	0	13437762
73	18	1	13437766
62	57	0	13437768
47	123	1	13437774
75	65	0	13437780
64	55	1	13437784
47	118	0	13437790
50	111	1	13437792
43	113	0	13437793
49	112	0	13437799
51	109	1	13437801
50	107	0	13437805
56	88	1	13437820
47	109	0	13437823
59	69	0	13437830
50	92	0	13437843
75	17	0	13437847
49	116	1	13437852
50	105	0	13437855
:	:	:	:

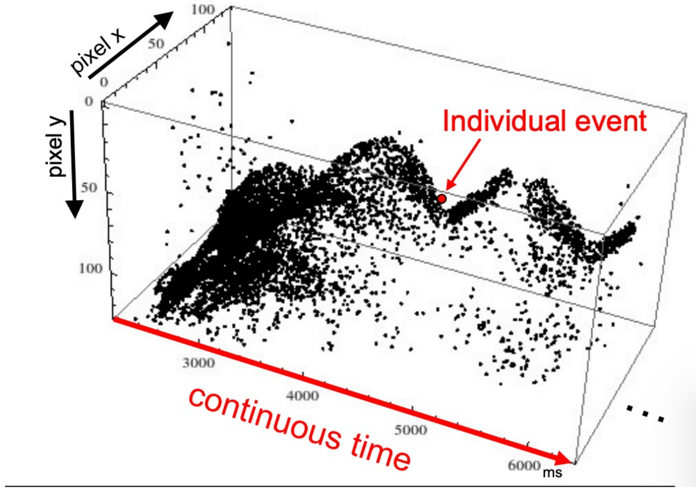
Event package / stream

$$L(\mathbf{u}_k, t_k) - L(\mathbf{u}_k, t_k - \Delta t_k) \geq p_k C, \qquad (1)$$



Event Visualization





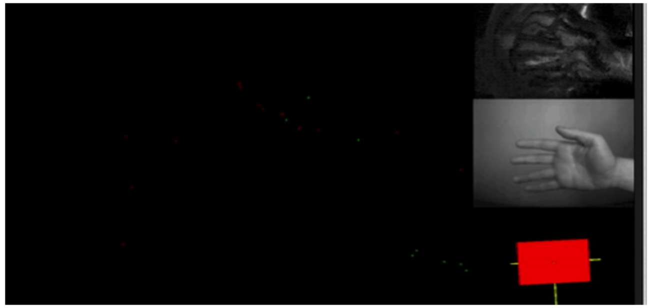
Left: Image of a hand obtained with a frame-based camera.
Center: Output of an event-based camera accumulated in the image plane.
Right: Output of an event-based camera in the space-time domain.

All representations correspond to the same moving hand.

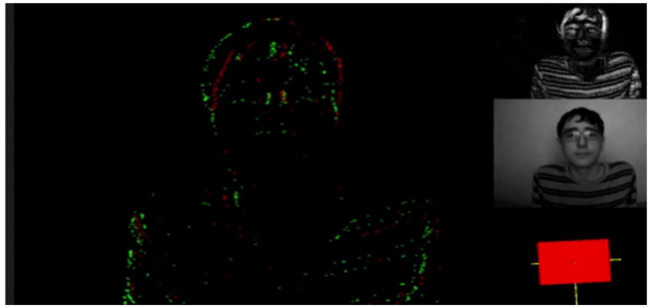
Images From
[http://rpg.ifi.uzh.ch/docs/ICRA17workshop/Conradt.pdf]

Event Camera: High dynamic range & no blur

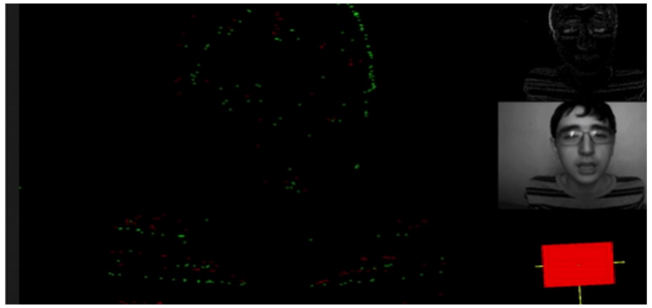
Day - Hand



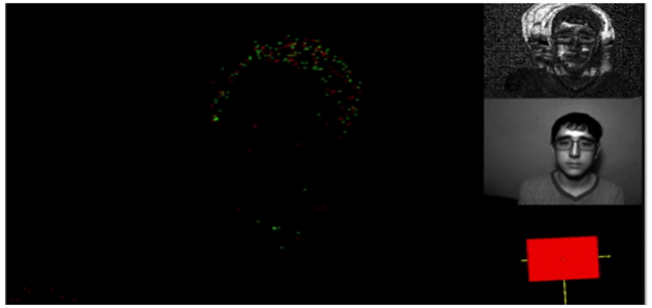
Day - Blink



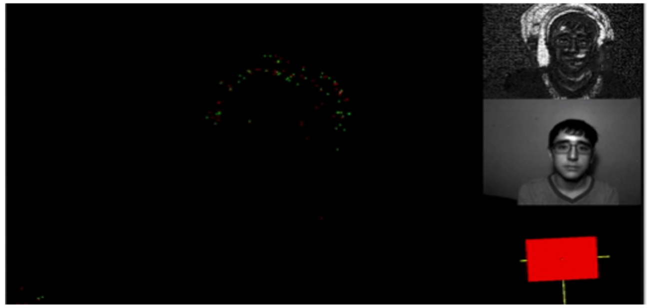
Day - Talking



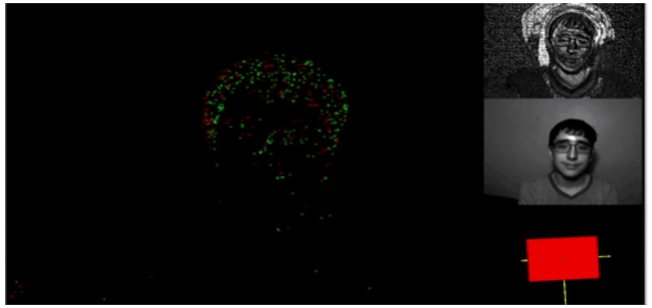
Night - Generic Move



Night - Blink

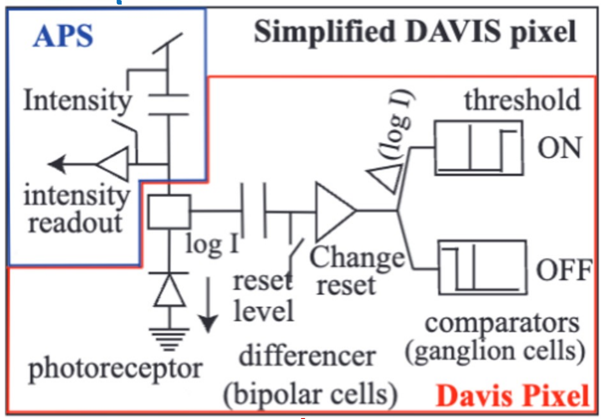


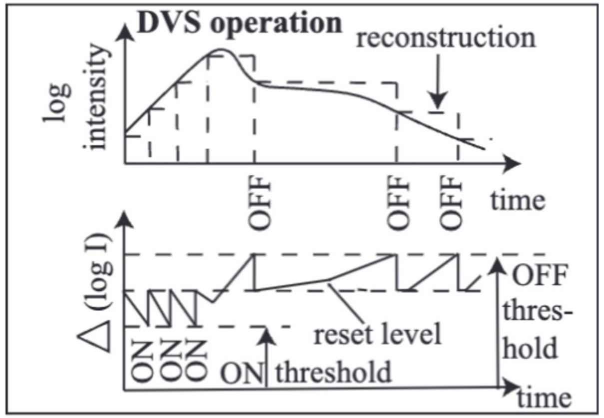
Night - Expressions

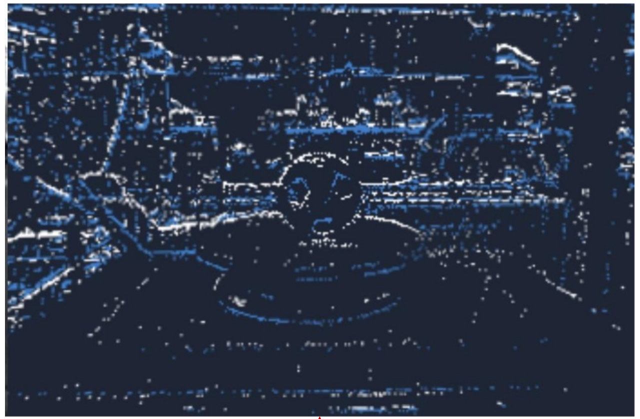


Event-based cameras working principle

Frame (APS) generation



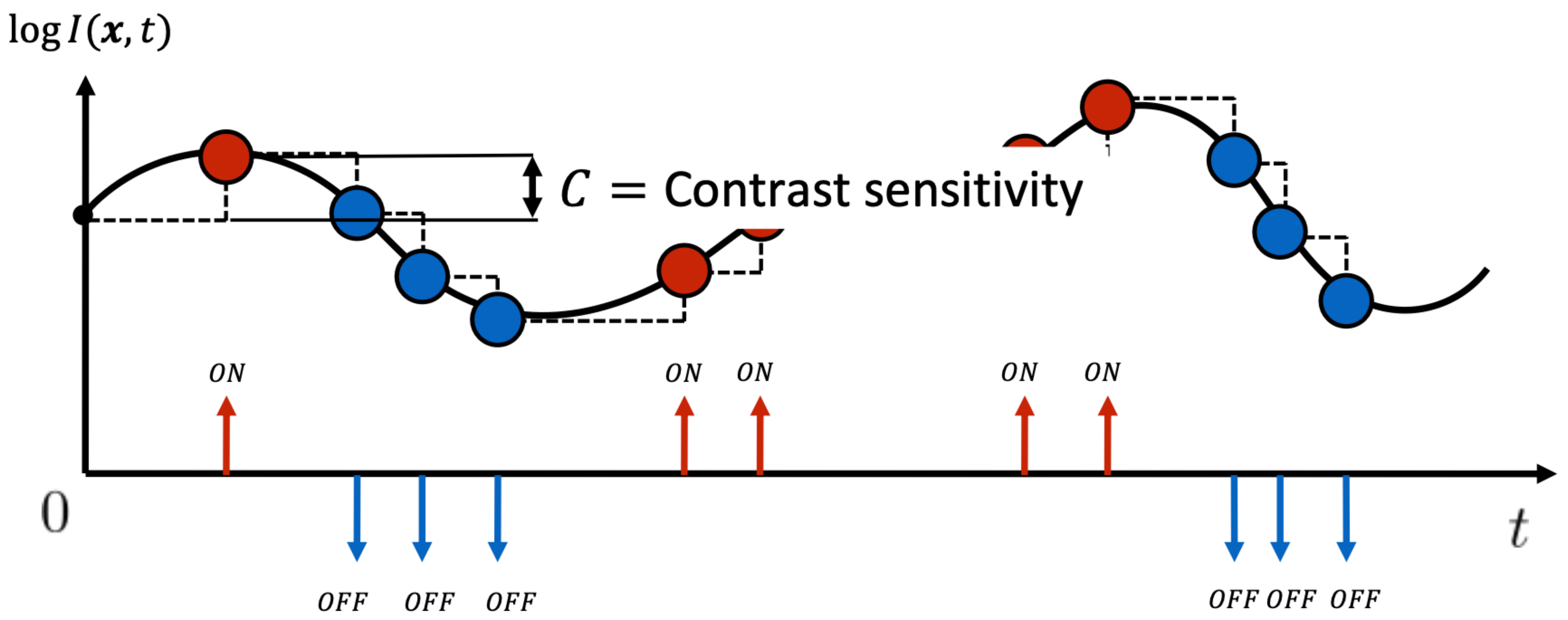




Asynchronous events generation

Event Generation Model

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

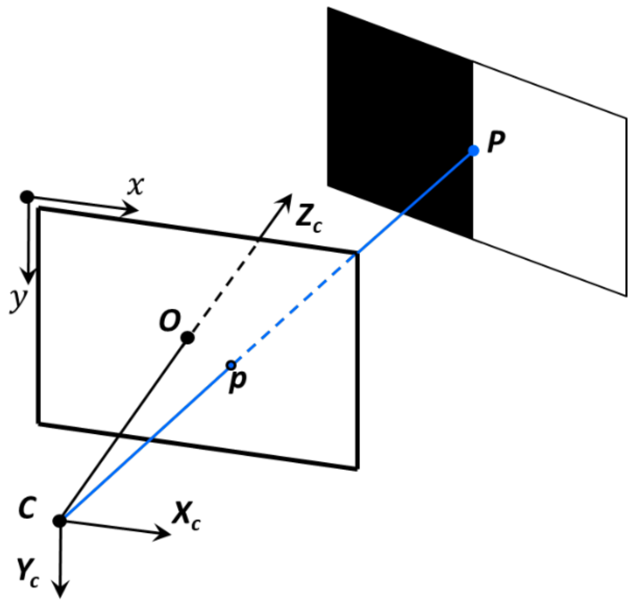


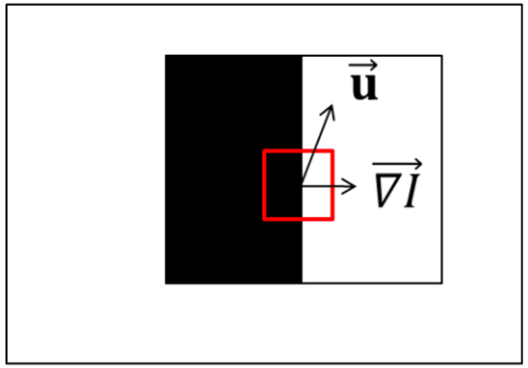
Events are triggered **asynchronously**

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:

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





Gallego et al., Event-based Vision: A Survey, arXiv, 2019. PDF

Proof

The proof comes from the ***brightness constancy assumption***, which says that 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 $t + \Delta t$, we get:

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

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

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

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.

From Gallego et at. Event-based Vision: A Survey, 2020

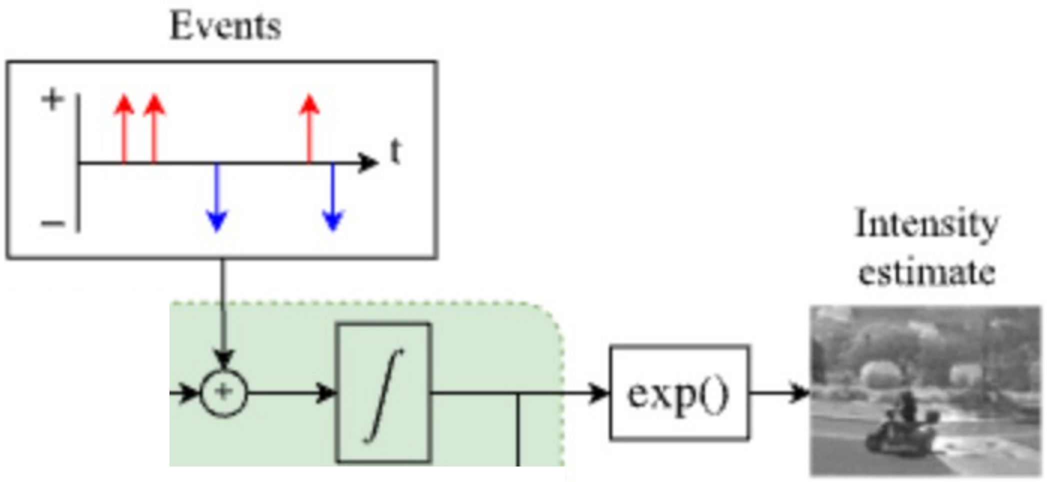
Pixel reconstruction

Pixel Generation Model

$$\boxed{L(\mathbf{u}_k, t_k) - L(\mathbf{u}_k, t_k - \Delta t_k) \geq p_k C,}$$

with $\quad L(\mathbf{u}_k, t_k) \doteq \log(I(\mathbf{u}_k, t_k)), \qquad \mathbf{u}_k = (x_k, y_k)^T$

Simple pixel reconstruction by Integration (no noise model)

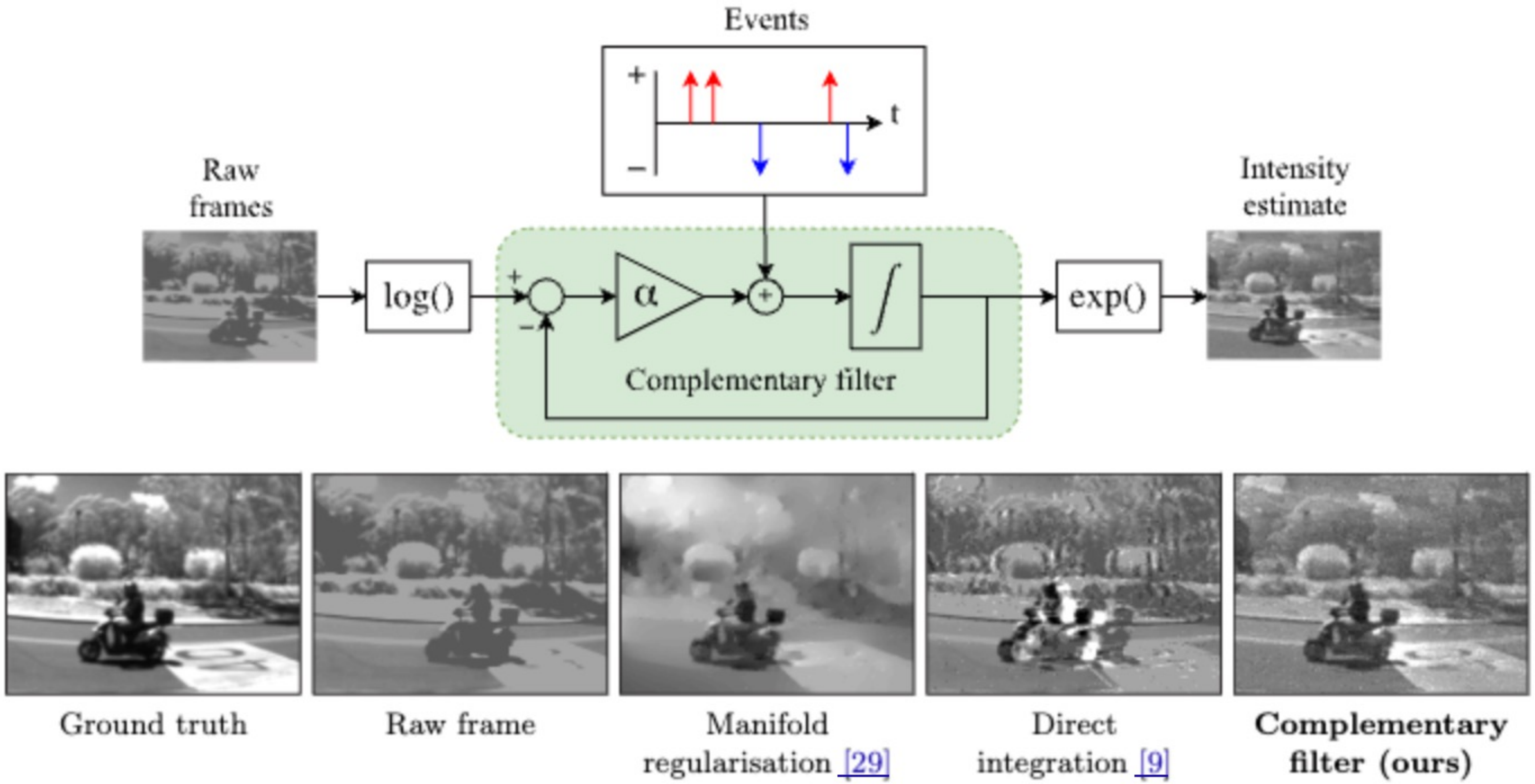


$$\sum L(u_k, t_k) - L(u_k, t_k - \Delta t_k) = \sum p_k C$$

$$L(u_k, t_N) - L(u_k, t_0) = \sum_{k=0}^{N} p_k C$$

$$\boxed{I(u_k, t_N) = e^{C \sum_{k=0}^{N} p_k} + I(u_k, t_0)}$$

Pixel reconstruction



Continuous-time Intensity Estimation Using Event Cameras by Cedric Scheerlinck et al, 2018

Event-based camera: some applications

Visual-Inertial Odometry

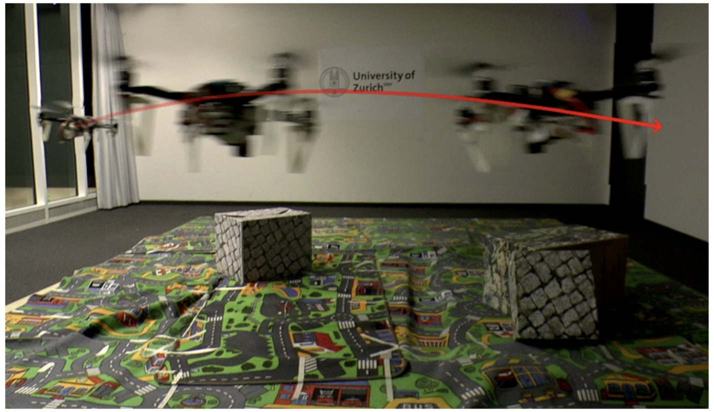
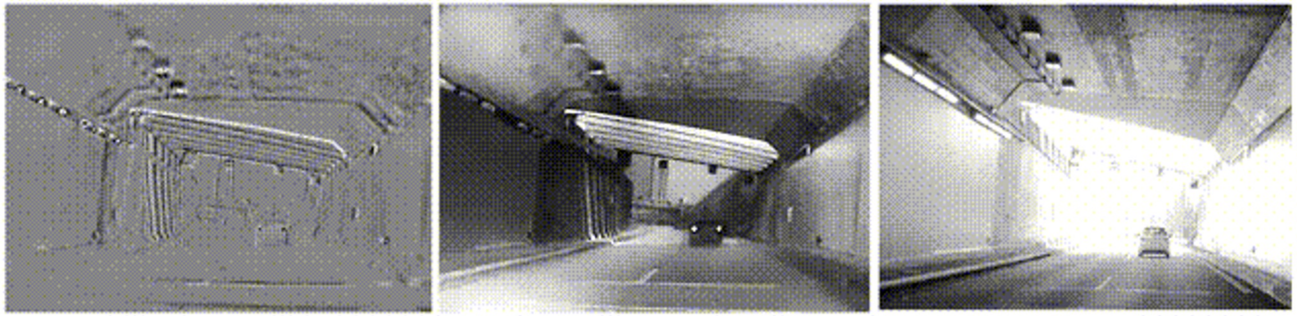


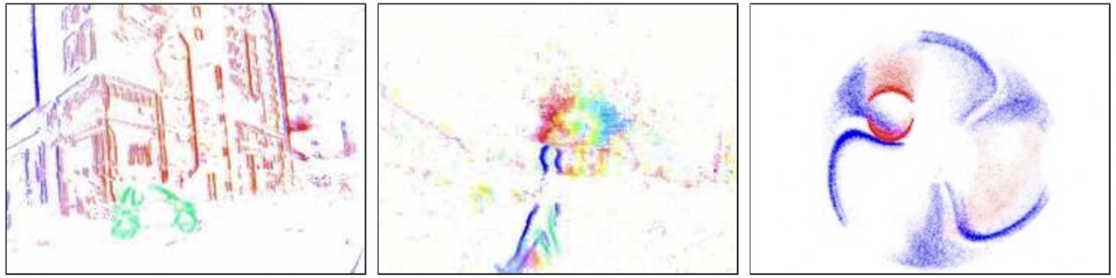
Image Reconstruction



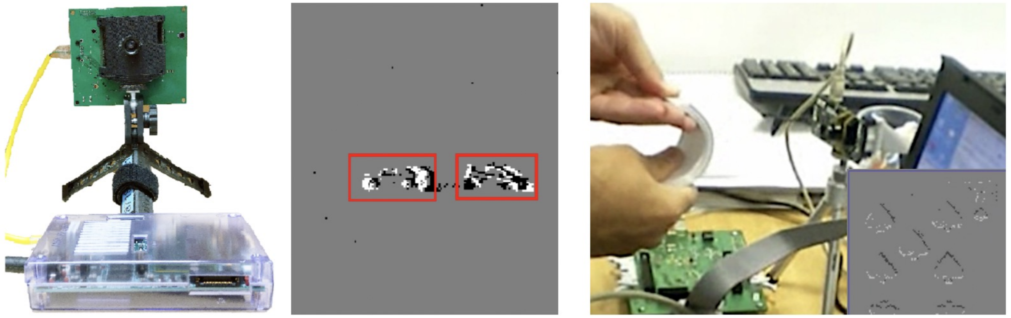
Events | Our reconstruction | Phone camera

Motion Segmentation

Buildings and car | Street, facing the sun | Fan and coin



Recognition

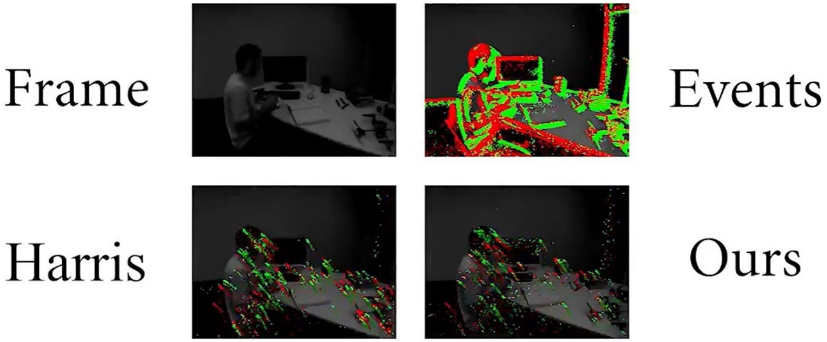


(a) Event camera and IBM TrueNorth | (b) Poker-DVS

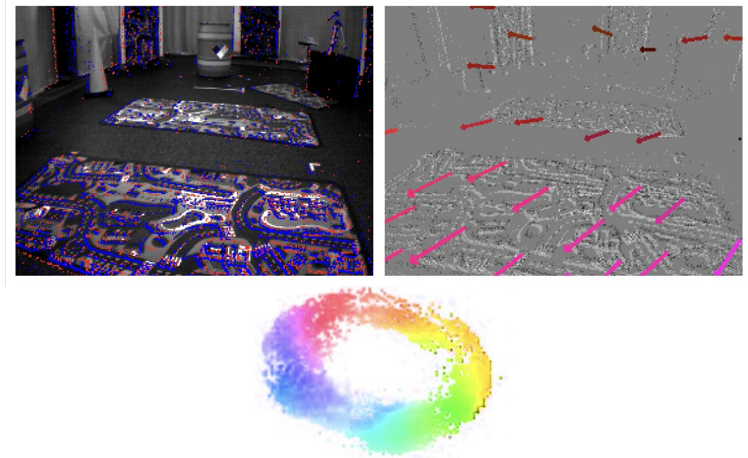
Event-based camera: some applications

Feature Detection and Tracking

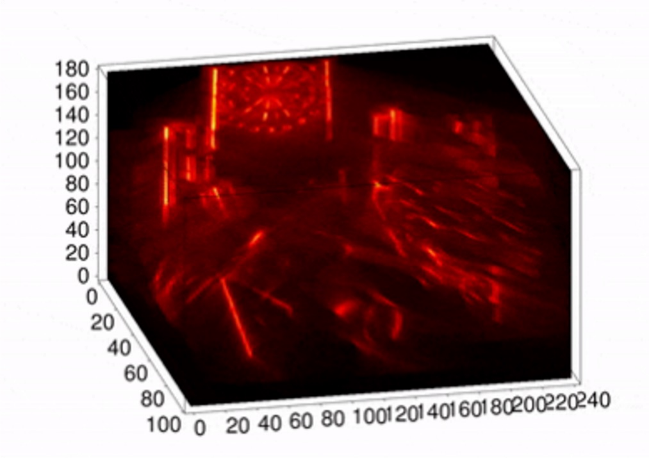
FAST Event-based Corner Detector



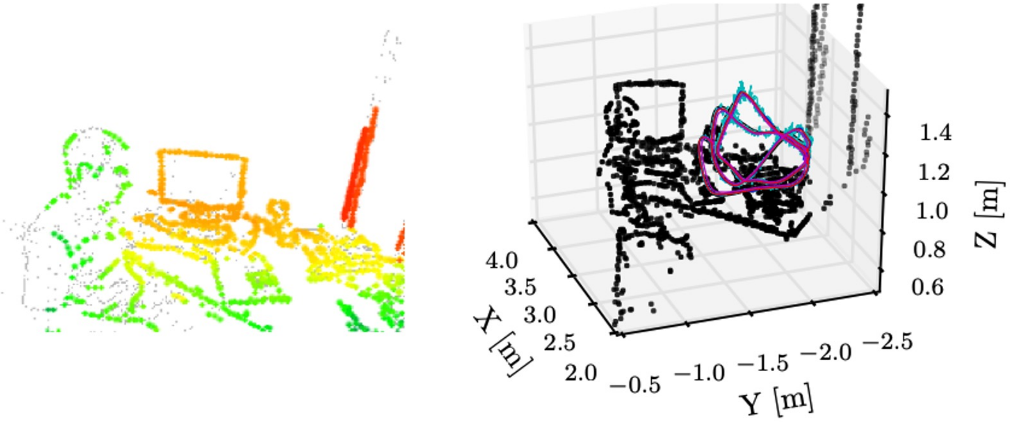
Optical Flow Estimation



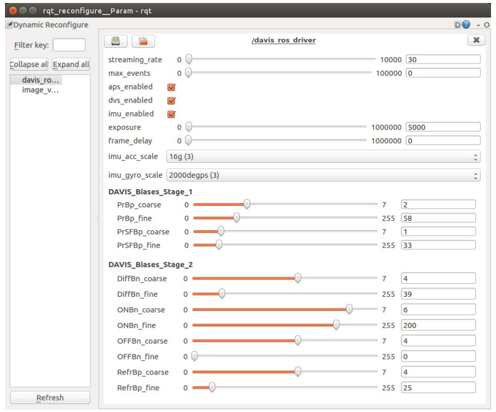
3D Reconstruction: Monocular and Stereo



Pose Estimation and SLAM



PROPHESEE SDK

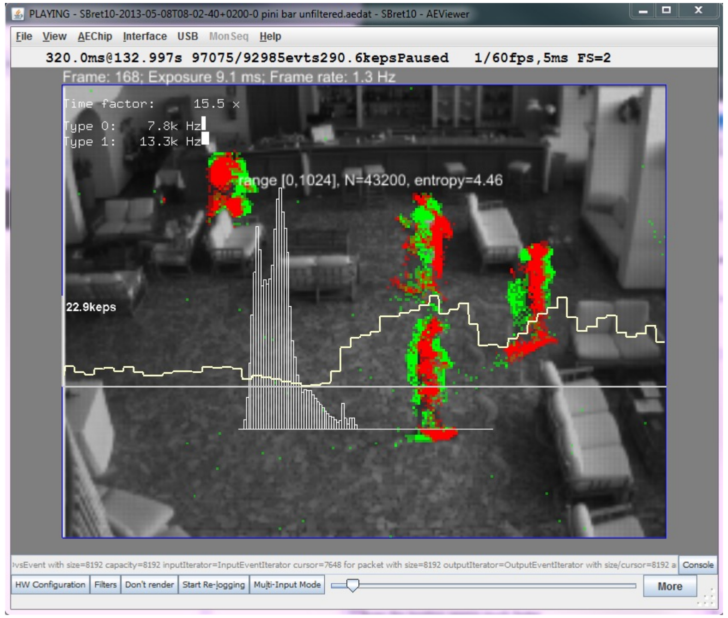


Some Software tools

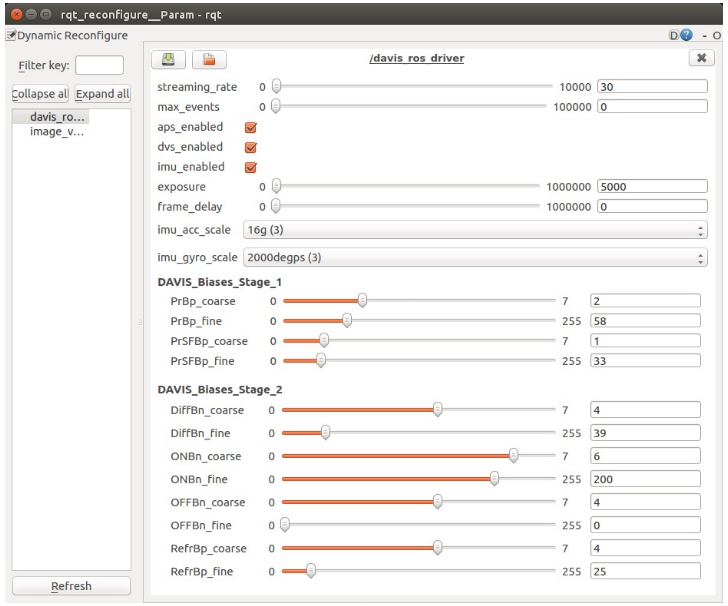
To date, there is no standard open source library integrated into OpenCV that provides algorithms for event-driven vision.

However, there are many well-developed open source software resources

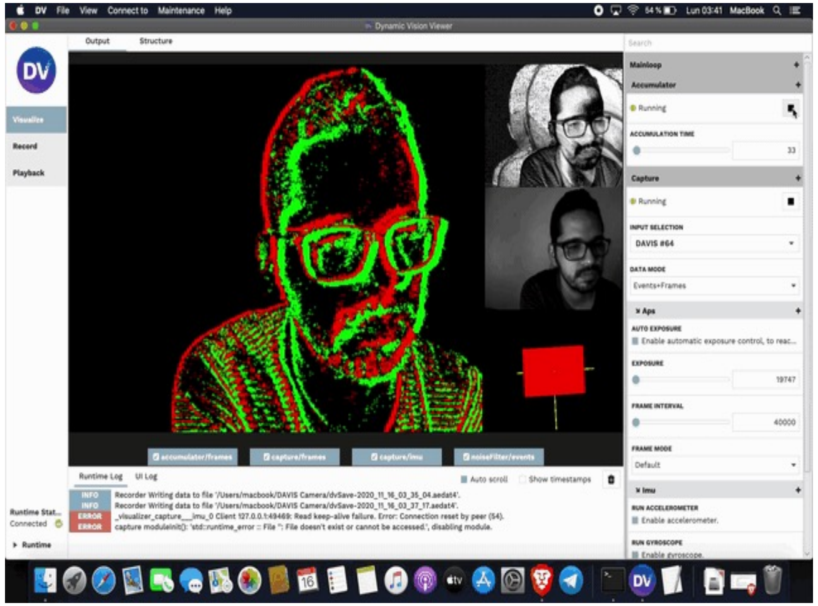
jAER [23]



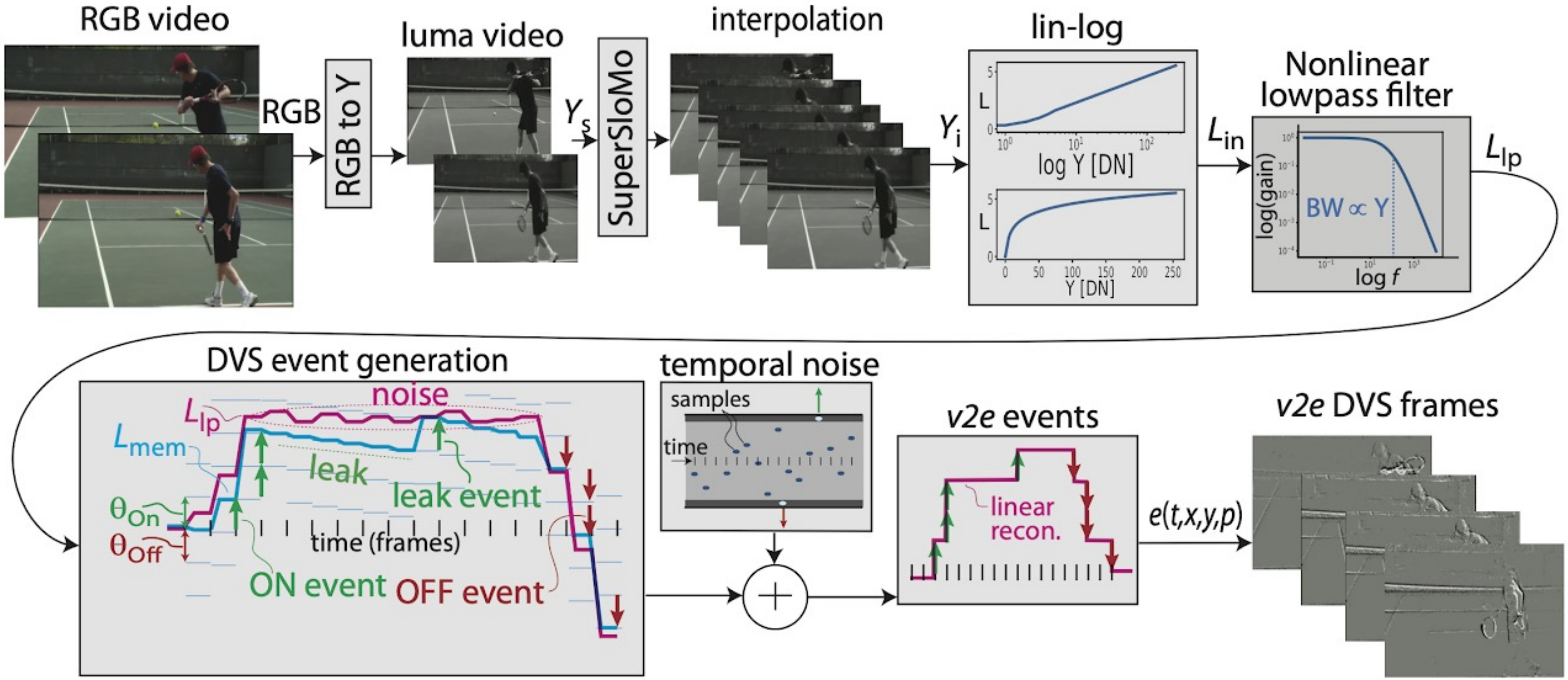
ROS DVS Package [24]



DV Software [25] & DV Python [26]



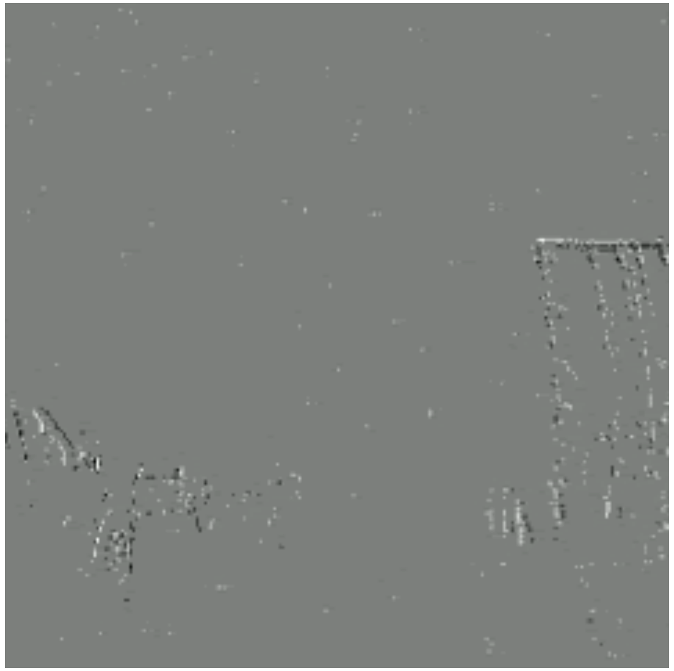
v2e: Event generation from video frames



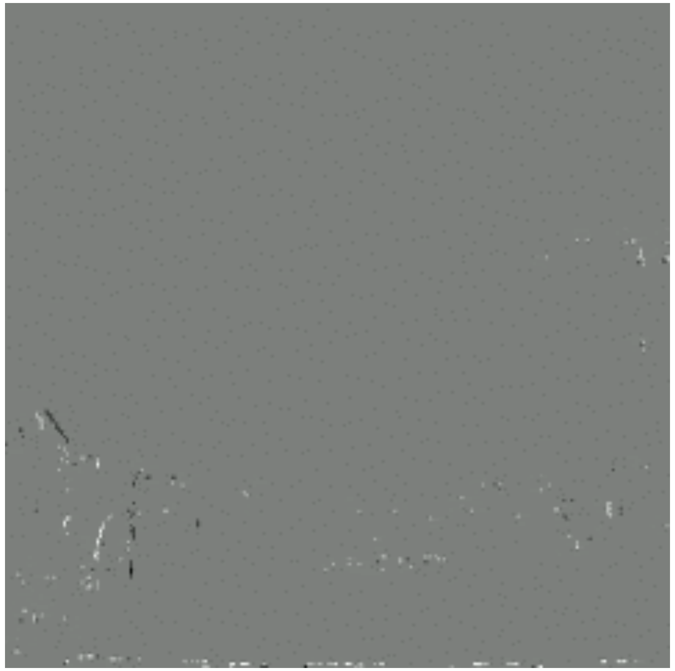
[43]

Event generation from video frames

Input frames



Ground-truth DVS events



Emulated DVS events

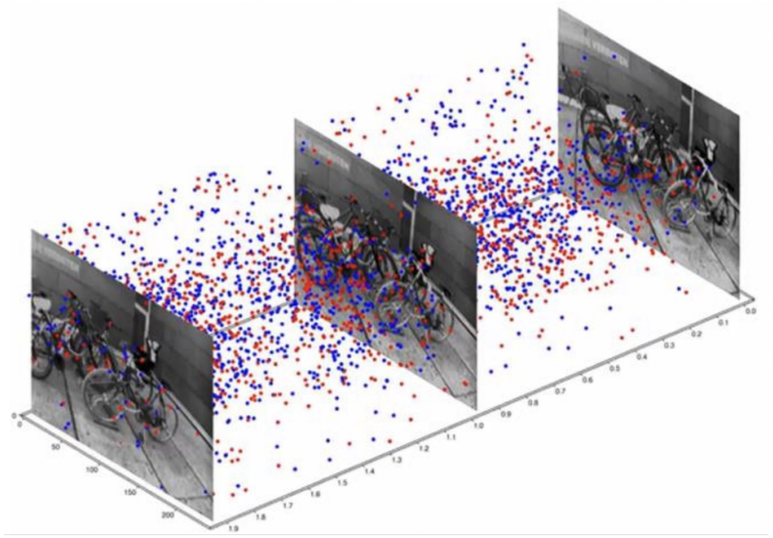
[43]

Event representations

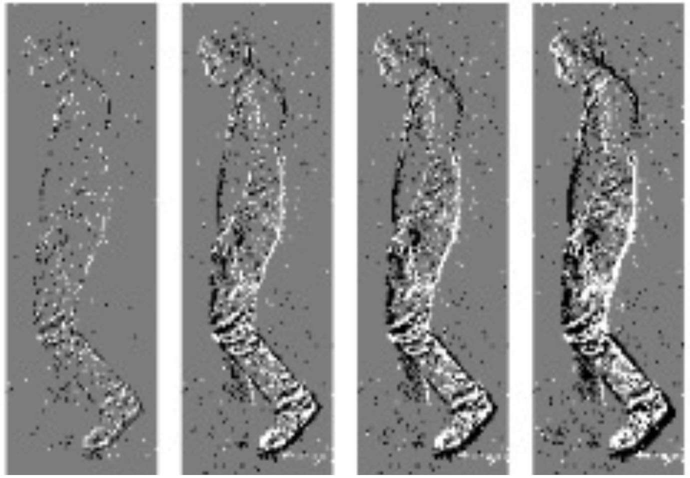
One of the key aspects of event cameras is **how** to extract meaningful information from the data, considering temporality.

Depending on the number of events being processed simultaneously, there are:

I. Methods that operate on an **event-by-event**, where the state of the system can change with the arrival of a single event, achieving minimal latency
II. Methods that operate on **groups or packets of events**, which introduce some latency.



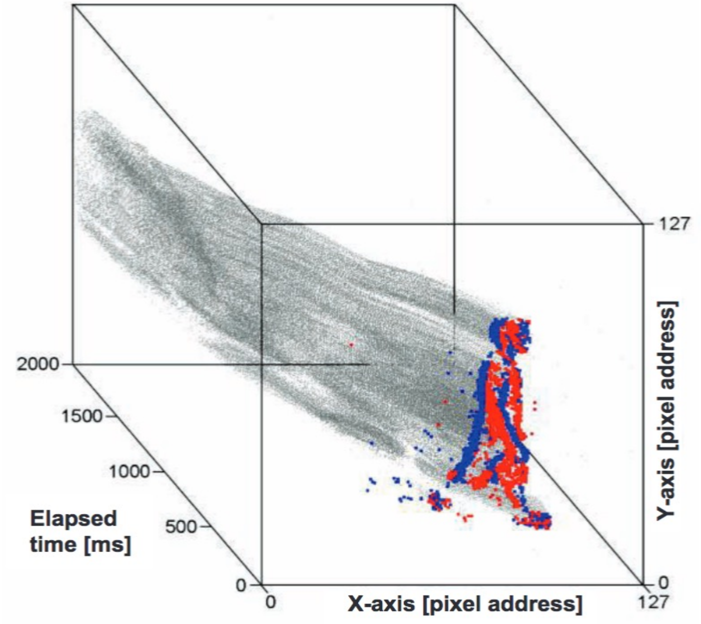
How to handle the events? How to represent these spatio-temporal information?



Events accumulated over various periods of 60ms, 40ms, 30ms and 10ms.

The right time span for analysis has to be determined.

Image from [Eibensteiner 2017].



Person crossing the field of view in space-time domain.

Image from [Wiesmann 2012].

Types of event representations

Individual events

Utilizados por los métodos de procesamiento evento por evento, como los filtros probabilísticos y las SNN

Event packet

Los eventos en una vecindad espacio-temporal se procesan juntos para producir una salida

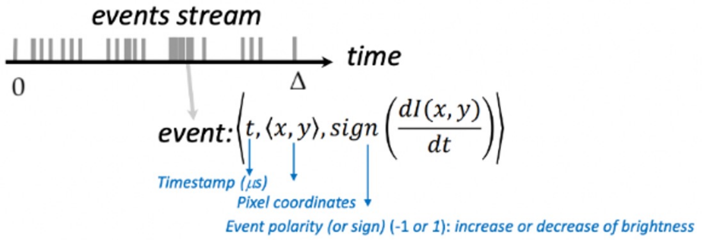
Event frame

Los sucesos en una vecindad espacio-temporal se convierten en una imagen para alimentar algoritmos de CV

Time surface

Mapa 2D en el que cada pixel almacena un único valor temporal. Los eventos se convierten en una imagen cuya "intensidad" es una función del historial de movimiento en ese lugar

$$I(\mathbf{x}, t) \doteq \exp\left(-\frac{t - t_{last}(\mathbf{x})}{\delta}\right)$$



Voxel Grid

Histograma espacio-temporal (3D) de eventos, donde cada vóxel representa un píxel y un intervalo de tiempo concreto

3D point set

Los eventos en una vecindad espacio-temporal se tratan como puntos en el espacio 3D

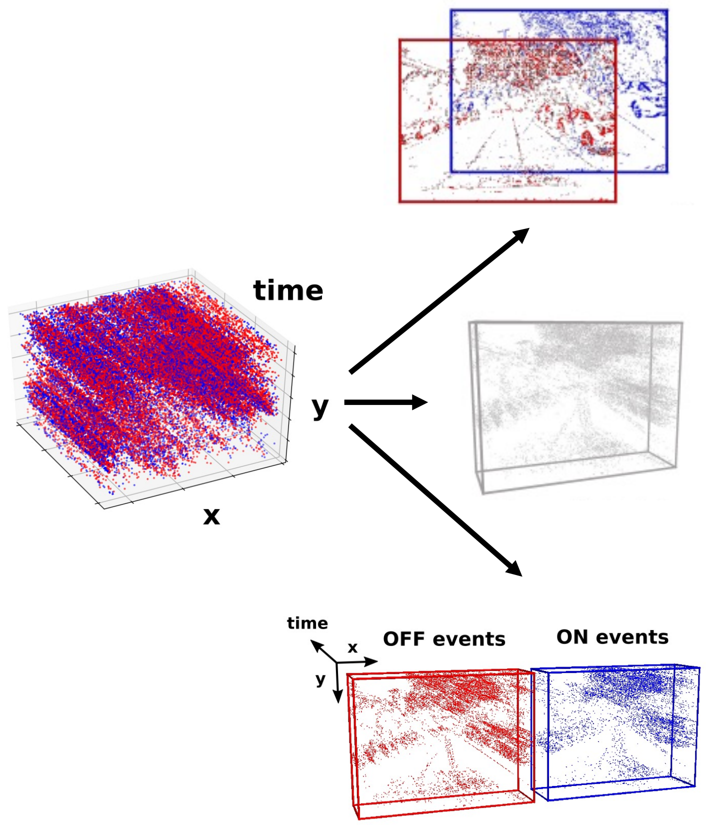
Point sets on image

Los eventos se tratan como un conjunto evolutivo de puntos 2D en el plano de la imagen

Motion event image

Representación que depende no sólo de los eventos, sino también de la deformación y movimientos de los mismos.

Representation: Some Examples



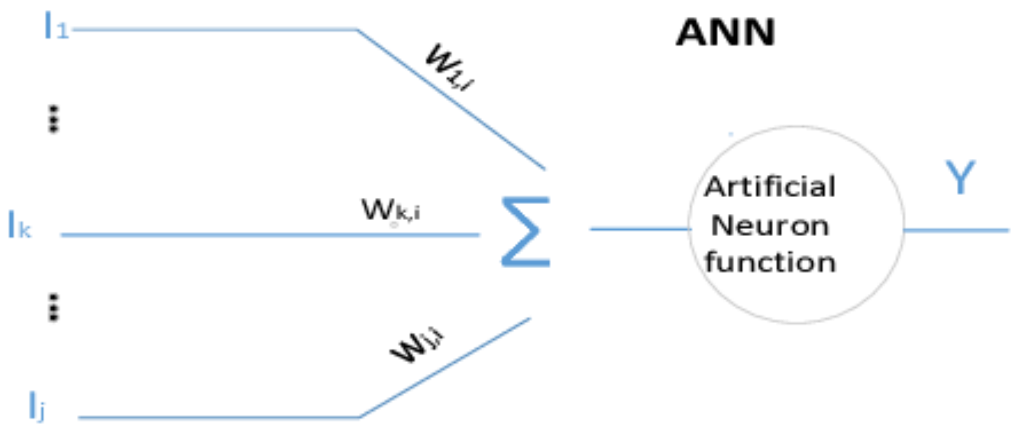
[Maqueda 18], [Zhu 18]
- Aggregate positive and negative events into separate channels
- Discards temporal information

[Zhu 18], [Rebecq, 19], [Zhu, 19]
- Represent events in space-time into a 3D voxel grid (x,y,t)
- Each voxel contains sum of ON and OFF events falling within the voxel
- Preserves temporal information but discards polarity information

[Gehrig,19]
- Represent events in space-time as a 4D Event Spike Tensor (x,y,t,p)
- Polarity information is preserved

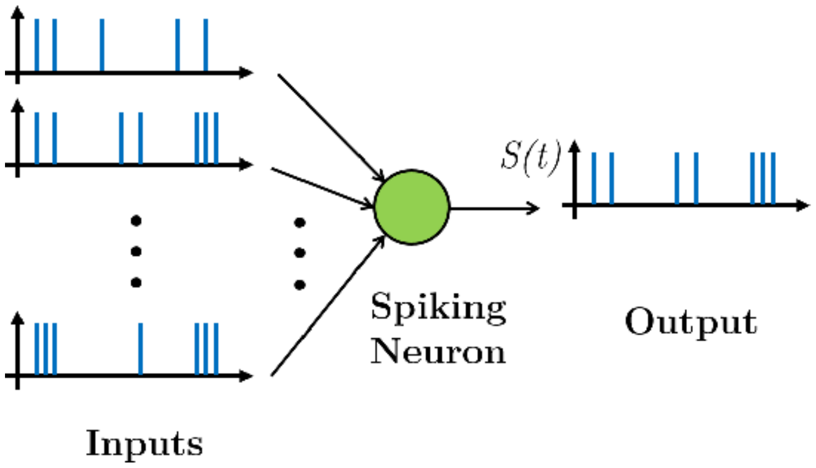
What neural networks architecture should we use?

- **Synchronous**, **Dense**, Artificial Neural Networks (ANNs), Deep Neural Networks (DNNs), etc



- **Asynchronous**, **Sparse** ANNs

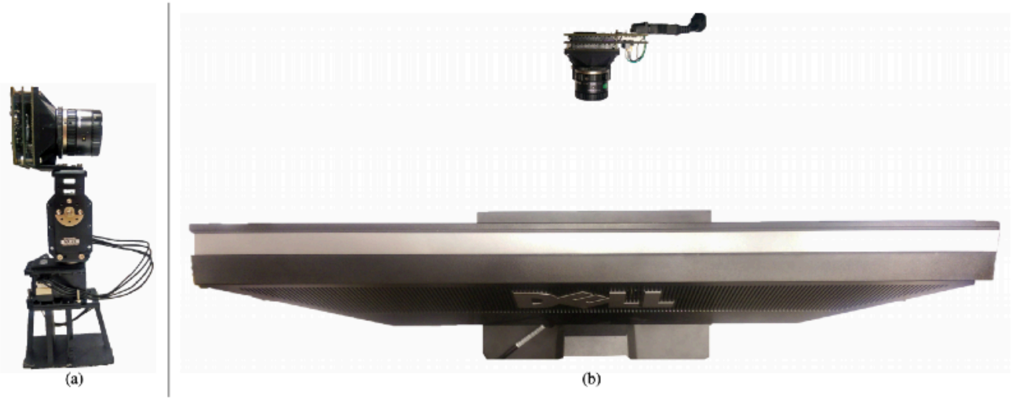
- **Asynchronous**, **Spiking** Neural Networks (SNNs)



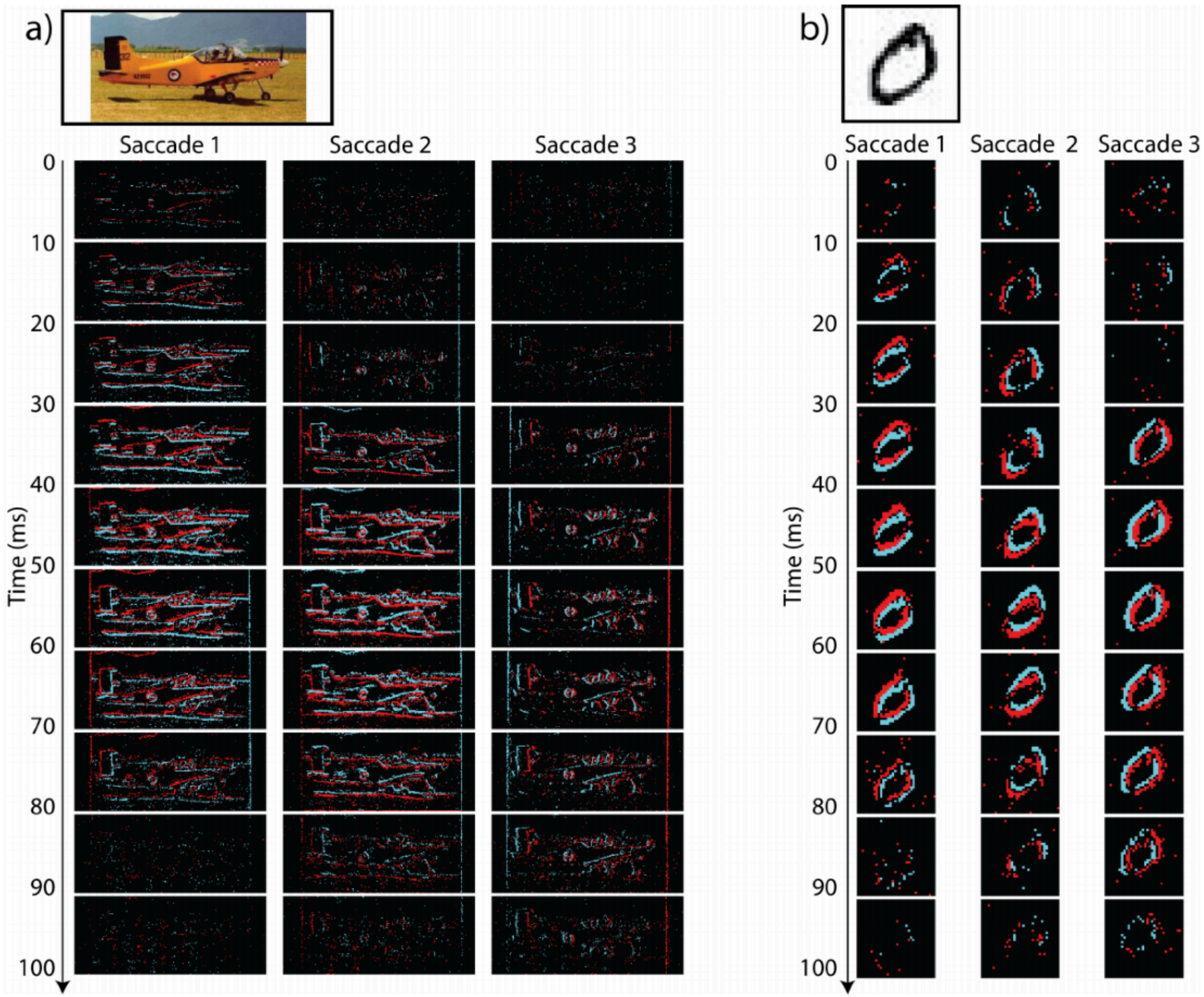
Some Datasets

- The Object Tracking, Action Recognition, and Object Recognition Database [Hu 2016].
- The Object Classification Database (CIFAR10-DVS) [Li 2017].
- End-to-end DAVIS driving dataset [Binas 2017].
- The Pose Estimation, Visual Odometry, and SLAM Database [Mueggler 2016].

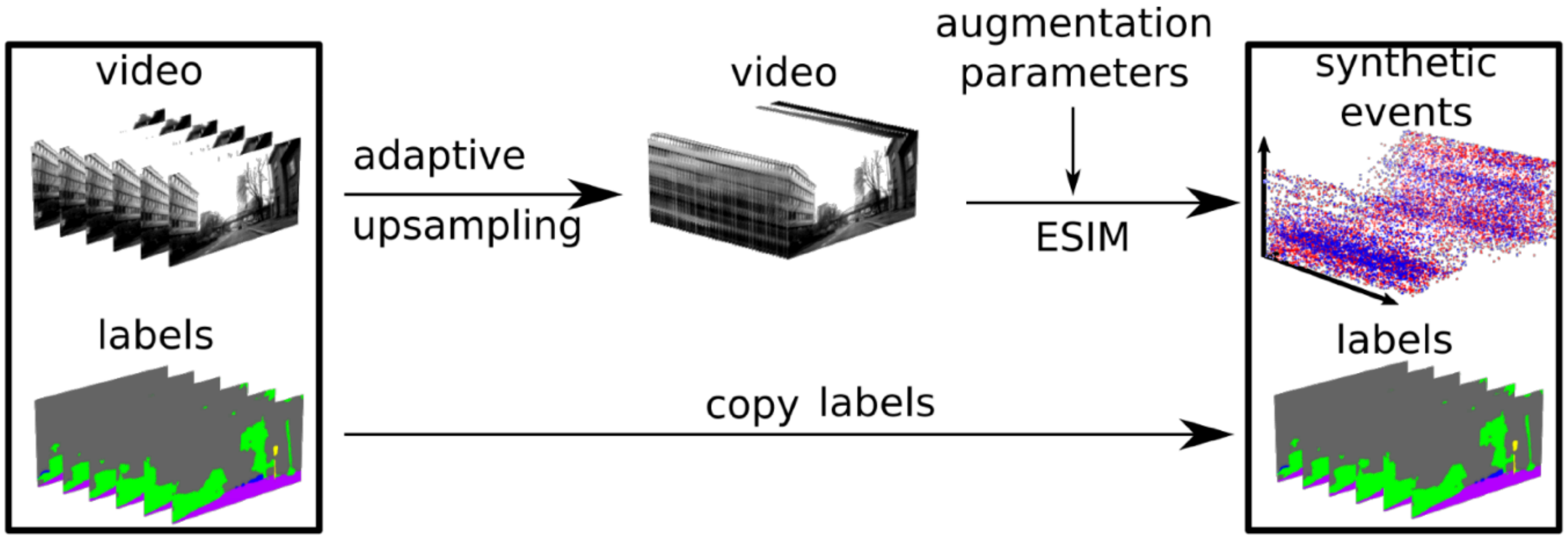
Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades

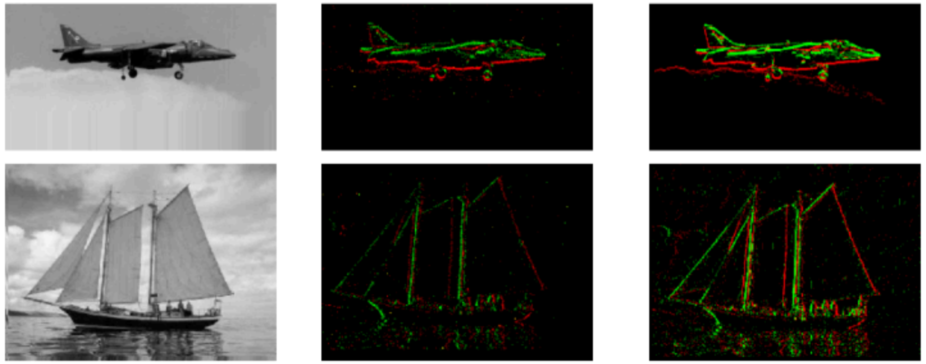


Camera is moved in from a monitor



[Orchard, 2015[

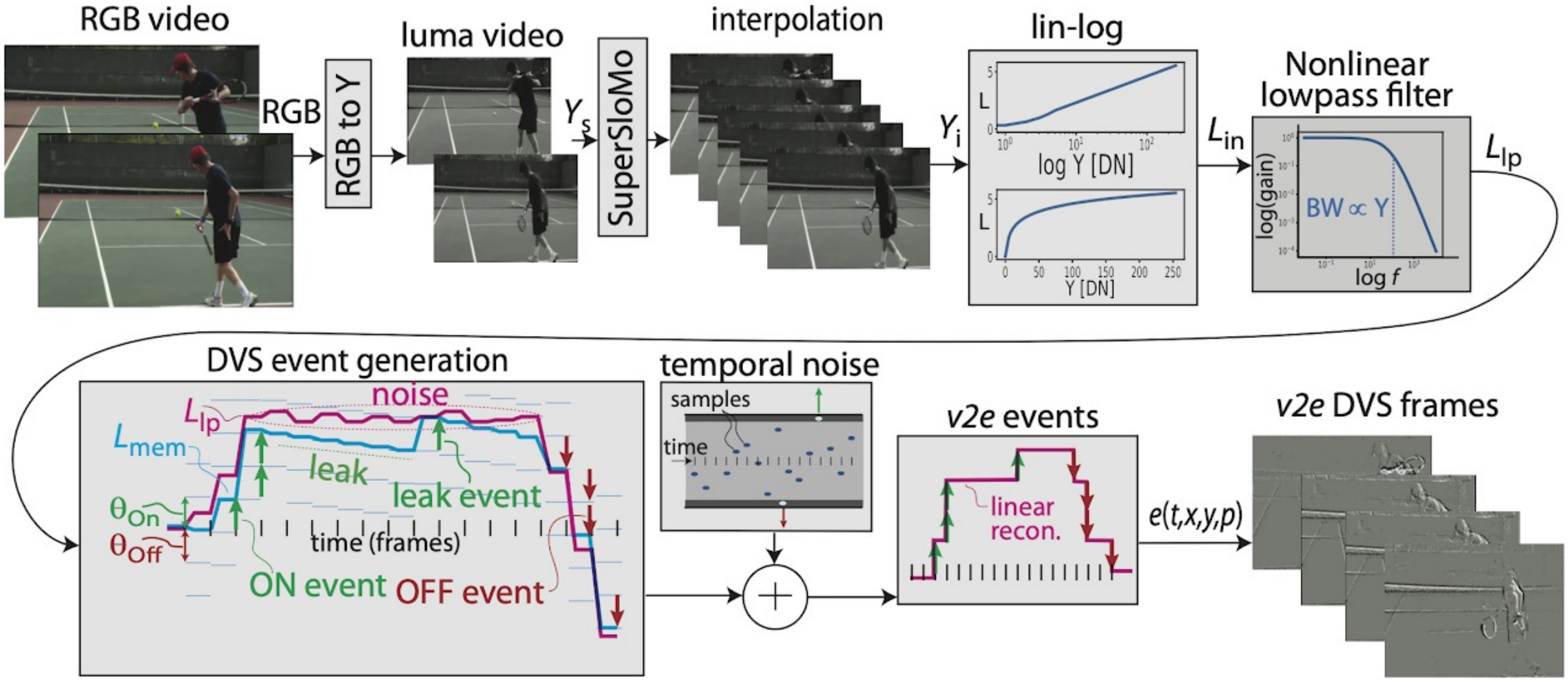




(a) preview (b) real events (c) synthetic events

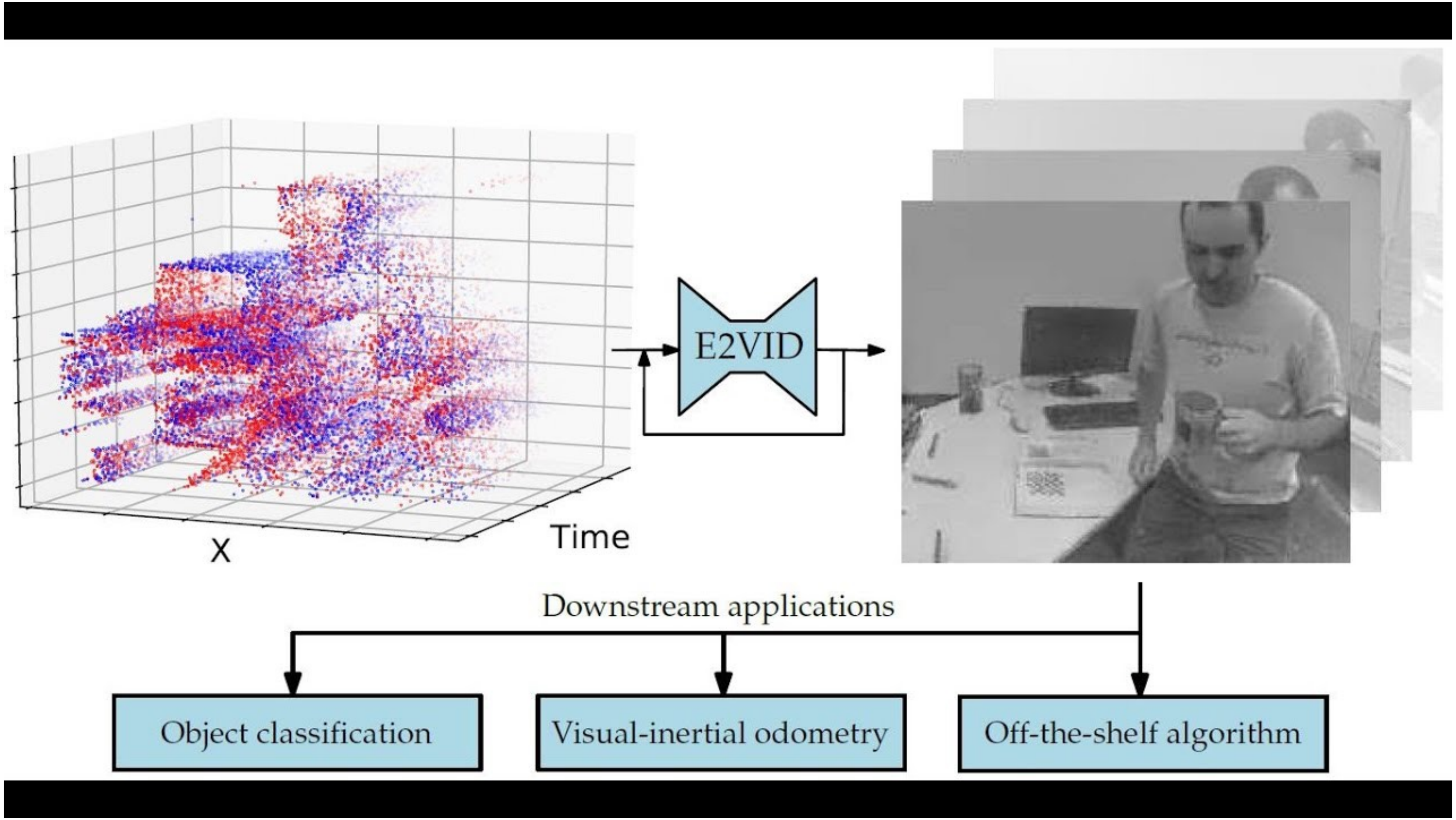
“Video to Events: Recycling Video Datasets for Event Cameras”, [Gehrig, CVPR20]

v2e: Event generation from video frames



[43]

Video to Events: Recycling Video Datasets for Event Cameras



[Gehrig, 2020]

https://www.youtube.com/watch?v=IdYrC4cUO0I

Events to videos:
A recurrent network is used to reconstruct videos from a stream of events

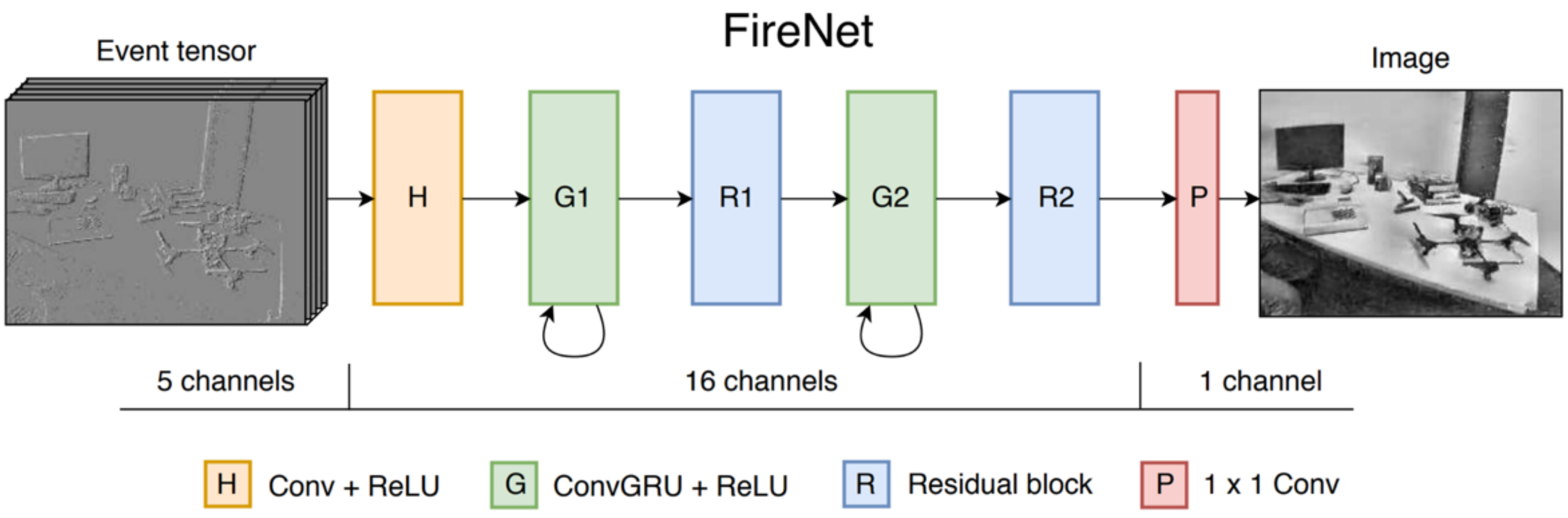


Image from [Rebecq 2019a]

https://youtu.be/eomALySSGVU

Example Applications

Face Analysis & Gesture recognition

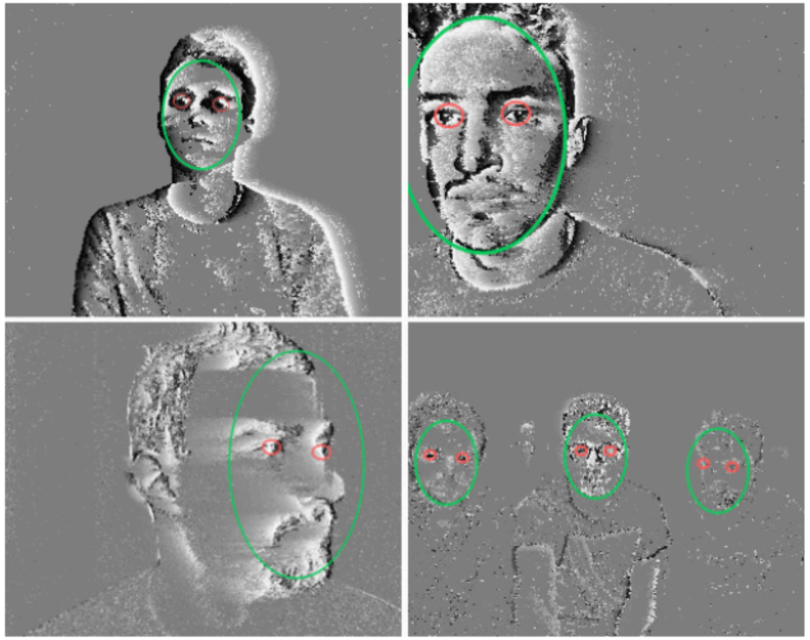


Figure 1. Event-based face tracking in different scenes. From left to right, top to bottom: **a**) indoors **b**) varying scale **c**) with one eye occluded **d**) multiple faces at the same time.

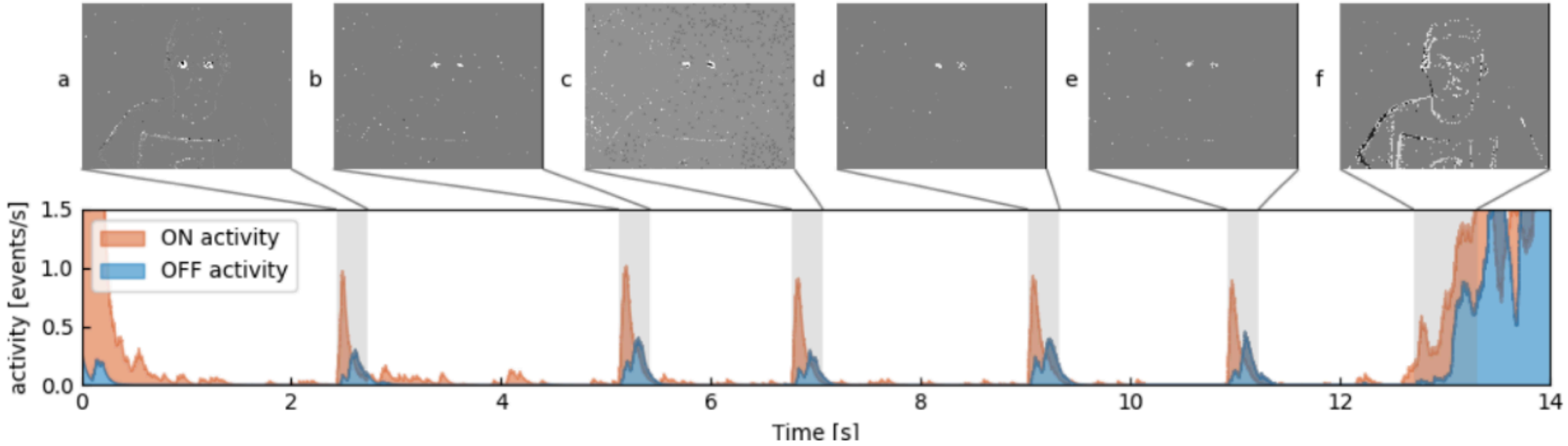


Figure 4. Showing ON (red) and OFF (blue) activity for one tile which lines up with one of the subject's eyes. Multiple snapshots of accumulated events for 250 ms are shown, which corresponds to the grey areas.**a-e)** Blinks. Subject is blinking. **f)** Subject moves as a whole and a relatively high number of events is generated.

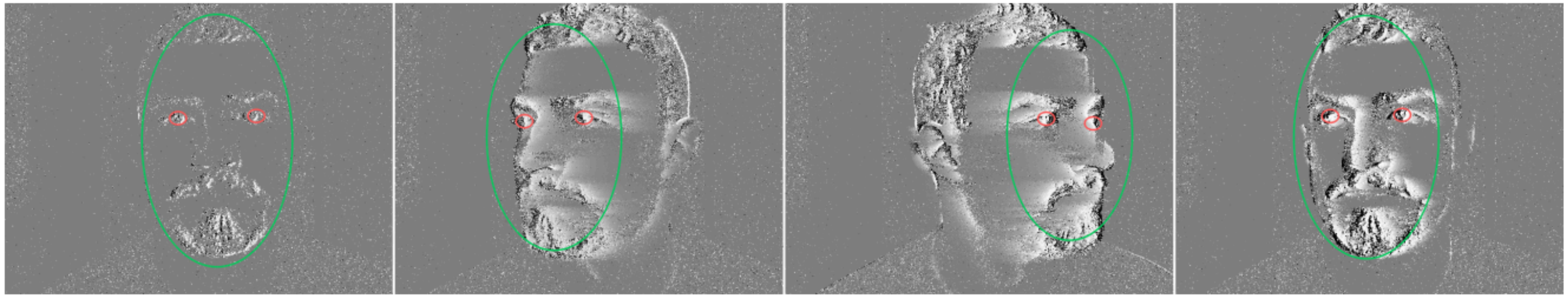


Figure 9. Pose variation experiment. **a**) Face tracker is initialised after blink. **b**) subject turns to the left. **c-d**) One eye is occluded, but tracker is able to recover.

Event-based Face Detection and Tracking in the Blink of an Eye, [Lenz 2019]

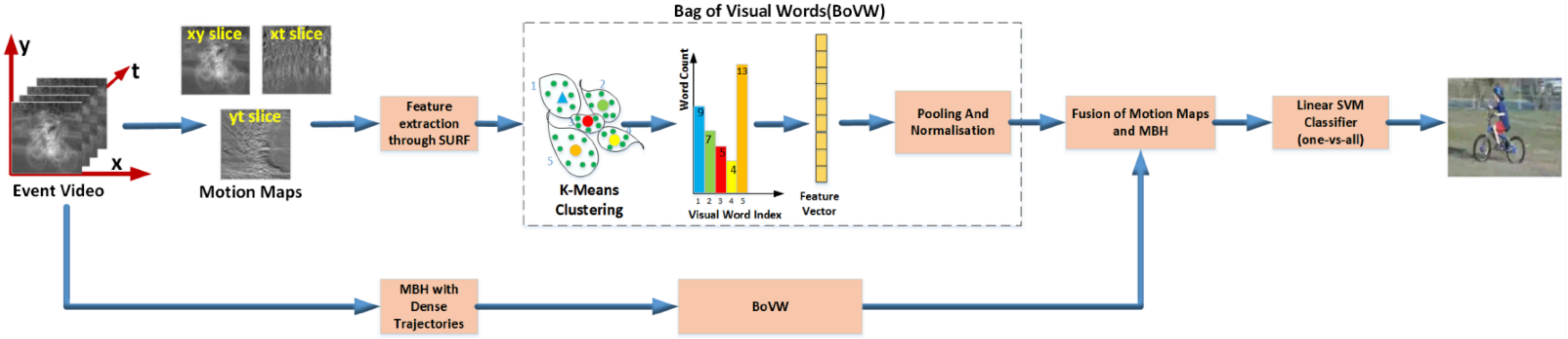


Figure 1: Our proposed method: The event stream from DVS is converted into video at 30 fps. Motion maps are generated through various projections of this event video and SURF features are extracted. MBH features using dense trajectory are also extracted. Bag of features encoding from both these descriptors are combined and given to linear SVM classifier (*one-vs-all*).

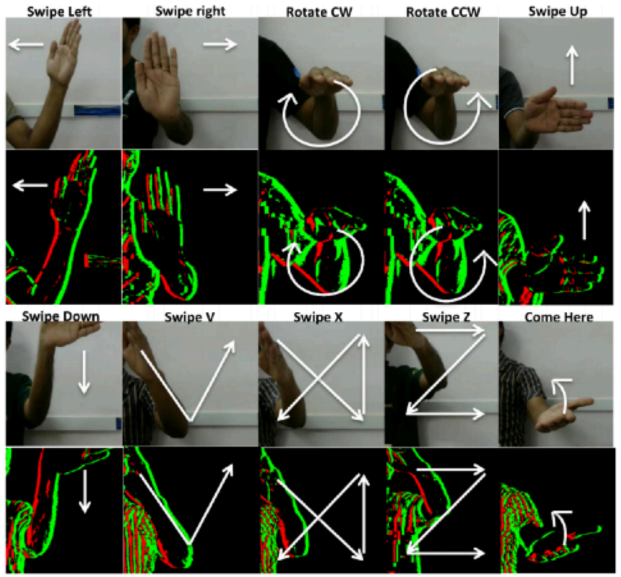


Figure 3: Gestures from the DVS dataset collected by us. Ground truth from an RGB camera is also shown.

Figure 2: YouTube Action Data Set [1]

"The DVS data was created by the authors by re-recording the existing benchmark UCF11 videos played on a monitor using a DAViS240C vision sensor."

Dynamic Vision Sensors for Human Activity Recognition, [Baby 2018]

DHP19: Dynamic Vision Sensor 3D Human Pose Dataset

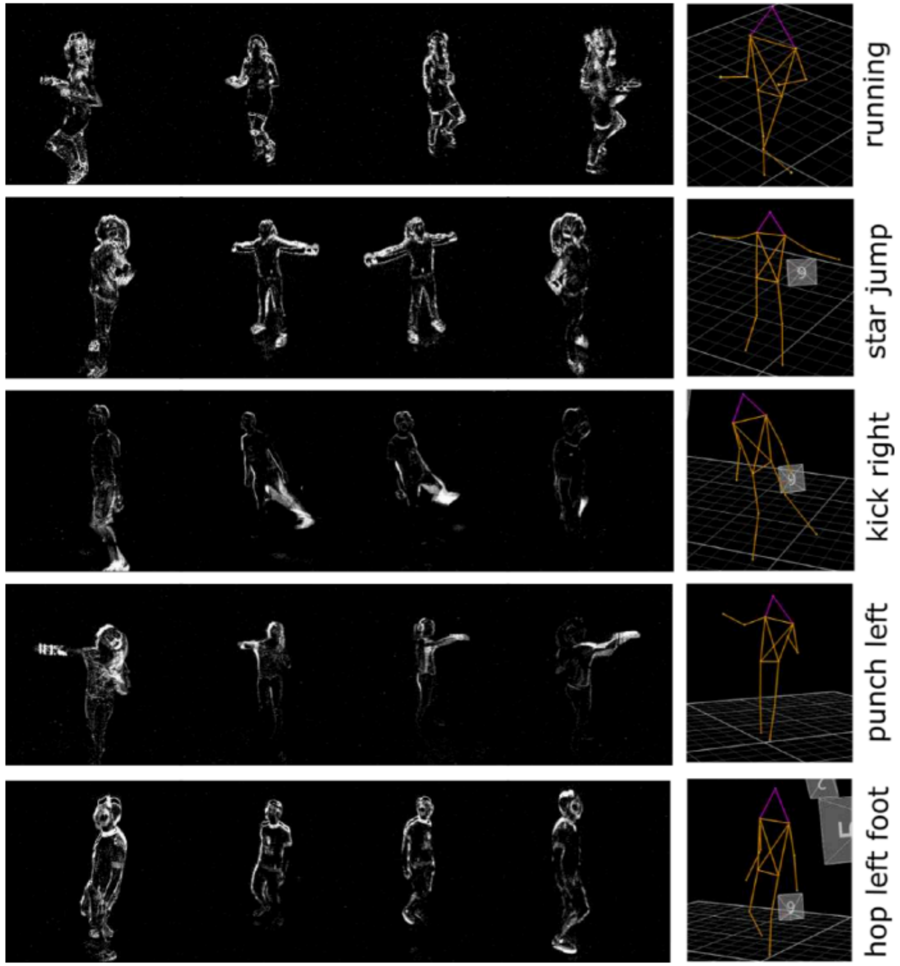


Figure 1. Examples from DHP19: DVS recordings (left) and Vicon labels (right) from 5 of the 33 movements. For visualization, the DVS events are here accumulated into frames (about 7.5 k events per single camera), following the procedure described in Sec. 4.

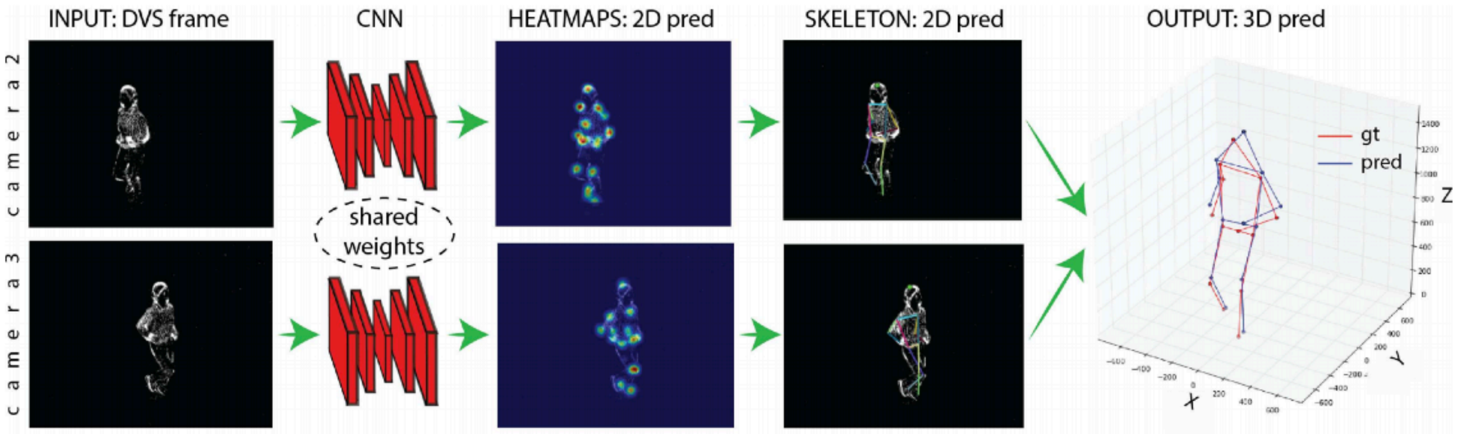


Figure 4. Overview of our proposed approach. Each camera view is processed by the CNN, joint positions are obtained by extracting maximum over the 2D predicted heatmaps, and 3D position is reconstructed by triangulation.

[Calabrese, 2019]

EventCap: Monocular 3D Capture of High-Speed Human Motions using an Event Camera

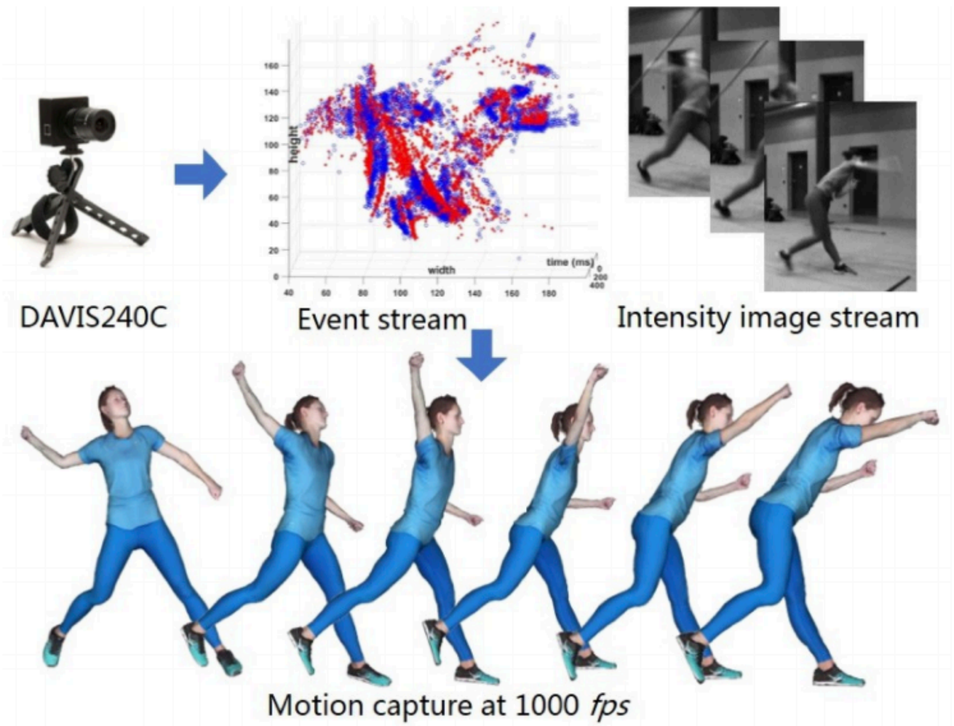


Figure 1: We present the first monocular event-based 3D human motion capture approach. Given the event stream and the low frame rate intensity image stream from a single event camera, our goal is to track the high-speed human motion at 1000 frames per second.

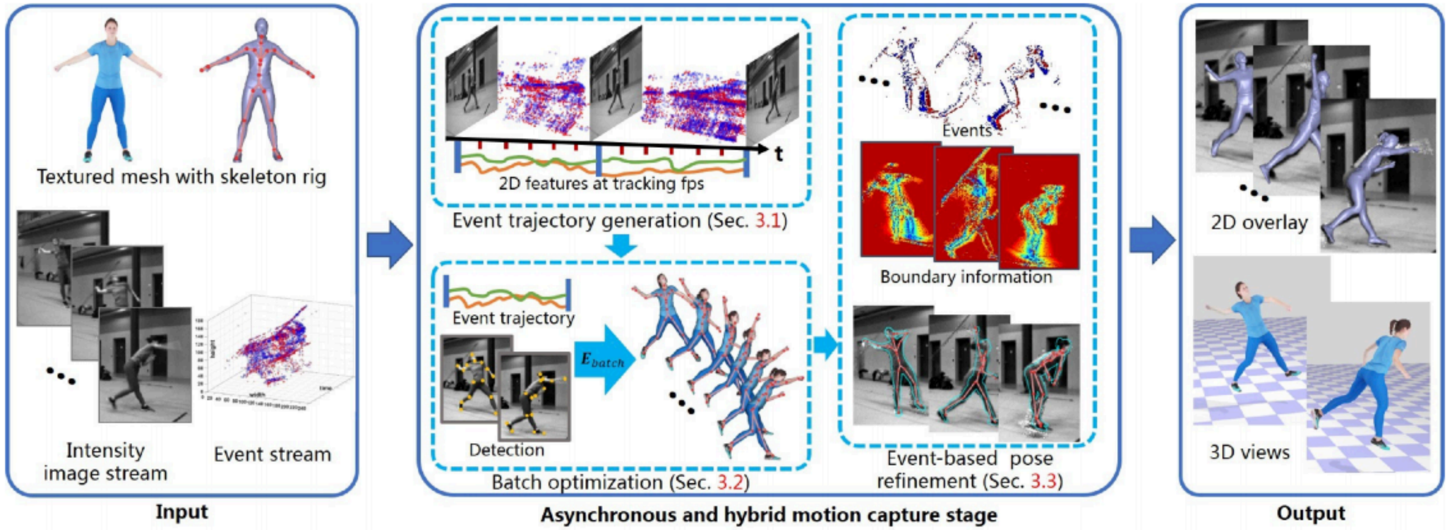
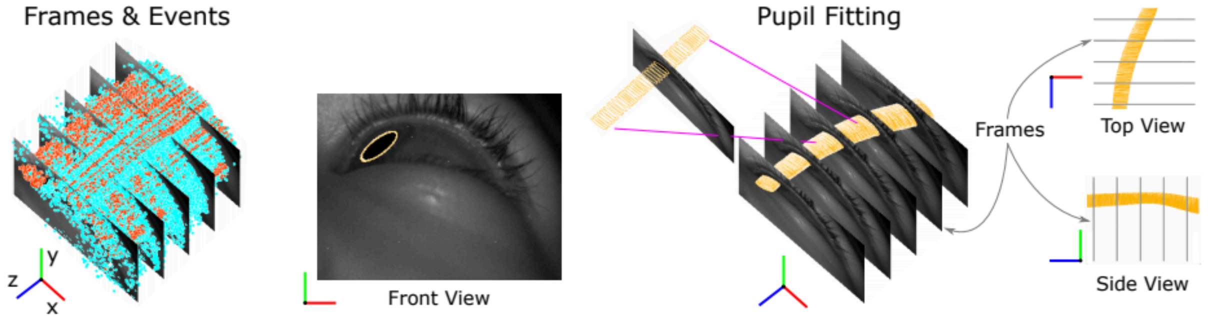
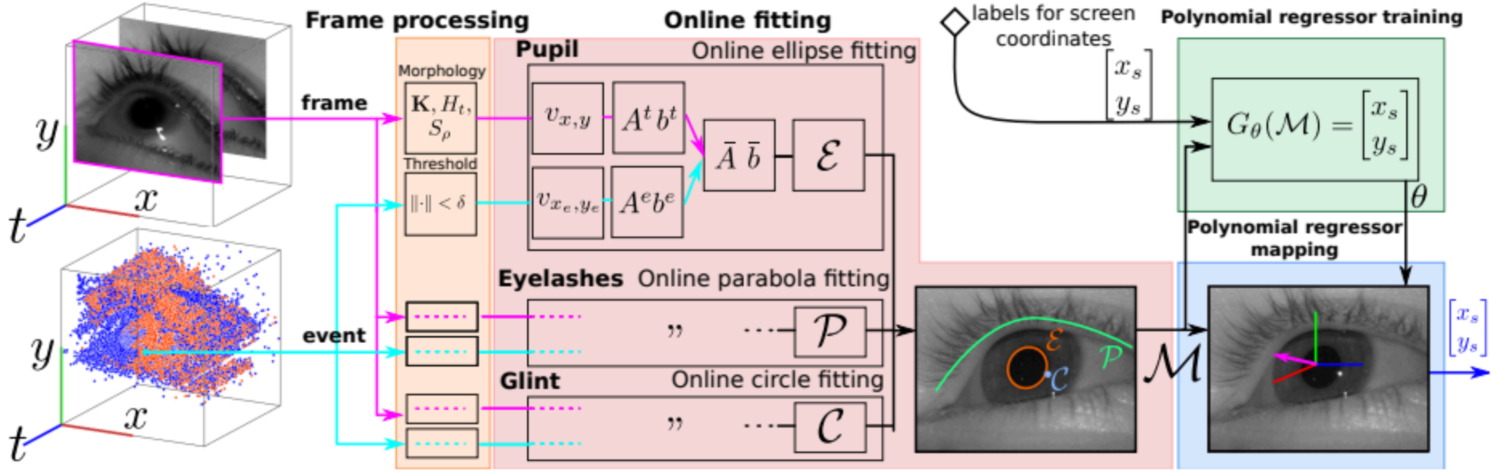


Figure 2: The pipeline of EventCap for accurate 3D human motion capture at a high frame rate. Assuming the hybrid input from a single event camera and a personalized actor rig, we first generate asynchronous event trajectories (Sec. 3.1). Then, the temporally coherent per-batch motion is recovered based on both the event trajectories and human pose detections (Sec. 3.2). Finally, we perform event-based pose refinement (Sec. 3.3).

[Xu, 2019]

Event Based, Near-Eye Gaze Tracking Beyond 10,000Hz





[Angelopoulos 2020]

Event-based cameras for face expression & gesture recognition

Rodrigo Verschae

Institute of Engineering Sciences

Universidad de O'Higgins

rodrigo@verschae.org

Universidad de O'Higgins

June 6, 2024


Ignacio Bugueno

Gesture recognition

Rodrigo Verschae, Ignacio Bugueno

”Event-based Gesture and Facial Expression Recognition: A Comparative Analysis”, IEEE Access, 2023

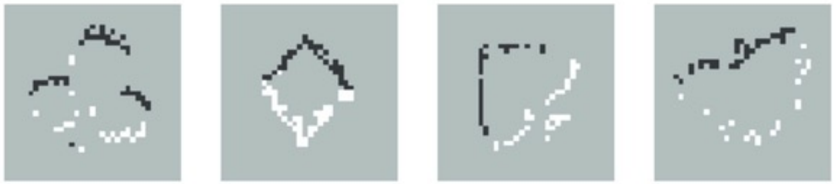
Context: object classification has been already studied, but…

» Non-deformable object data-bases (since 2013)

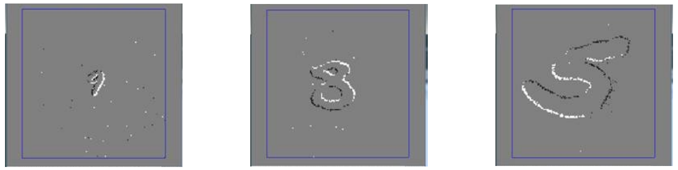
- Poker-DVS: set of 131 poker symbols extracted from DVS recordings.
- MNIST-DVS: set of 30,000 DVS recordings of different MNIST handwritten digit images.
- N-MNIST: spiking version of MNIST dataset, recorded with an ATIS sensor
- N-Caltech101: spiking version of the Caltech101 dataset, recorded with an ATIS sensor.

All of these are non-deformable objects.

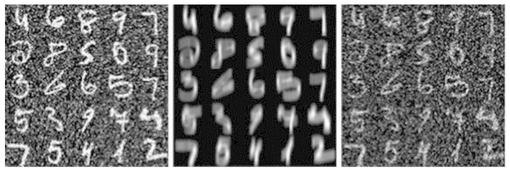
In these problems the temporal information is not that important



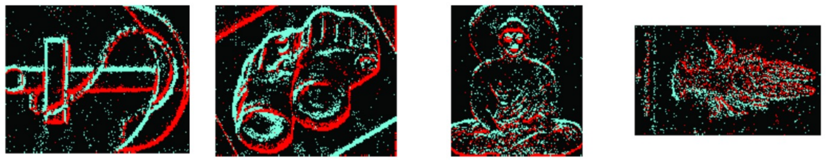
Poker-DVS [27]



MNIST-DVS [28]



N-MNIST [29]



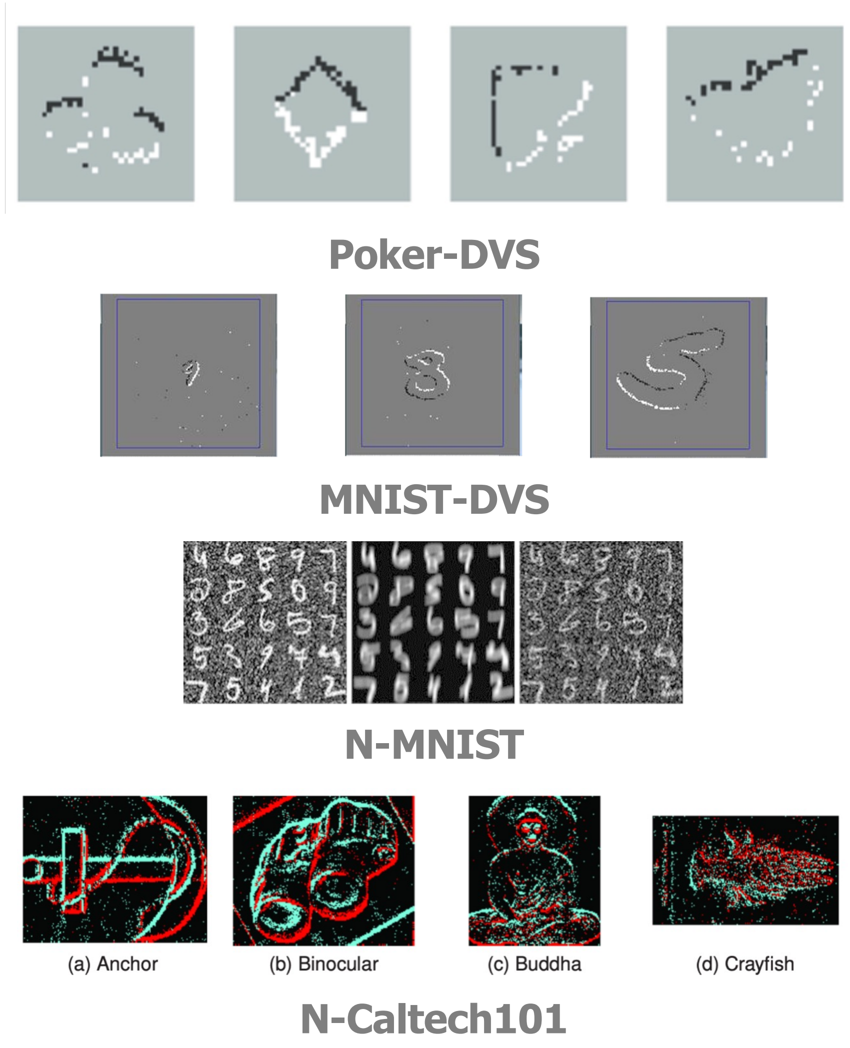
(a) Anchor (b) Binocular (c) Buddha (d) Crayfish

N-Caltech101 [30]

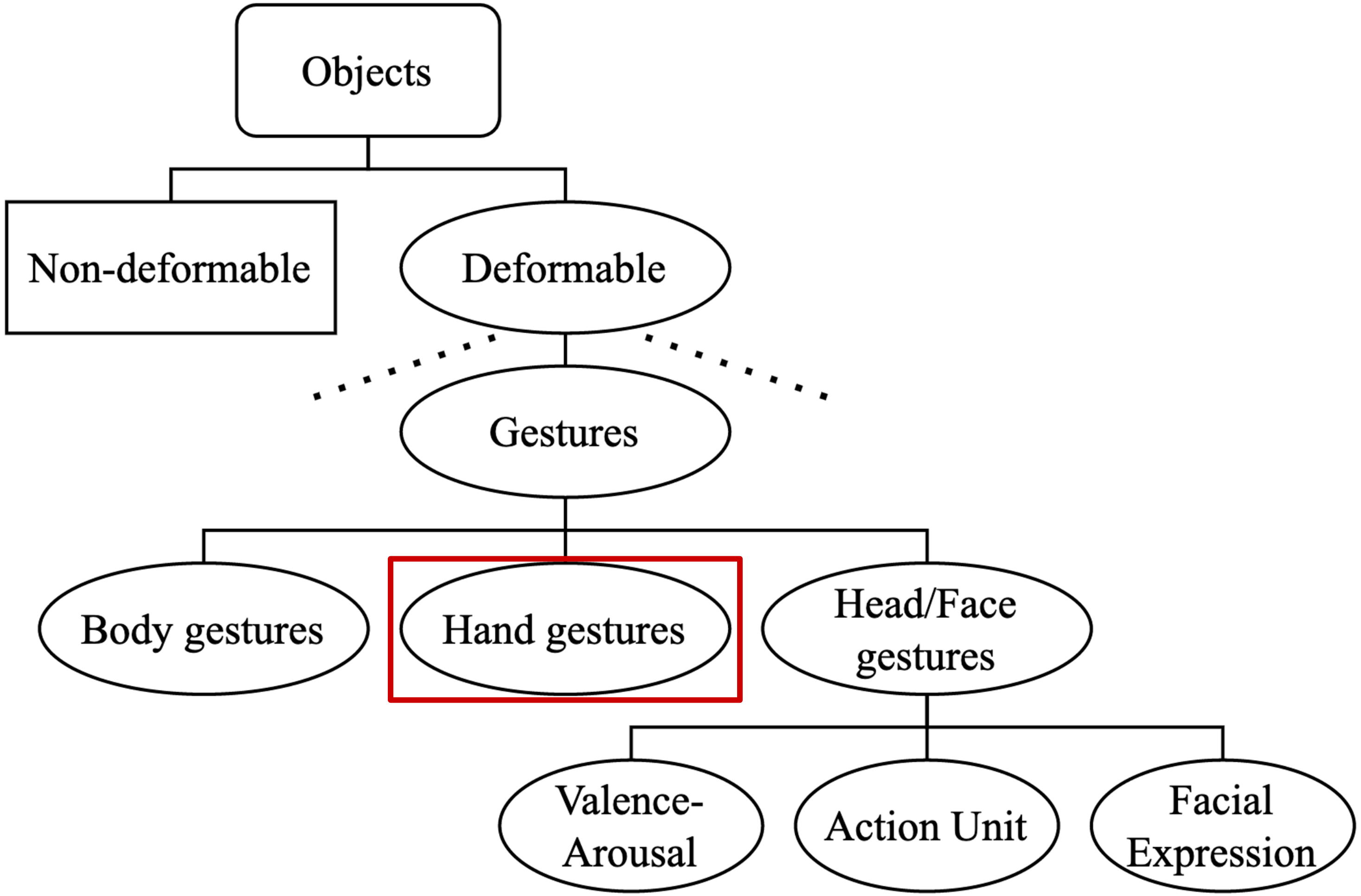
Application of interest: object classification

- Object classification is a problem addressed and studied in event cameras.

- Since 2013, multiple bases with object focus have been elaborated, highlighting:
  - **Poker-DVS:** *conjunto de 131 símbolos de póquer extraídos de grabaciones DVS*
  - **MNIST-DVS:** *conjunto de 30.000 grabaciones DVS de diferentes imágenes de dígitos manuscritos*

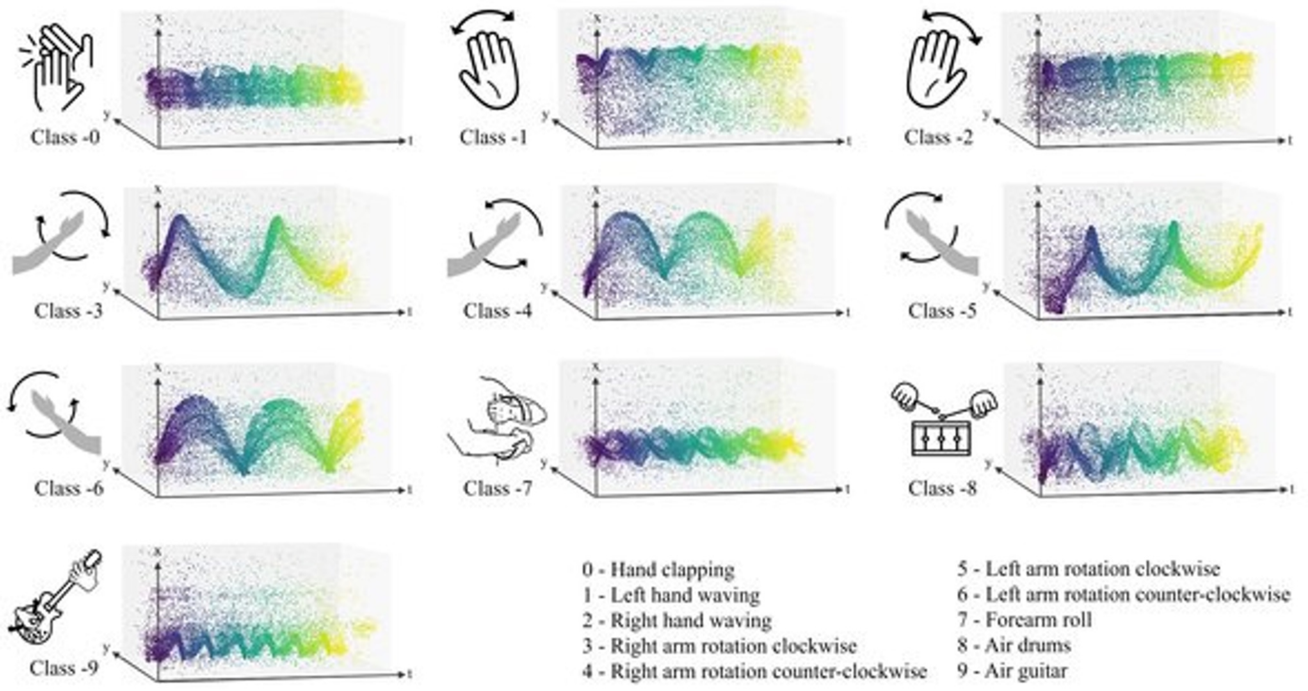


Object taxonomy: hand gestures



Bases de datos de gestos existentes

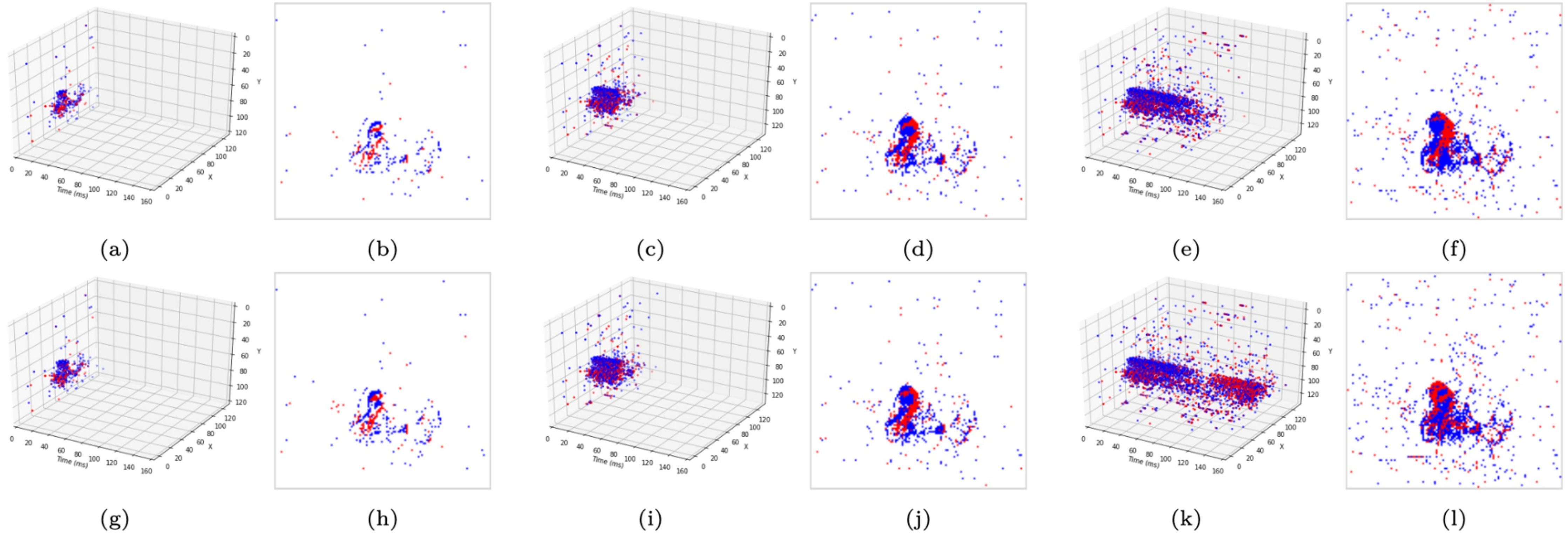
IBM - DVS128 Gesture Dataset



Model	**10 cl.**	**11 cl.**
Time-surfaces	96.59	90.62
SNN eRBP	N/A	92.70
Slayer	N/A	93.64
CNN	96.49	94.59
Space-time clouds	97.08	95.32
DECOLLE	N/A	95.54
TORE	N/A	96.2
EvT	98.46	96.20
RG-CNN	N/A	97.2
AlexNet - LSTM	97.5	97.53
Inception3D + Voting	99.58	**99.62**

¿Se podrá mejorar?

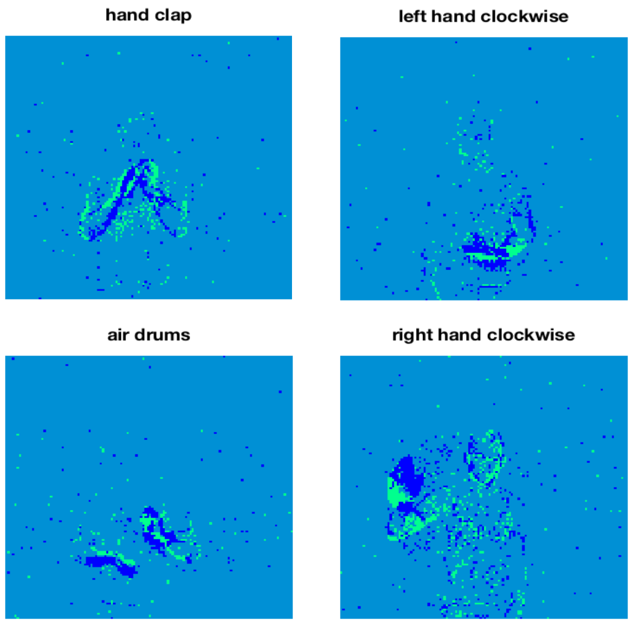
Bases de datos de gestos existentes
IBM - DVS128 Gesture Dataset



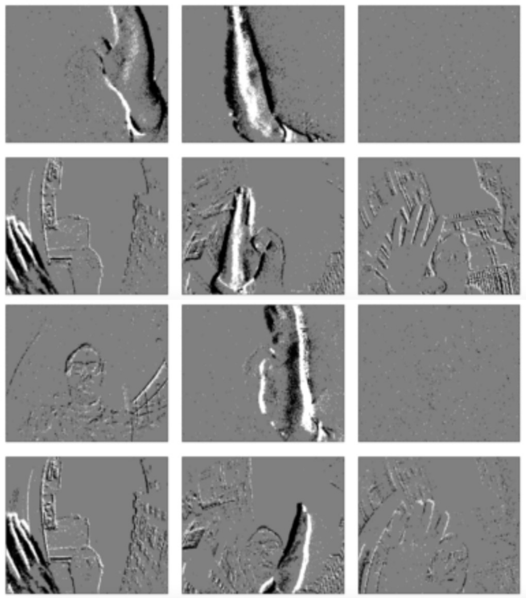
IBM DVS128 Gesture Dataset - 3D Temporal evolution and 2D Events representation with 128x128 spatial resolution. Time window analysis: (a,b) 10ms; (c,d) 33ms; (e,f) 100ms. Event window analysis: (g,h) 500 events; (i,j) 1500 events; (k,l) 4500 events.

Gesture databases

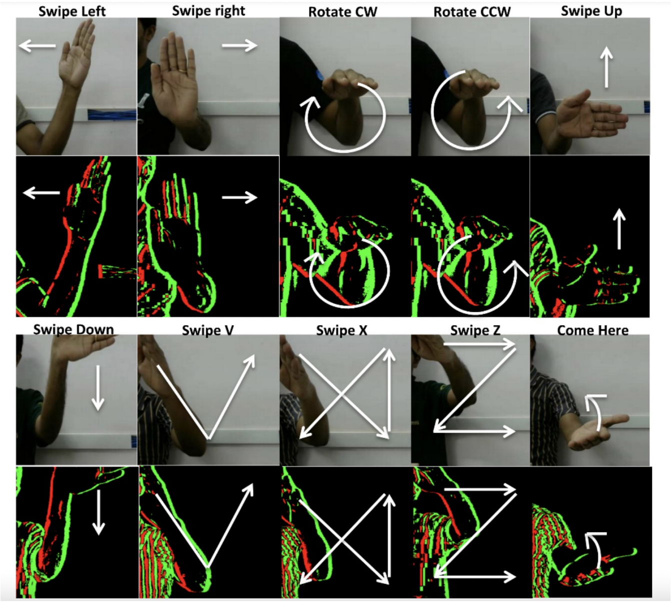
IBM - DVS128 Gesture Dataset **NavGesture Dataset** **IITM DVS128 Gesture Dataset**



[21]



[31]



[32]

Evaluation of state-of the art

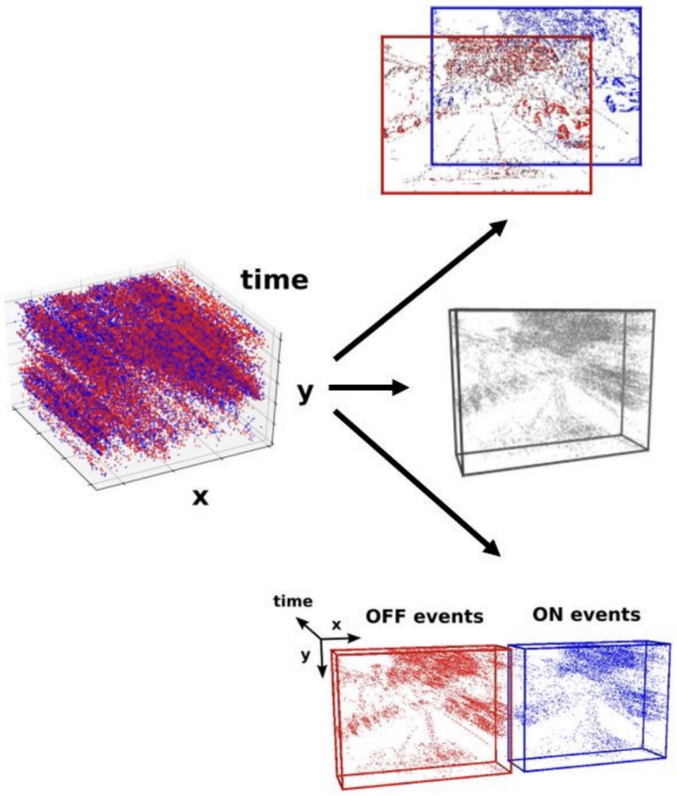
1. End-to-End Learning of Representations for Asynchronous Event-Based Data **(EST)**
2. Event-based Asynchronous Sparse Convolutional Networks **(Asynet)**
    - Standard **(Asynet I)**
    - Sparse (**Asynet II)**

On the datasets

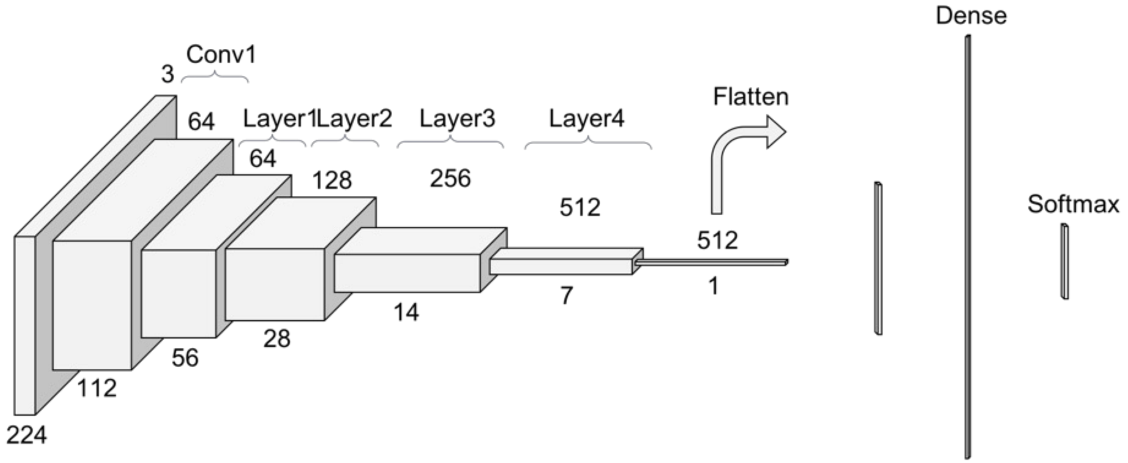
» **IBM - DVS128 Gesture Dataset**
» *NavGesture*
» *IITM DVS128 Gesture Dataset*

under a sensitivity analysis, varying the size of the time window and the event window

1. End-to-End Learning of Representations for Asynchronous Event-Based Data



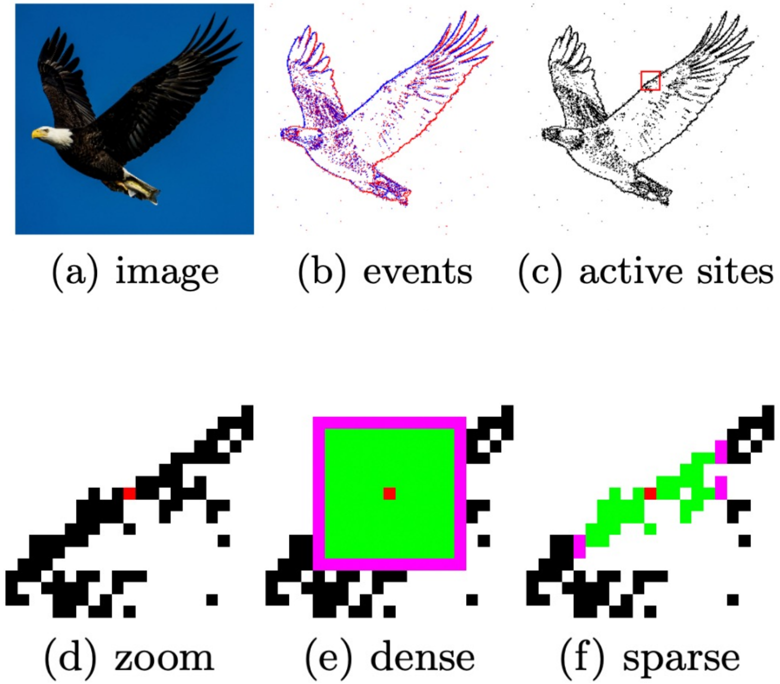
Trilinear Kernel

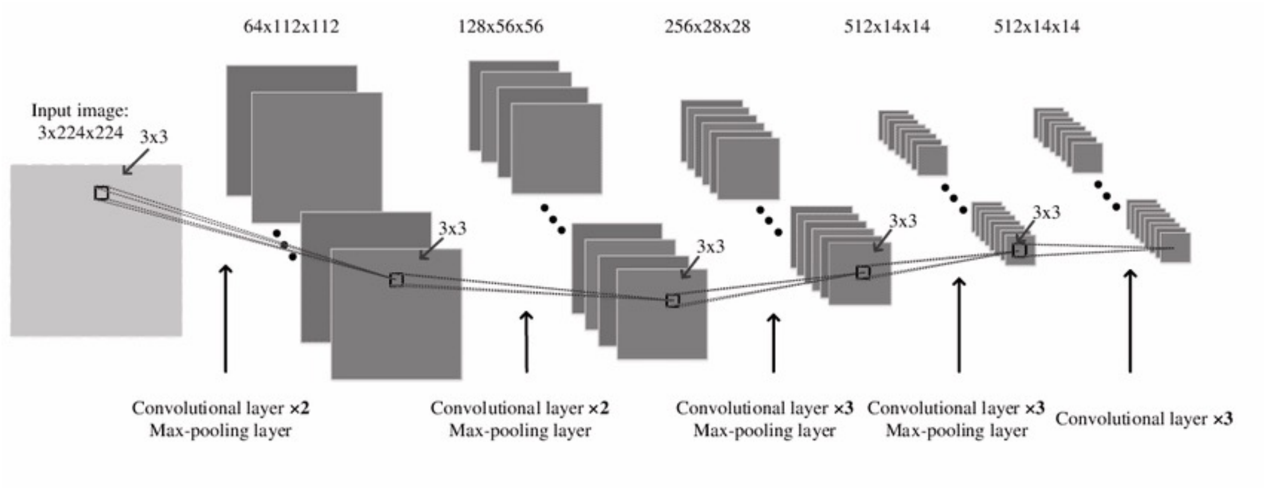


Resnet-34

[Gehrig et at, 2019]

2. Event-based Asynchronous Sparse Convolutional Networks

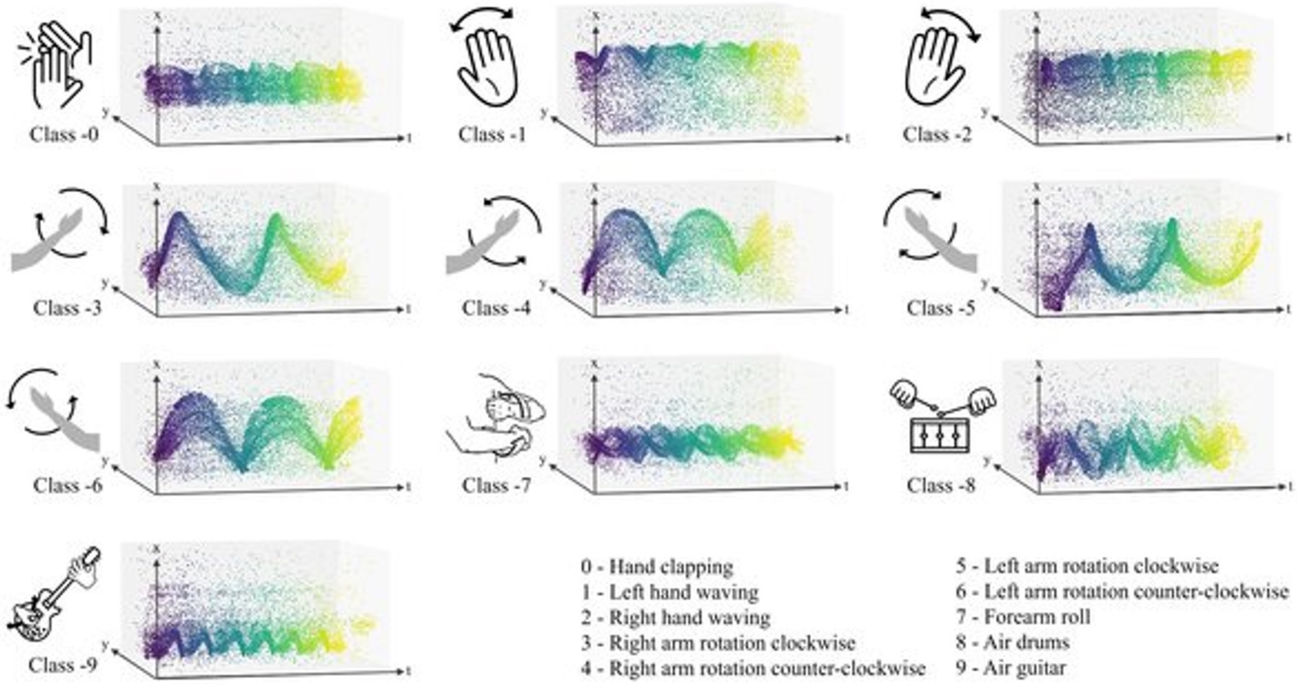


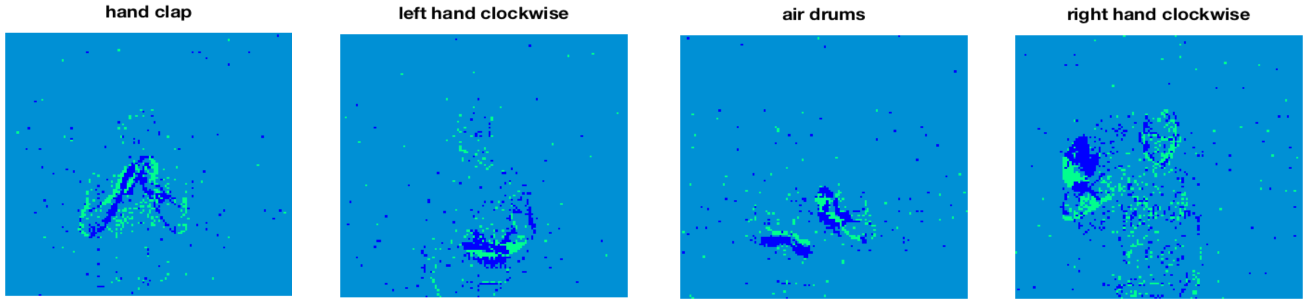


VGG-13

Messikommer et at, 2020

IBM - DVS128 Gesture Dataset





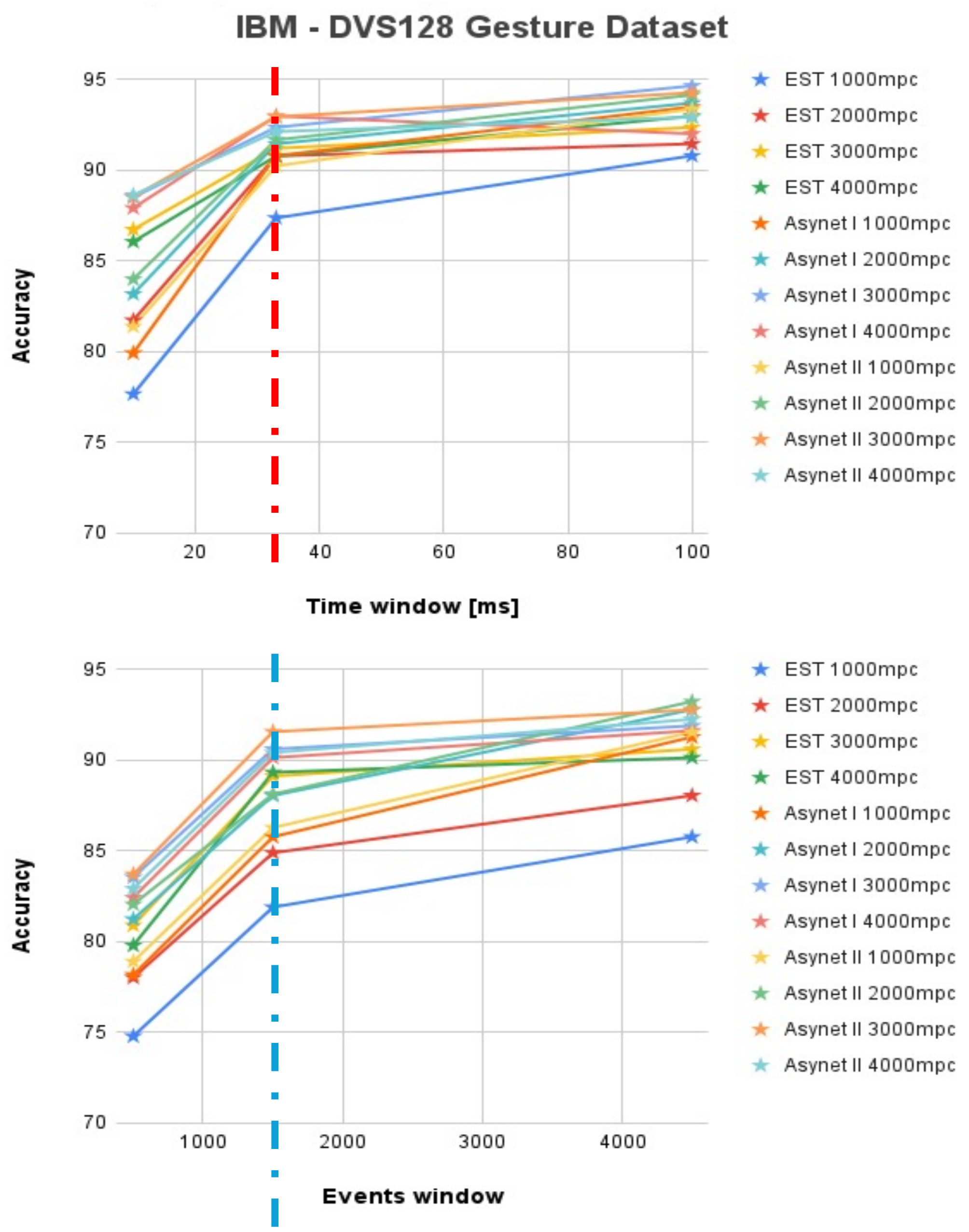
Model	10 cl.	11 cl.
Time-surfaces	96.59	90.62
SNN eRBP	N/A	92.70
Slayer	N/A	93.64
CNN	96.49	94.59
Space-time clouds	97.08	95.32
DECOLLE	N/A	95.54
TORE	N/A	96.2
EvT	98.46	96.20
RG-CNN	N/A	97.2
AlexNet - LSTM	97.5	97.53
Inception3D + Voting	99.58	**99.62**

IBM - DVS128 Gesture Dataset

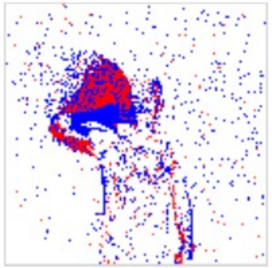
(assuming 11 classes) 72 experiments are performed

	Time approach	Event approach
EST	Accuracy: 93,82% Time window: 100ms Method: 3000mpc	Accuracy: 91,15% Event window: 4500 Method: 2000mpc
Asynet I	**Accuracy: 94,66% Time window: 100ms Method: 3000mpc**	Accuracy: 92,79% Event window: 4500 Method: 2000mpc
Asynet II	Accuracy: 94,27% Time window: 100ms Method: 3000mpc	**Accuracy: 93,24%** Event window: 4500 Method: 2000mpc

**mpc: *samples per class*



Event-based Gesture and Facial Expression Recognition Pipeline



$$\{(x_k, y_k, t_k, p_k)\}_{k=1}^{N}$$

Evaluated methods

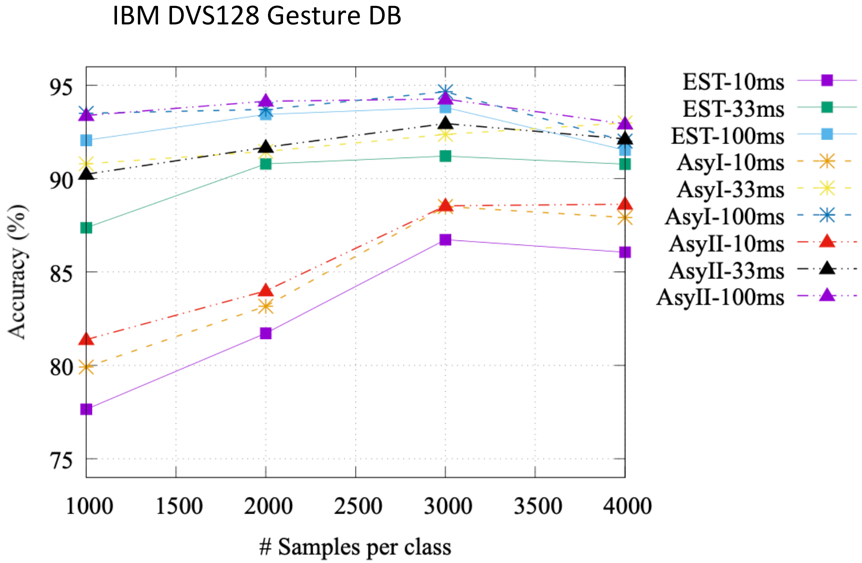
- EST
- Asynet
- ESTM

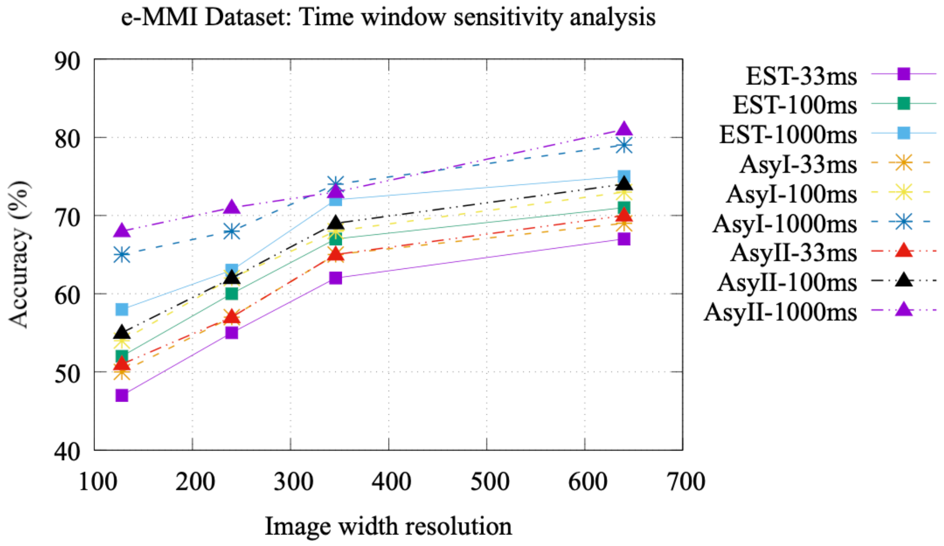
Datasets

- IBM DVS128
- NavGesture
- e-MMI
- E-CK+

Analysis includes

- Temporal length of the sample
- Number of events per sample
- Sensor Resolution
- Use of LSTM





	NavGesture - Walk					
	Time window			Event window		
	10ms	33ms	100ms	500e	1500e	4500e
EST	88.3	**88.5**	87.2	75.2	87.3	**88.7**
AsyI	89.5	**90.6**	88.1	78.6	88.4	**90.9**
AsyII	89.7	**91.3**	90.8	84.5	88.9	**91.2**

Method	Reference	DGX-1 t_{exec}
EST	[5]	2.1 [ms]
Asynet	[6]	23.4 [ms]
ESTM	Our implementation	31.9 [ms]

Event-based Gesture and Facial Expression Recognition: A Comparative Analysis, IEEE Access 2023

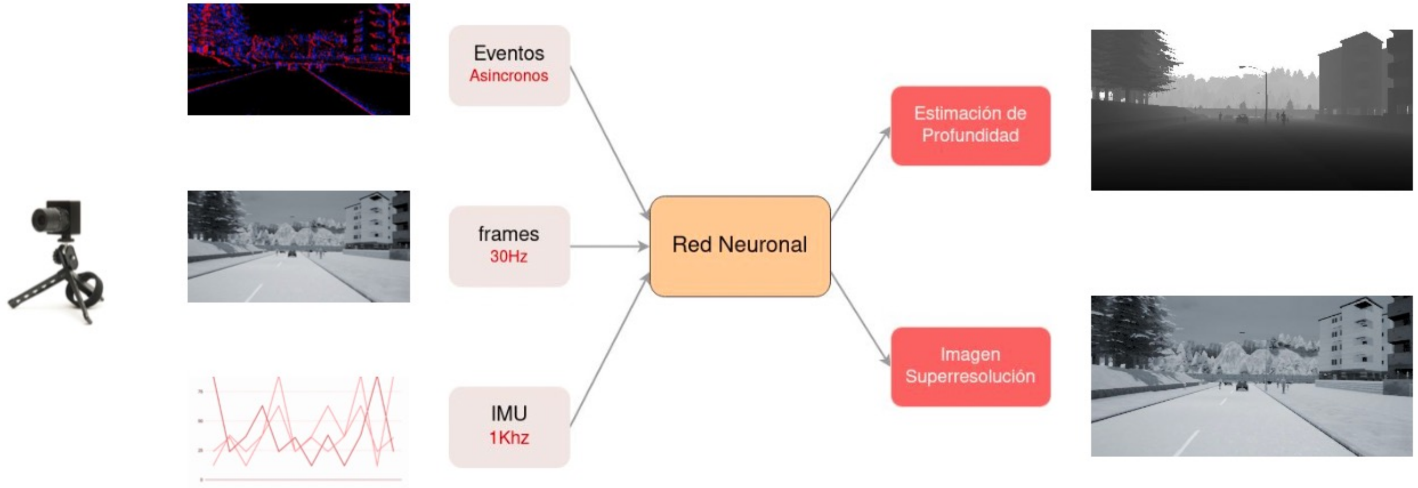
Jose Astorga

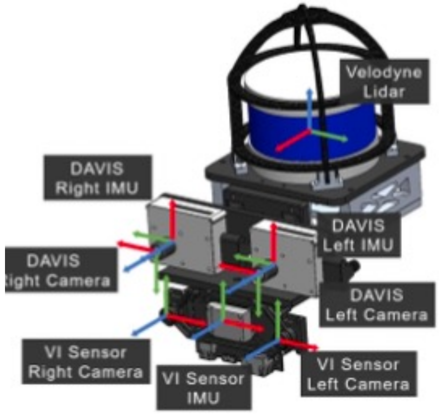
Depth and odometry estimation using Event-based cameras, IMU and FRAME information

Jose Astorga, Rodrigo Verschae

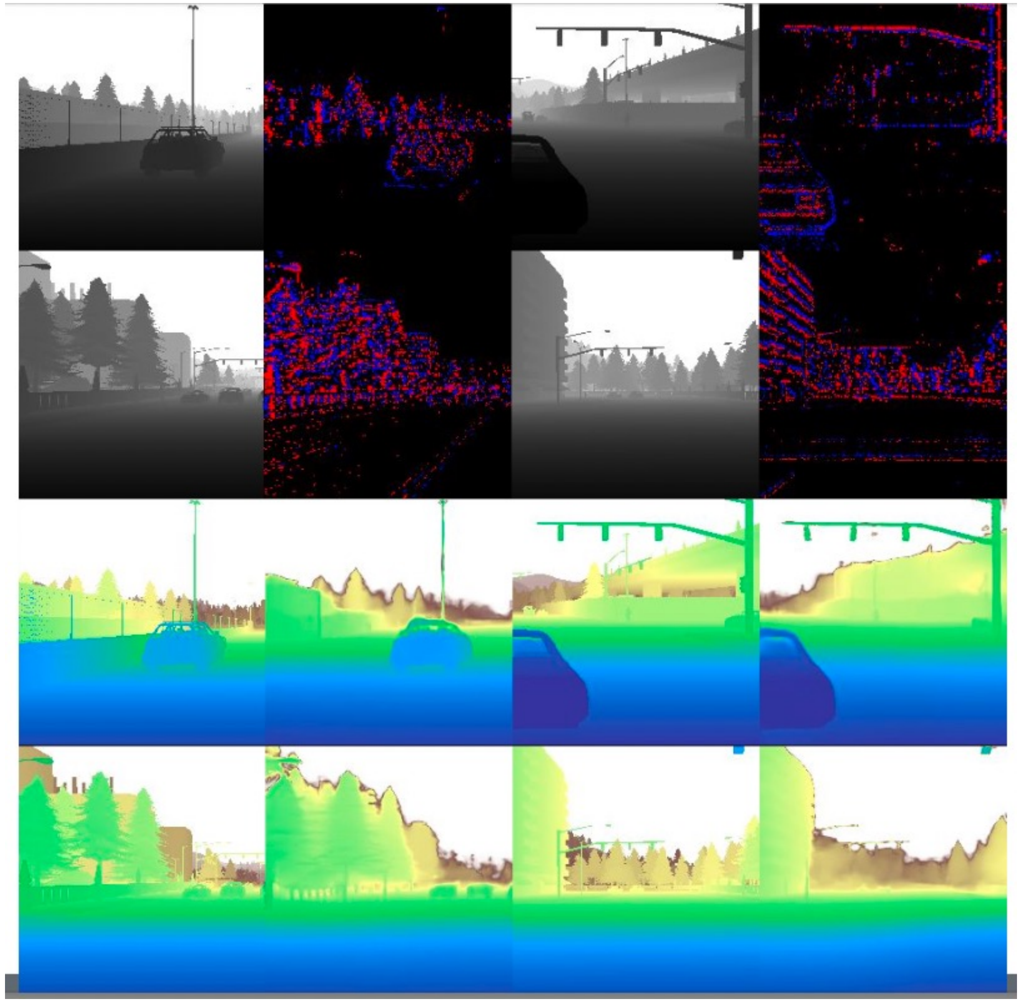
Depth estimation using Event-based cameras using IMU and FRAME information

Jose Astorga



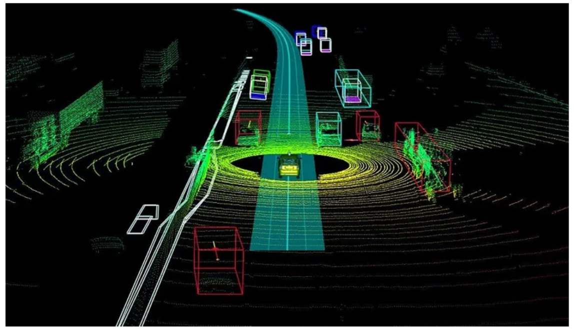


Preliminary Results

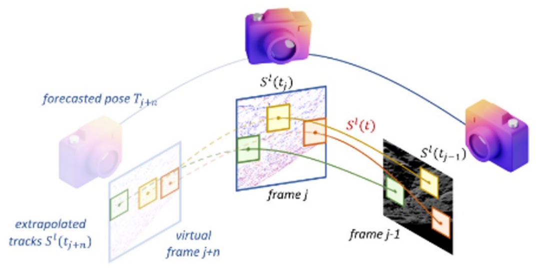


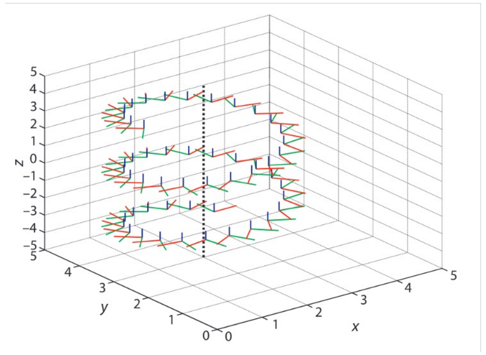
Depth estimation and odometry are important for robotics and autonomous vehicles

Navigation in Autonomous vehicles



Depth Estimation + Odometry





Motivation: Depth Estimation

Stereo Cameras

- Dense depth map.
- Problems in low areas in features.
- Narrow range.

Depth Cameras

- Dense depth map.
- Only works indoors.
- Narrow range.

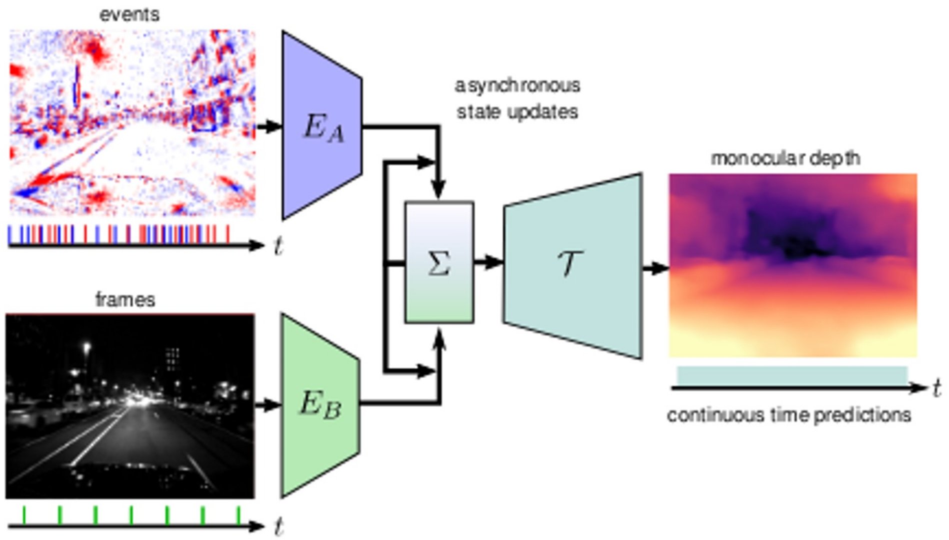
LiDAR

- High range and accuracy, large FOV.
- Very expensive.
- High structure information but low semantic information.

Related Work: Monocular Depth Estimation in Event Cameras

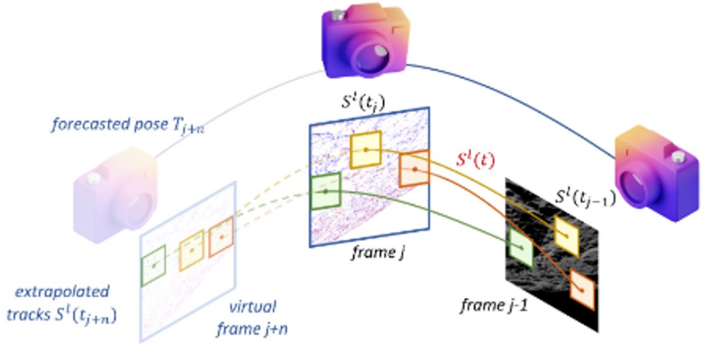
» The good results obtained by deep learning methods in traditional cameras have motivated its use in event cameras, in stereo and monocular configurations.

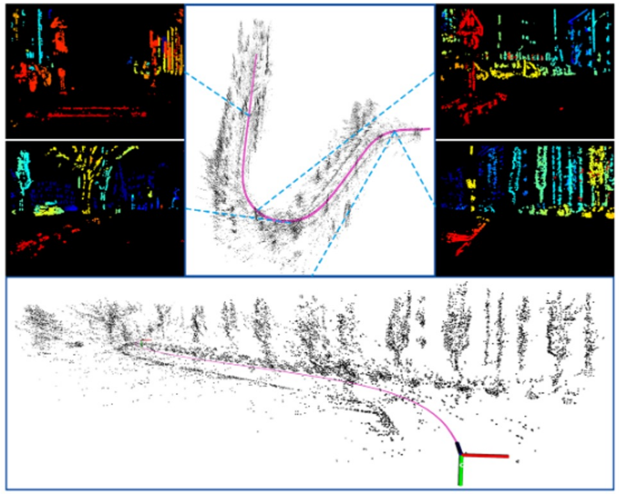
- Recurrent networks [1, 29, 30].
- Vision Transformers networks [29, 36, 37].
- In fusion with images [1, 32, 34, 35, 36].
- Unsupervised learning [2, 5].



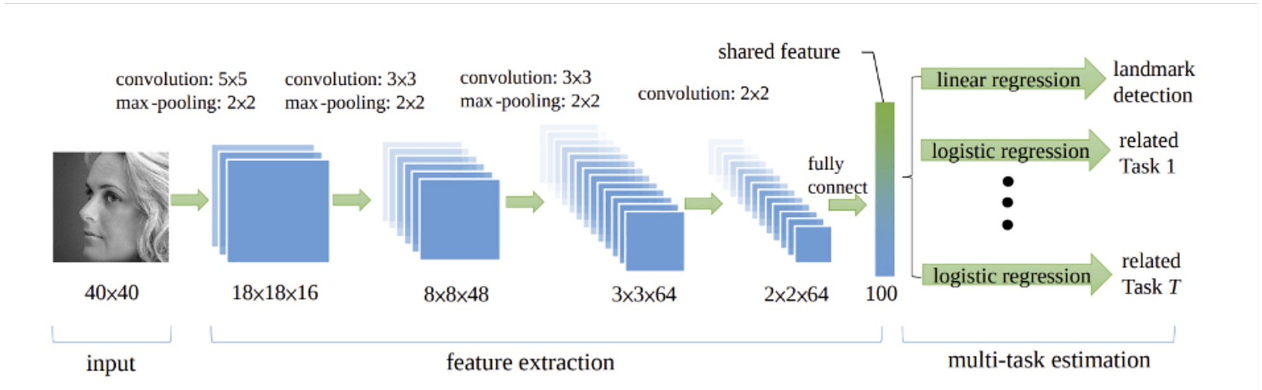
Related Work: Visual and Inertial Odometry in Event Cameras.

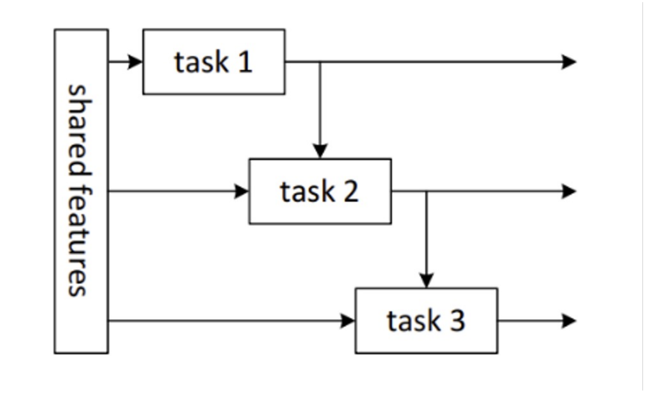
- It is a widely explored task in event cameras
- Usually encompassed with traditional feature tracking or direct methods but recently explored with learning frameworks [39].
- Some direct methods fuse events and IMU [38].
- Recently, learning methods that fuse images and events have been proposed [37].

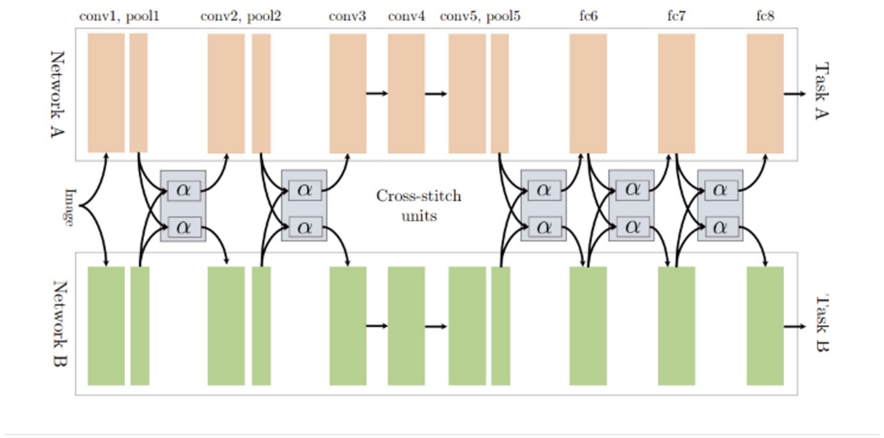




Approach: multitask learning







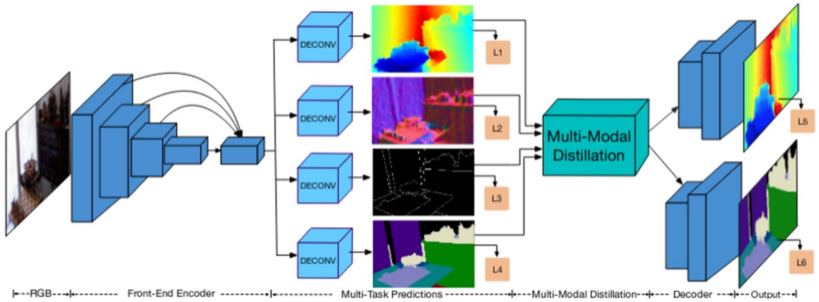
98

- [43] M. Crawshaw, “Multi-Task Learning with Deep Neural Networks: A Survey.” 2020.

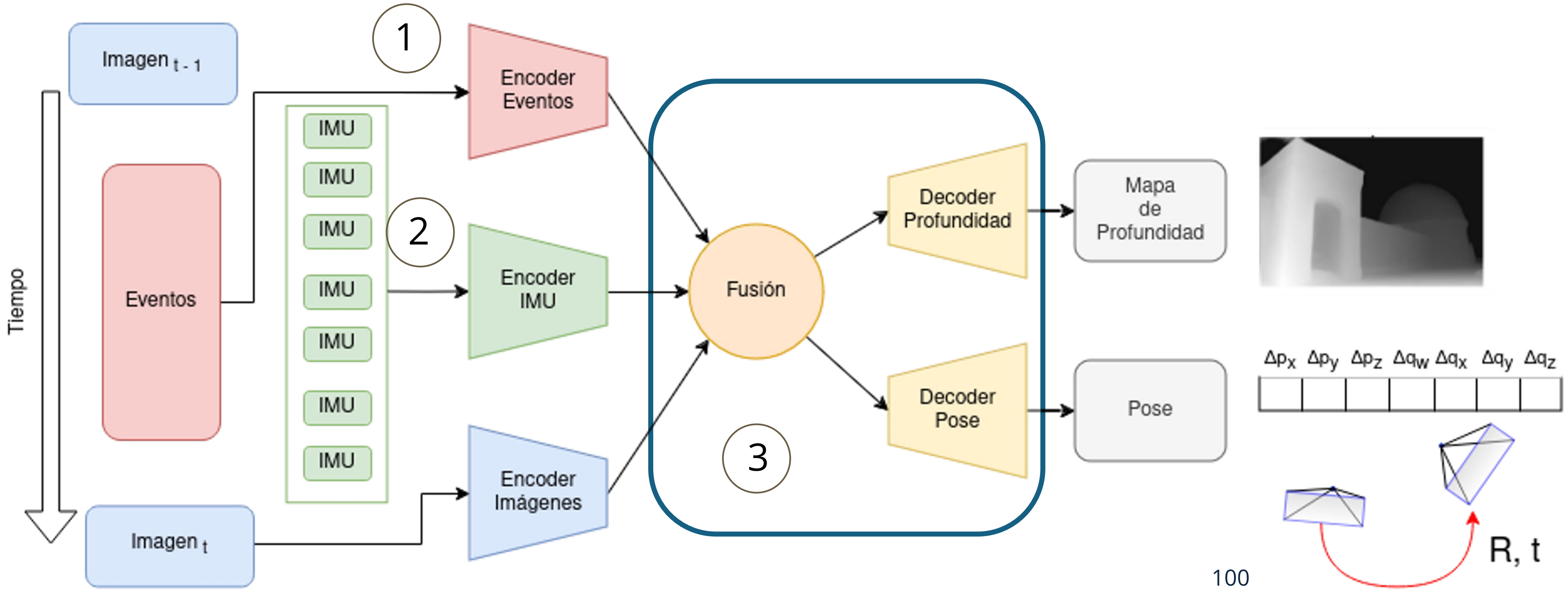
Approach: sensor fusion

- Multimodal learning has shown that it is possible to improve the state of the art through multiple types of input.
- Event cameras such as DAVIS346 give us events, images and inertial data, we can try to use all of them to improve the results.
- There are works that mix events with images [1], lidar [33] , and others.

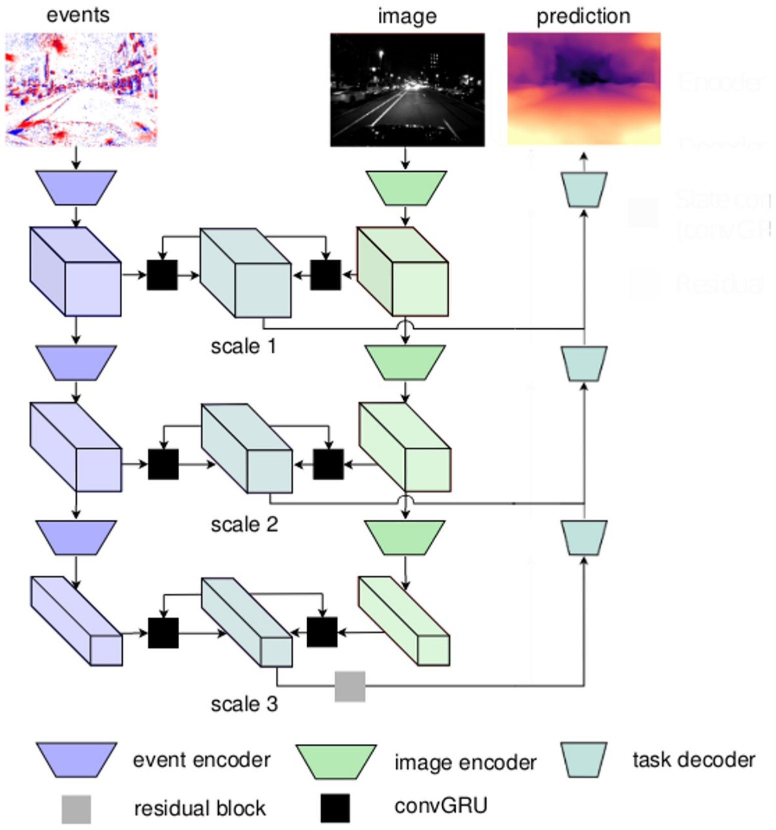
DAVIS346:
IMU+EVENTS+FRAMES



Challenge: How to mix the different inputs (sensor fusion) and how to use them for different tasks?

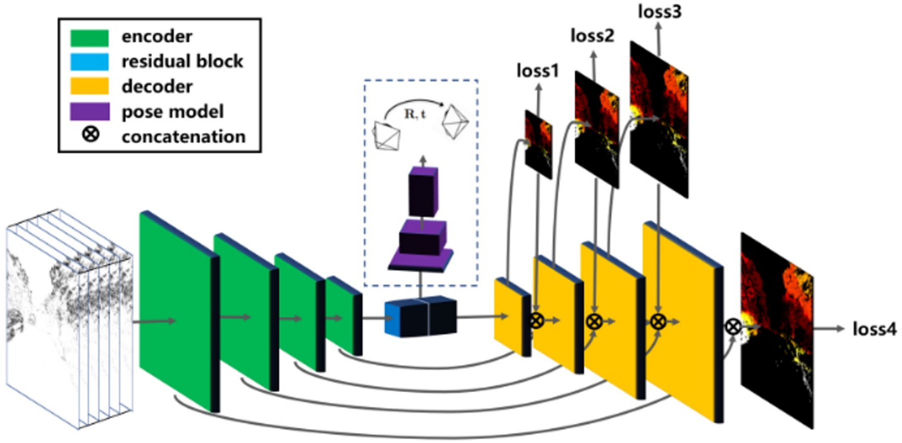


Base architectures



RAMNET

- [1] Daniel Gehrig, Michelle Rüegg, Mathias Gehrig, Javier Hidalgo-Carrió, Davide Scaramuzza, "*Combining Events and Frames using Recurrent Asynchronous Multimodal Networks for Monocular Depth Prediction*". 2021.



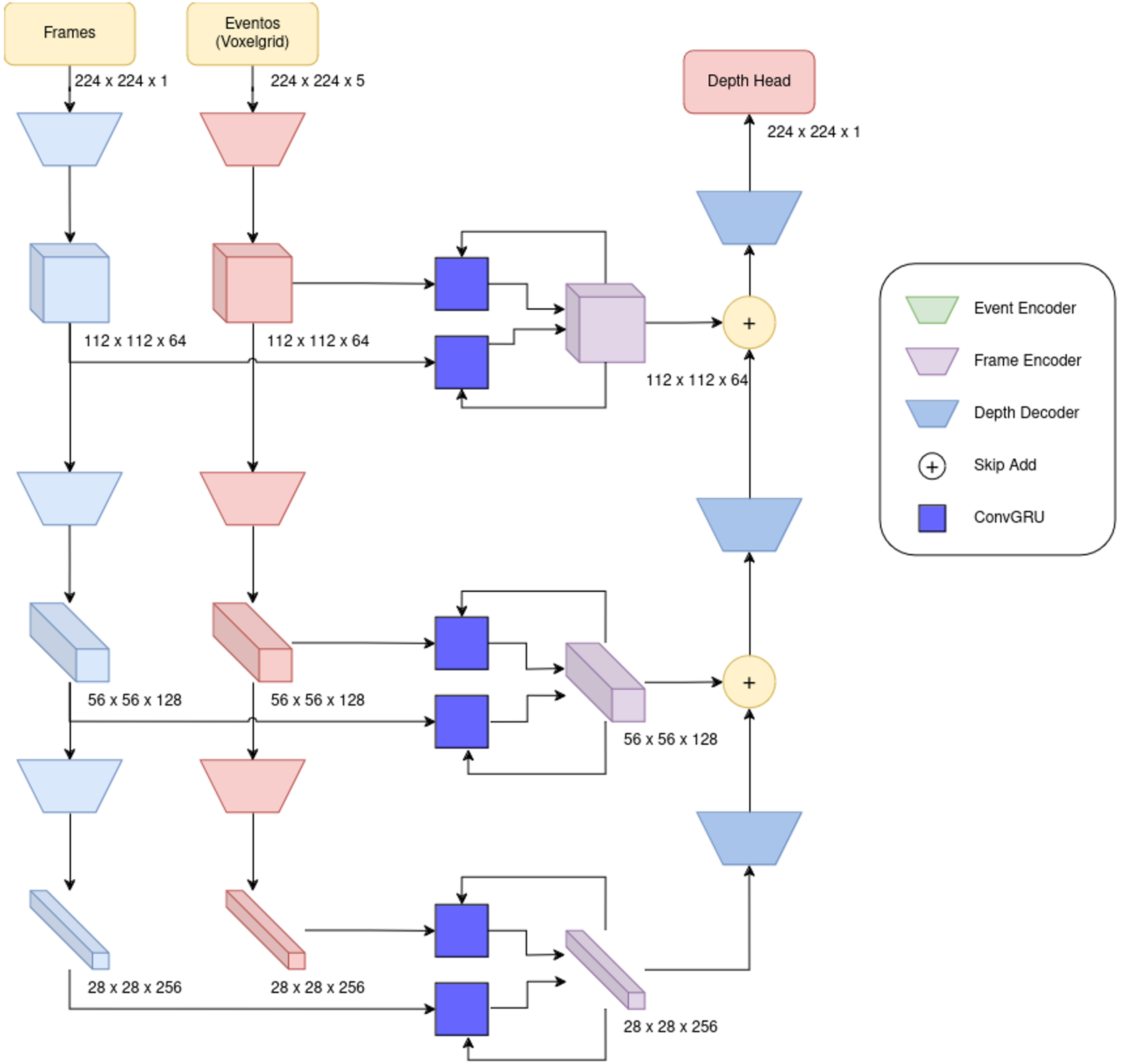
VIT

- [2] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, Kostas Daniilidis, "*Unsupervised Event-based Learning of Optical Flow, Depth, and Egomotion*". 2018

Proposed architecture 1

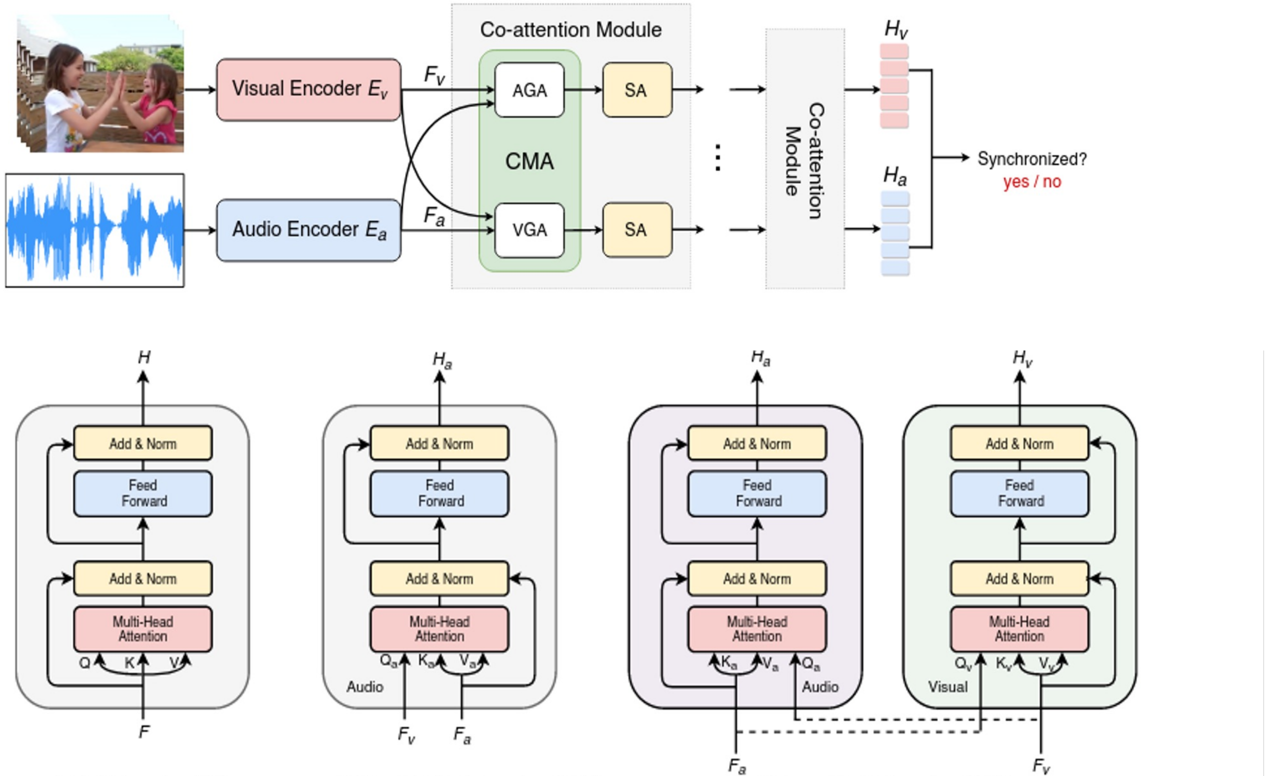
We start from the RAMNet base, challenges:

- How to add Pose estimation?
- How to add IMU?



¿How to add an IMU?

- Mixing inertial features with visual features is neither simple nor straightforward due to the different domain and form of the two.
- We take inspiration from what has been done in the field of vision + speech processing or vision + audio processing [9, 10, 11].



(a) Self-Attention (SA) (b) Visual-Guided Attention (VGA) (c) Cross-Modal Attention (CMA)

[11] Ying Cheng, Ruize Wang, Zhihao Pan, Rui Feng & Yuejie Zhang (2020): Look, Listen, and Attend: Co-Attention Network for Self-Supervised Audio-Visual Representation Learning

Evaluation: Depth Estimation Metrics

- Absolute Relative Error:

$$AbsRel = \frac{1}{N} \cdot \sum_{i}^{N} \frac{|D_i - \hat{D}_i|}{|D_i|}$$

- Mean Error:

$$MAE = \frac{1}{N} \cdot \sum_{i}^{N} |D_i - \hat{D}_i|$$

Reported for different maximum depth cuts (10m, 20m, 30m).

- Accuracies, Percentage of pixels that comply:

$$\max\left(\frac{\hat{D}_i}{D_i}, \frac{D_i}{\hat{D}_i}\right) = \delta < thr$$

With thr = {1.25, 1.252, 1.253}

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N} \cdot \sum_{i}^{N} |D_i - \hat{D}_i|}$$

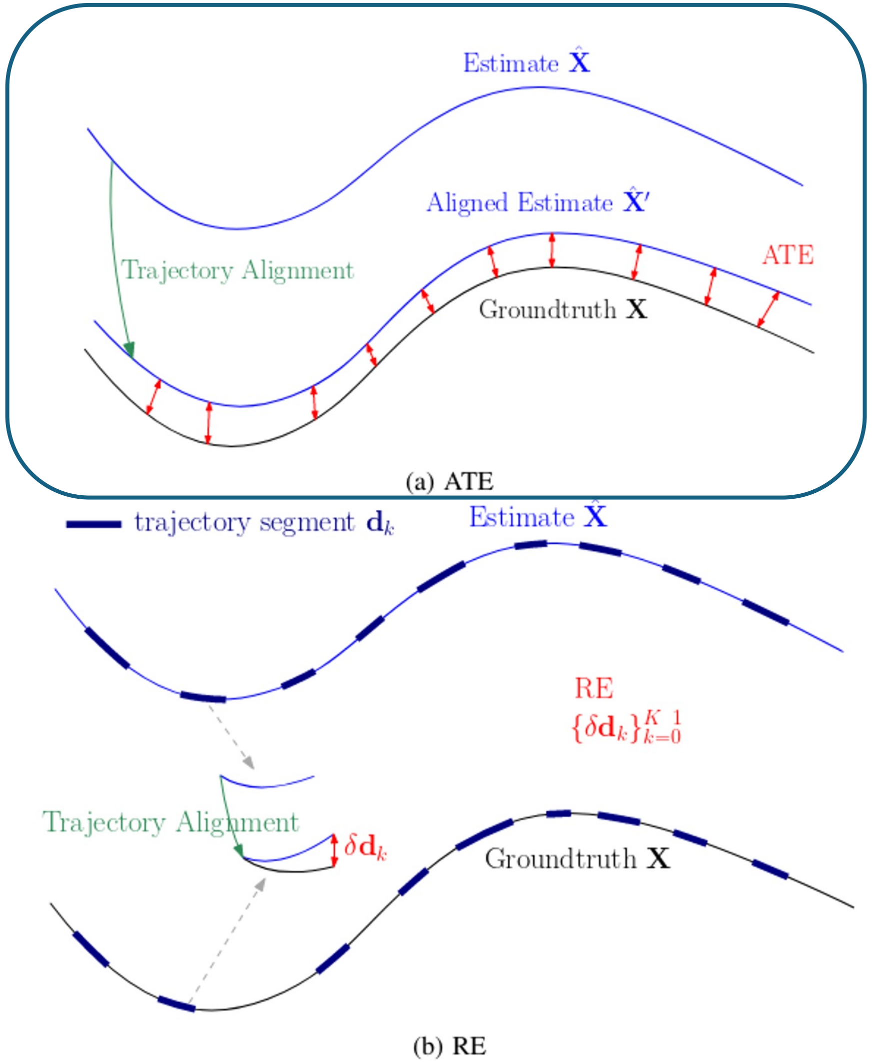
$$RMSELog = \sqrt{\frac{1}{N} \cdot \sum_{i}^{N} |\log(D_i) - \log\left(\hat{D}_i\right)|}$$

Evaluation: Position Estimation Metrics

Absolute Trajectory Error (ATE)

Measures the absolute difference between two complete trajectories. Generally a previous alignment must be made between both trajectories, in the case of VO a scaling operation is also performed.

It is sensitive to the time at which the error occurs, a deviation at the beginning of the trajectory generates higher ATE than if the same error occurs at the end of the trajectory.



[24] Z. Zhang and D. Scaramuzza, "A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry,". *2018*

Evaluation: Position Estimation Metrics

Relative Error (RE)

- Measures the difference between two aligned subpaths or for each estimated point.
- The subpaths correspond to a set of poses that are at a fixed time, distance or number of keyframes.

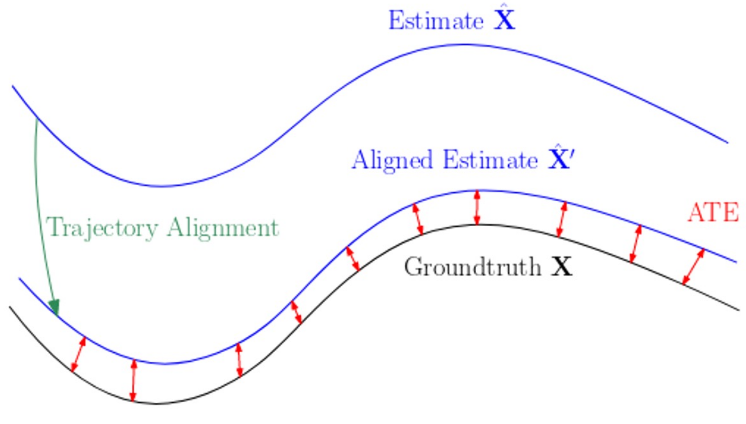
In addition for comparability, we used the errors defined in Zhao et al [2]:

- Relative Pose Error (RPE):

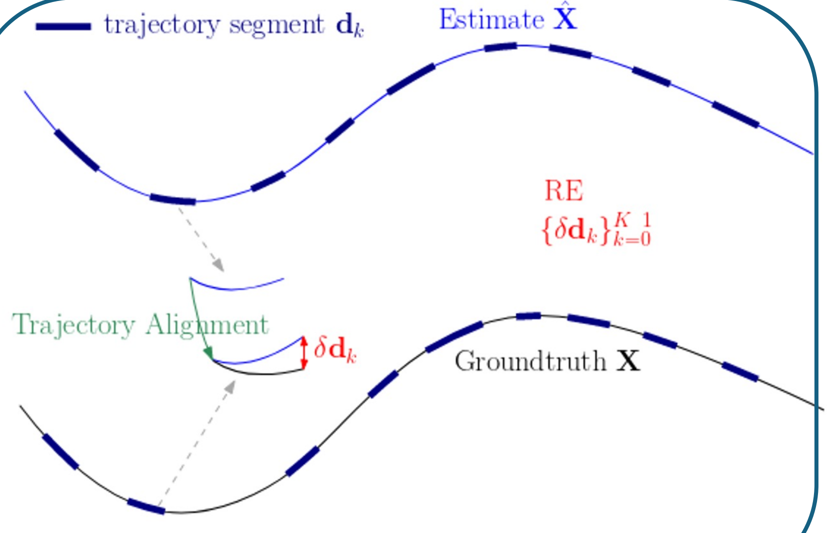
$$\arccos\left(\frac{t_{pred} \cdot t_{gt}}{||t_{pred}||_2 ||t_{gt}||_2}\right)$$

- Relative Rotation Error (RRE):

$$\left|\left|\operatorname{logm}\left(R_{pred}^{T} R_{gt}\right)\right|\right|_2$$



(a) ATE

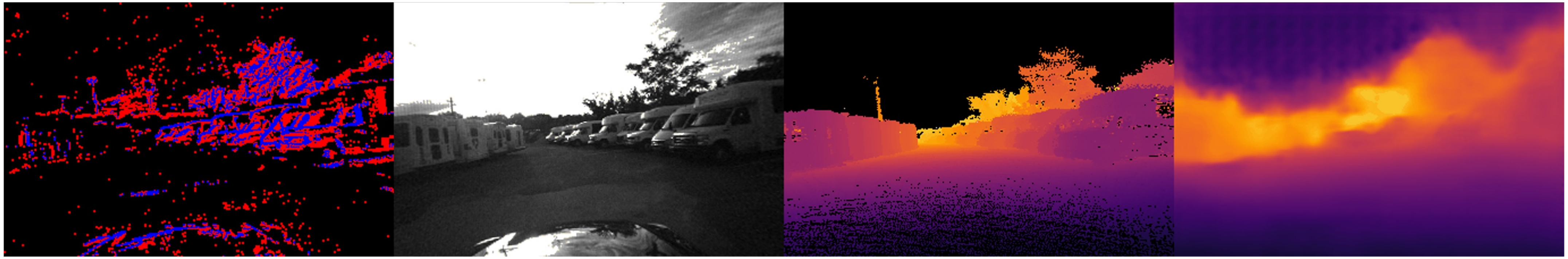


(b) RE

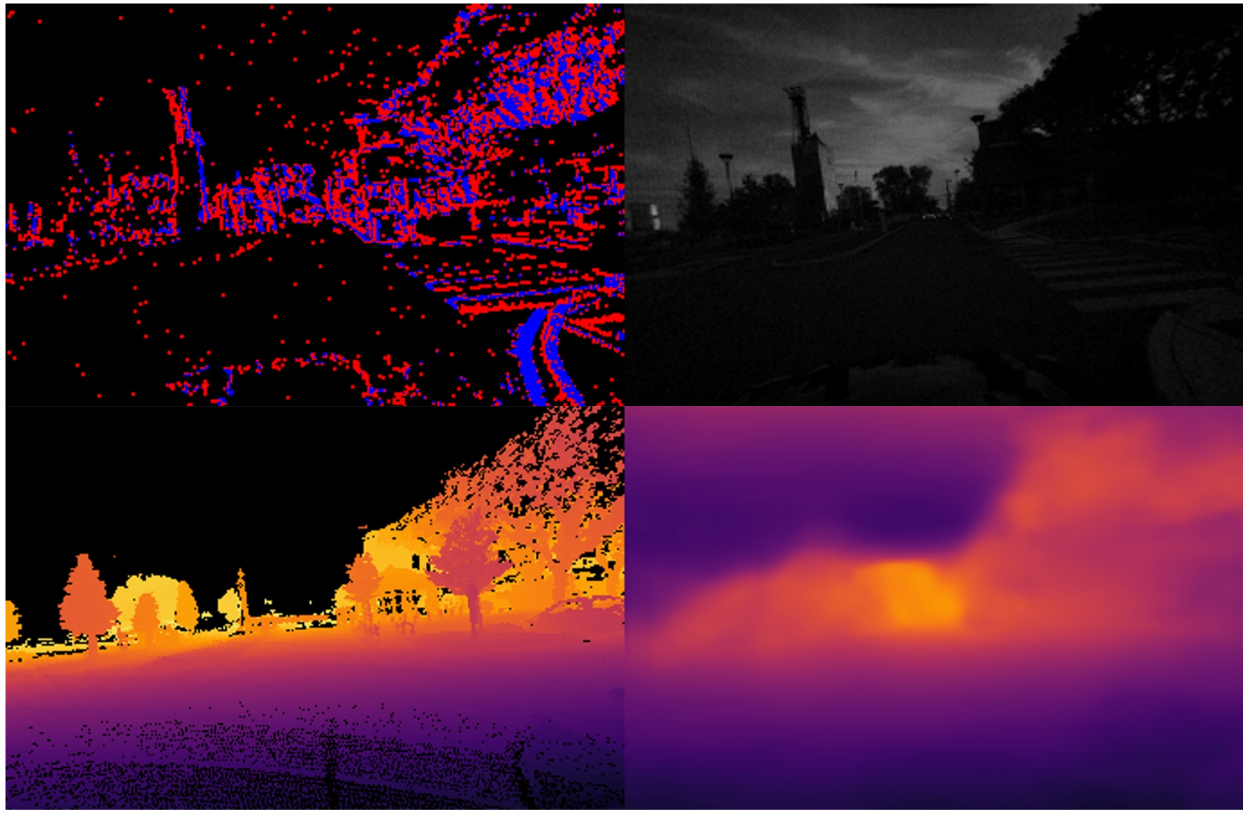
MVSEC experiments - RAMNet + IMU

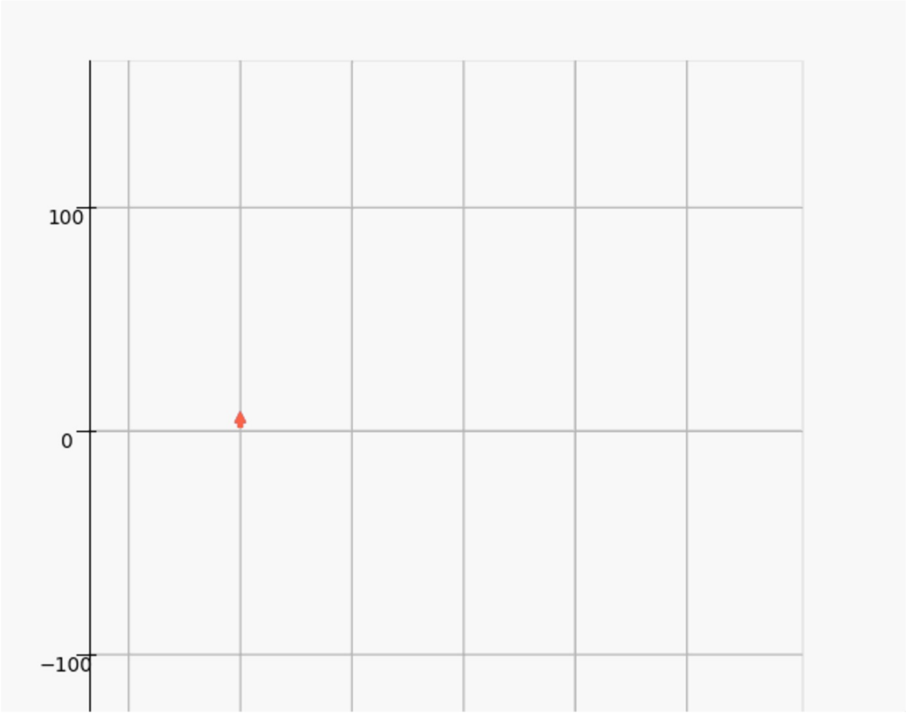
Depth Results

	Modelo	AbsRel ↓	RMSE Log ↓	SILog ↓	δ < 1.25 ↑	δ < 1.25^2 ↑	δ < 1.25^3 ↑	MAE 10m ↓	MAE 20m ↓	MAE 30m↓	MAE ↓
outdoor day 1	RAMNet baseline	0.304	0.282	0.044	0.539	0.778	0.877	1.337	**2.044**	2.660	4.728
	RAMNet + IMU (LSTM)	**0.284**	0.392	0.082	**0.586**	**0.799**	**0.896**	**1.347**	2.129	2.665	**4.247**
	RAMNet + IMU (Transformer)	0.288	0.374	0.077	**0.609**	**0.817**	**0.906**	1.374	2.115	**2.619**	**4.089**
outdoor night 1, 2, 3	RAMNet baseline	**0.360**	**0.497**	**0.115**	**0.474**	**0.685**	**0.819**	**1.631**	**2.655**	**3.539**	**5.248**
	RAMNet + IMU (LSTM)	0.391	0.543	0.146	0.465	0.671	0.800	1.996	3.058	3.892	5.429
	RAMNet + IMU (Transformer)	0.402	0.580	0.160	0.436	0.641	0.771	1.940	3.152	4.126	5.734



MVSEC experiments - RAMNet + IMU





RAMNet + IMU (LSTM) en outdoor_day1

DSEC experiments - RAMNet + IMU

Depth Results

Experimento	AbsRel ↓	RMSE Log ↓	SILog ↓	δ < 1.25 ↑	δ < 1.25^2 ↑	δ < 1.25^3 ↑	10m ↓	20m ↓	30m ↓	mean error ↓
Base (E + F) → D (baseline)	**0.114**	**0.155**	**0.013**	**0.871**	**0.978**	**0.995**	**0.563**	**1.312**	**1.830**	**2.538**
Base (E + F) → D + P	0.116	0.159	0.013	0.865	0.976	0.995	0.586	**1.301**	1.834	2.634
Encoder IMU LSTM + Pose CoAttn	**0.111**	**0.152**	**0.012**	**0.880**	**0.980**	**0.995**	0.577	**1.256**	**1.780**	**2.470**
Encoder IMU LSTM + Pose CNN	0.113	0.153	0.012	0.877	0.980	0.995	0.594	1.288	1.841	2.491
Encoder IMU Transf + PoseCoattn	0.113	0.157	0.013	0.870	0.977	0.994	0.594	1.258	1.803	2.569

E: Eventos, F: Frames (Imágenes), I: IMU

DSEC experiments - RAMNet + IMU

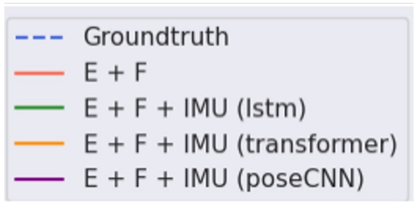
Odometry Results

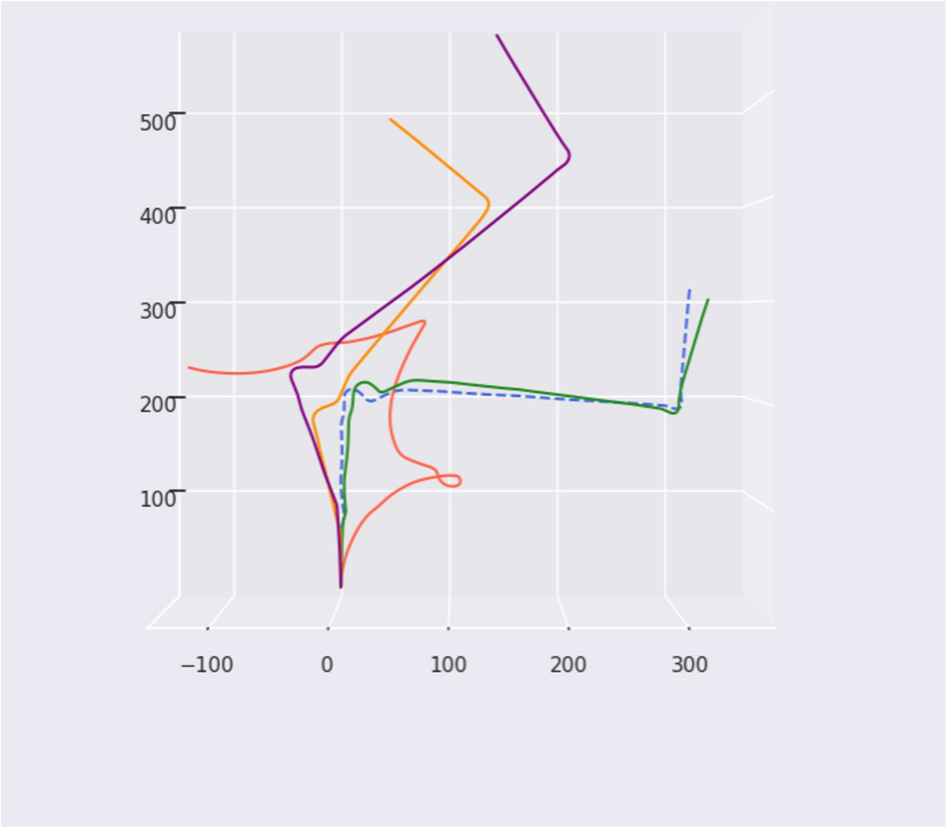
Experimento	**APE trans ↓**	**APE rotation ↓**	**RPE trans (Δ=1m) ↓**	**RPE rot deg (Δ=1m)↓**
Base (E + F) → D (baseline)	-	-	-	-
Base (E + F) → D + P	36.526	87.106	0.203	0.807
Encoder IMU LSTM + Pose CoAttn	18.754	36.240	0.096	0.446
Encoder IMU LSTM + Pose CNN	11.927	71.174	0.188	0.523
Encoder IMU Transf + PoseCoattn	**4.465**	**25.269**	**0.137**	**0.123**

E: Eventos, F: Frames (Imágenes), I: IMU

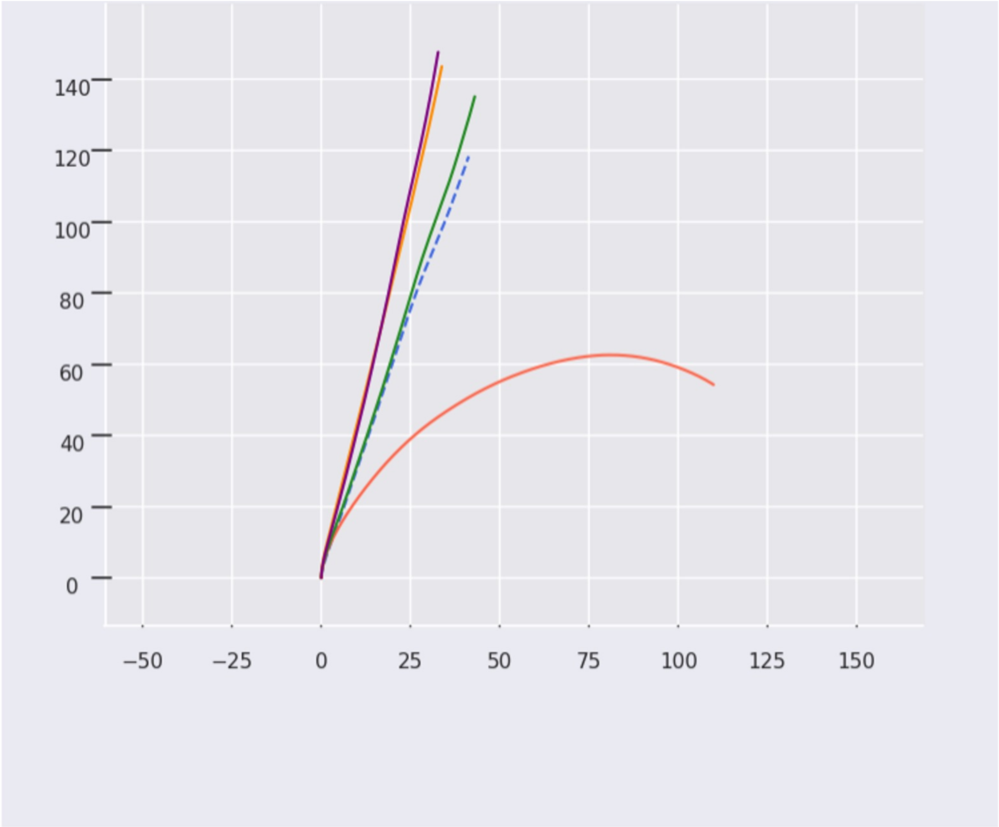
DSEC experiments - RAMNet + IMU

Odometry Results





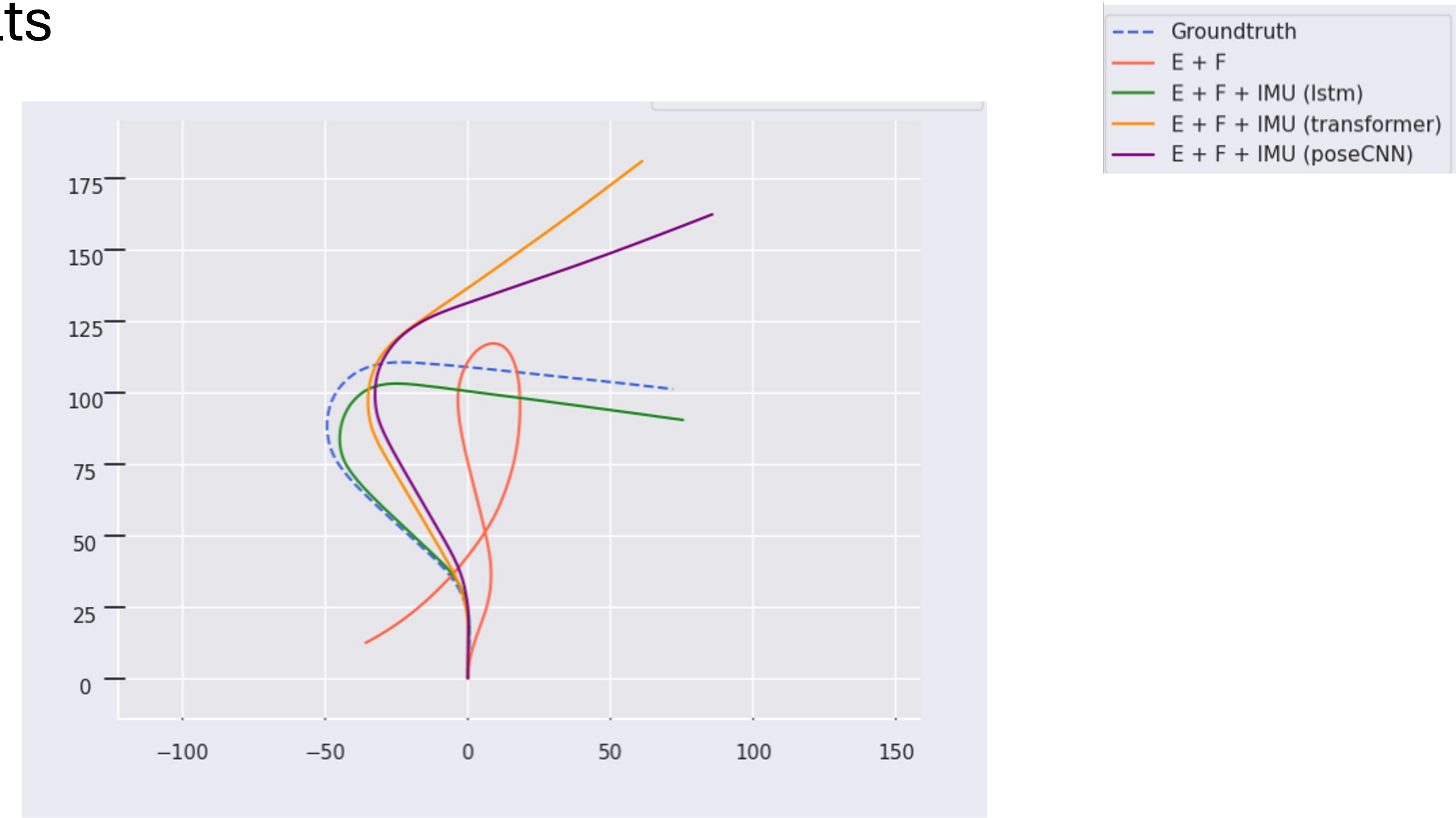
zurich_city_10_a



zurich_city_11_a

DSEC experiments - RAMNet + IMU

Odometry Results



zurich_city_04_a

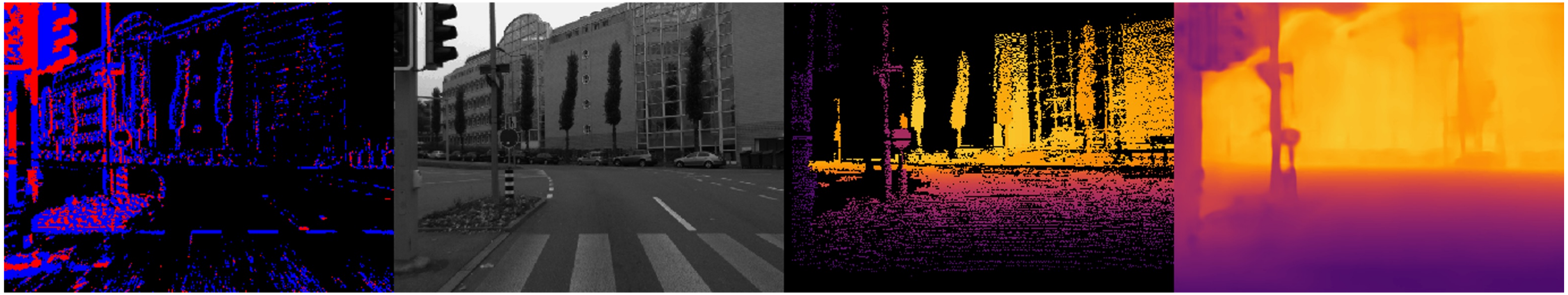
DSEC experiments - RAMNet + IMU

In Depth Results: Input / Output Study

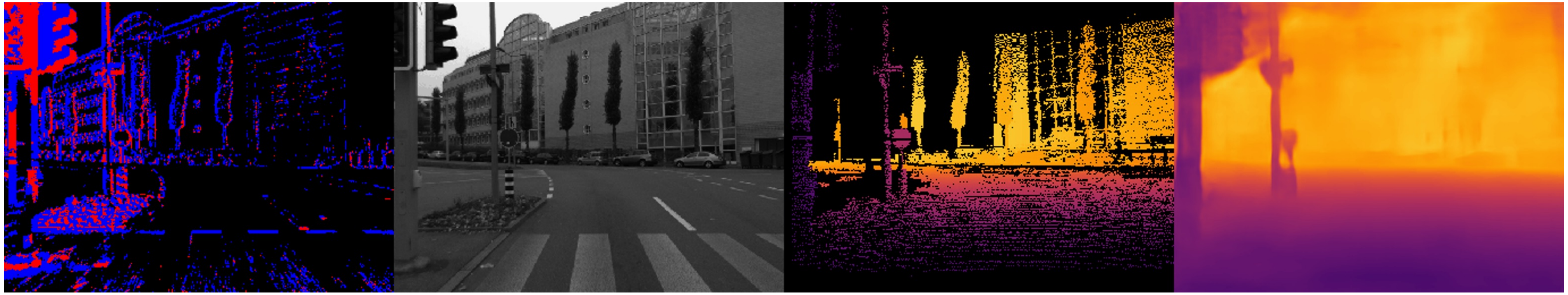
Experimento	AbsRel **↓**	RMSE Log ↓	SILog ↓	δ < 1.25 ↑	δ < 1.25^2 ↑	δ < 1.25^3 ↑	10m ↓	20m ↓	30m ↓	mean error ↓
Base (E + F) → D (baseline)	**0.114**	**0.155**	**0.013**	**0.871**	**0.978**	**0.995**	**0.563**	**1.312**	**1.830**	**2.538**
F → Depth	0.165	0.226	0.028	0.768	0.929	0.977	0.958	2.005	2.626	3.506
E → Depth	0.146	0.188	0.018	0.812	0.956	0.990	0.624	1.741	2.563	3.309
E + F → Pose + Depth	0.116	0.159	0.013	0.865	0.976	0.995	0.586	**1.301**	1.834	2.634
E + I → Pose + Depth	0.148	0.188	0.018	0.811	0.960	0.991	0.856	1.823	2.492	3.117
E + F + I → Pose + Depth	**0.111**	**0.152**	**0.012**	**0.880**	**0.980**	**0.995**	0.577	**1.256**	**1.780**	**2.470**

E: Eventos, F: Frames (Imágenes), I: IMU

Base (E + F) → Depth (baseline)



E + F + I → Pose + Depth



Event-based 6D pose estimation of moving objects

Rodrigo Verschae

Institute of Engineering Sciences

Universidad de O'Higgins

rodrigo@verschae.org

Universidad de O'Higgins

Research: DB for object pose estimation

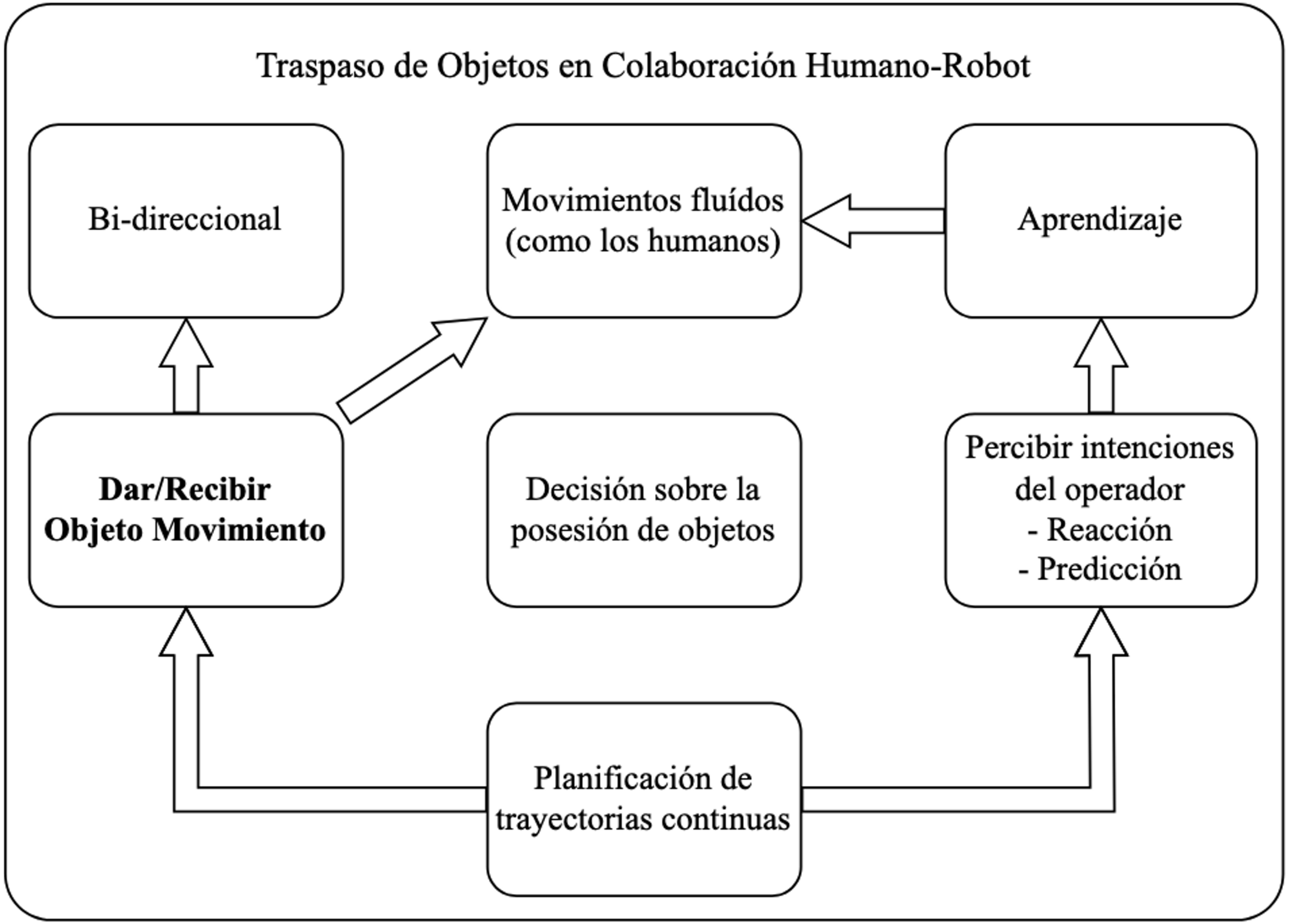
Ignacio Bugueno

Motivation: High-speed human-robot collaboration

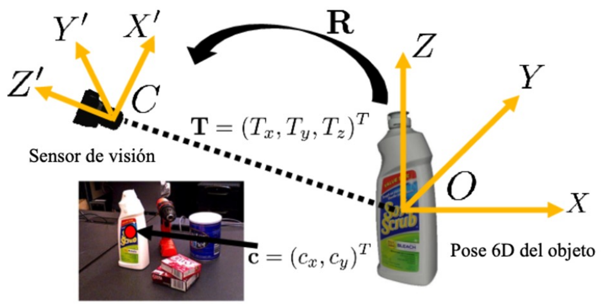
"SecondHands: A Collaborative Maintenance Robot for Automated Warehouses"

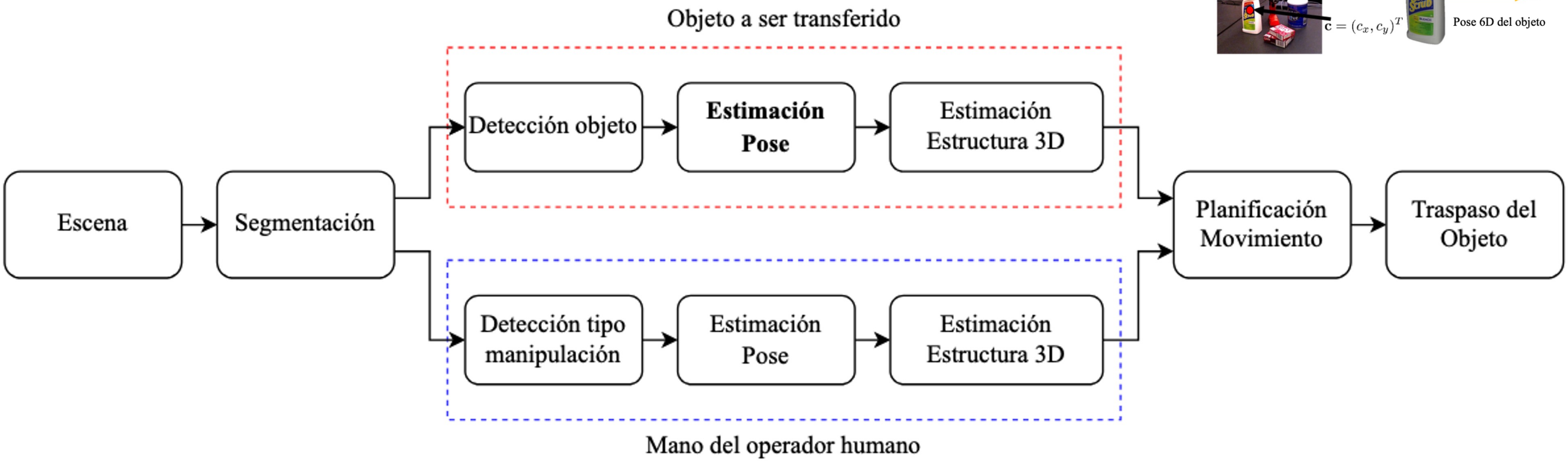
"High-speed, Small-deformation Catching of Soft Objects based on Active Vision and Proximity Sensing"

Motivación: Traspaso de objetos en colaboración humano-robot

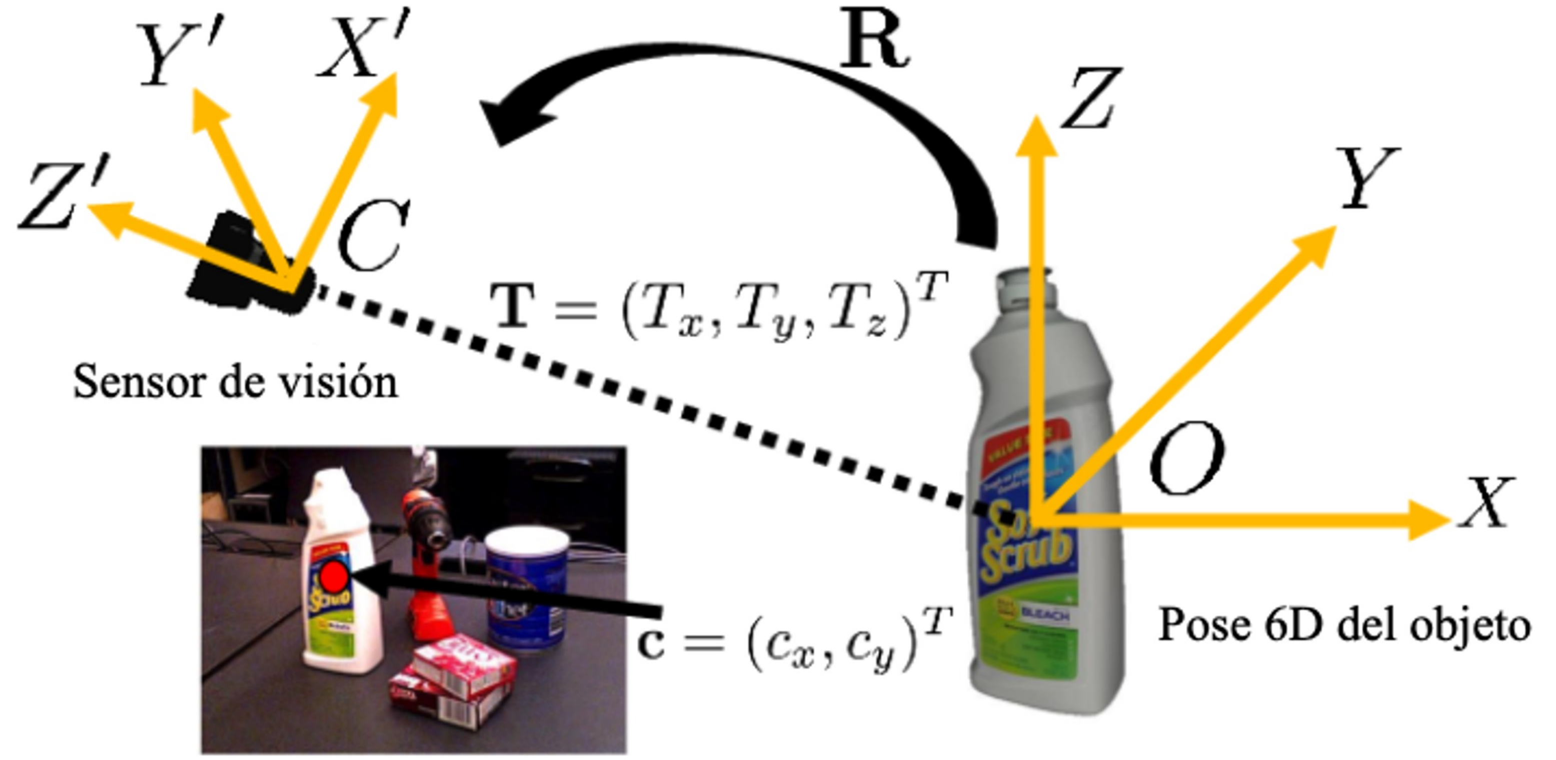


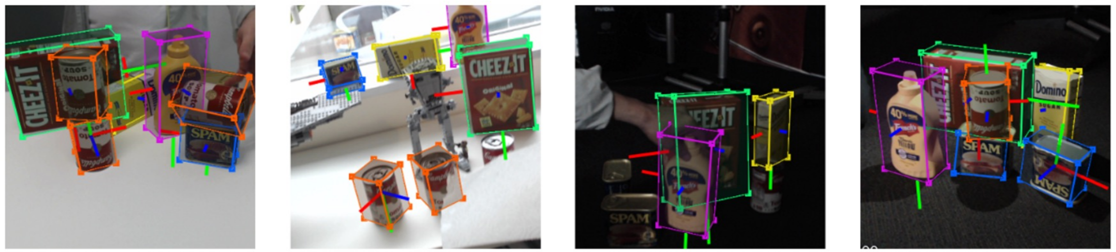
Motivación: Traspaso de objetos en colaboración humano-robot



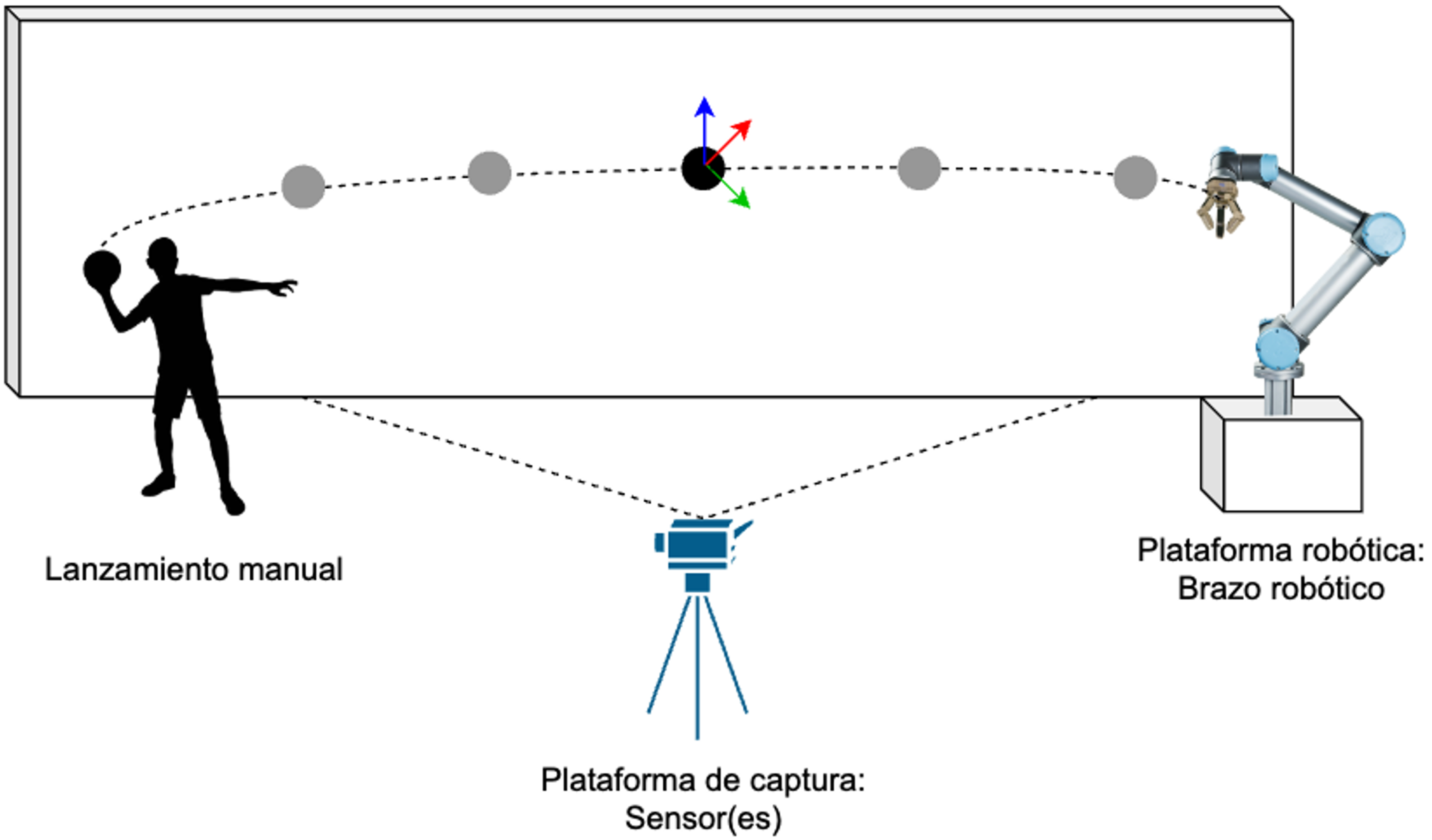


Motivación: La estimación de la pose 6D es clave en *handover*





Motivación: ¿Y si abordamos estimación de pose a alta velocidad?



Moving6DPoSe database objects

(1) Almohada

(2) Árbol

(3) Avión

(4) Boomerang

(5) Caja

(6) Caja

(7) Carro

(8) Cilindro

(9) Dinosaurio

(10) Frisbee

(11) Jarrón

(12) Cilindro

(13) Contene-dor

(14) Paleta

(15) Pelota

(16) Sombrero

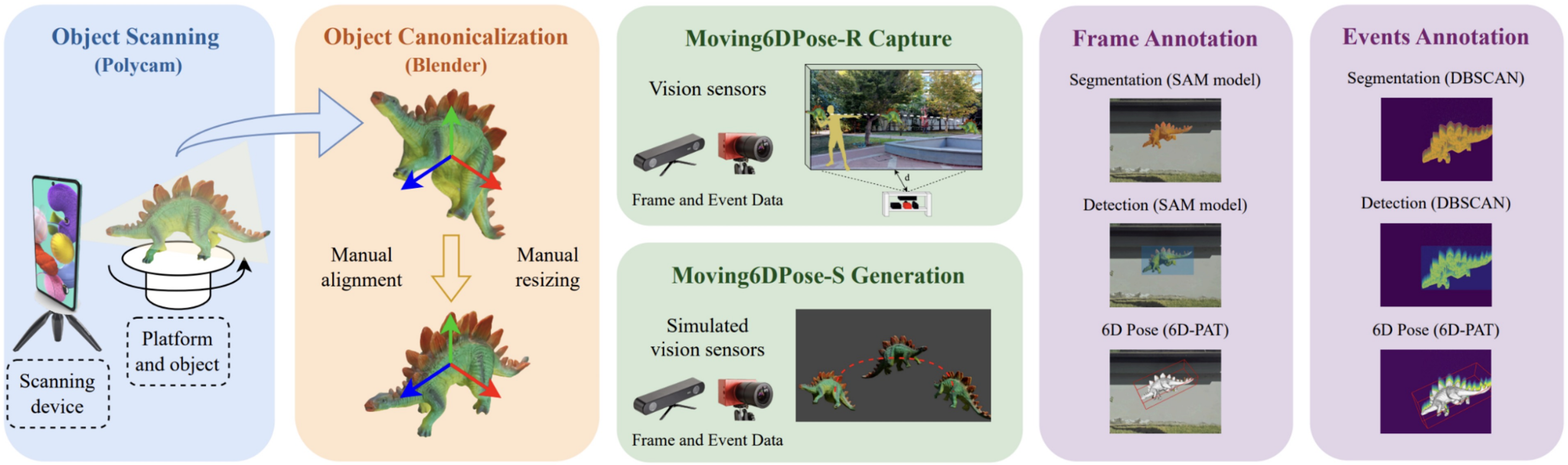
(17) Tarro

(18) Taza

(19) Toalla

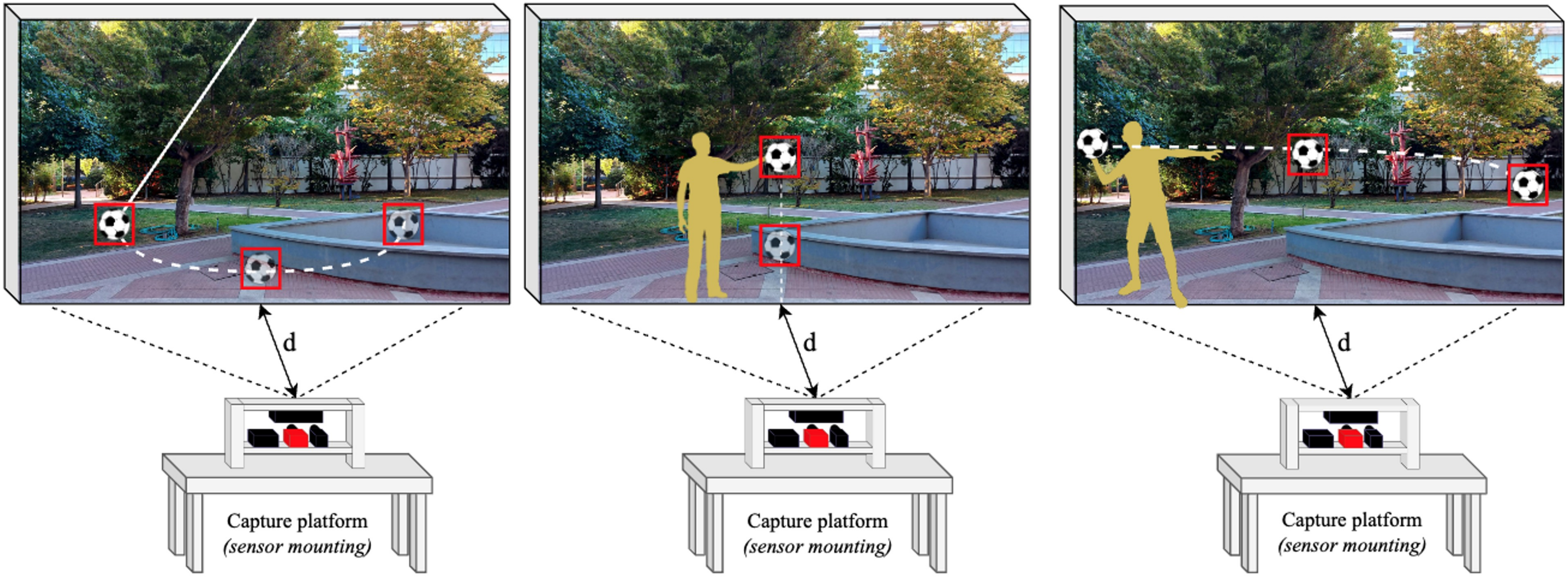
(20) Zapatilla

Moving6DPoSe database collection and annotation



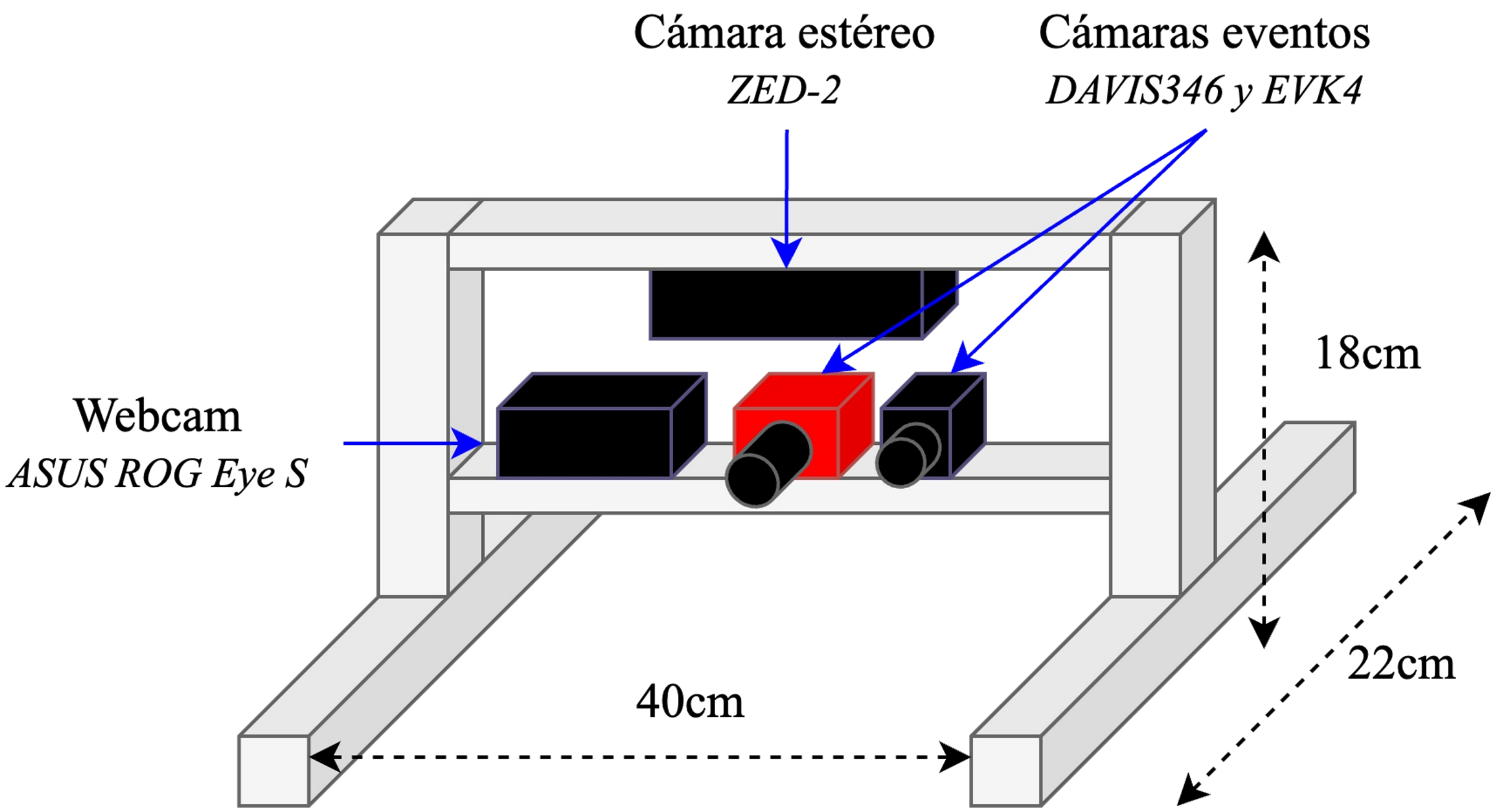
Moving6DPoSe-R: Real Dataset

Set-ups experimental



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Capture platform: spatial (FOV)-temporal synchrony

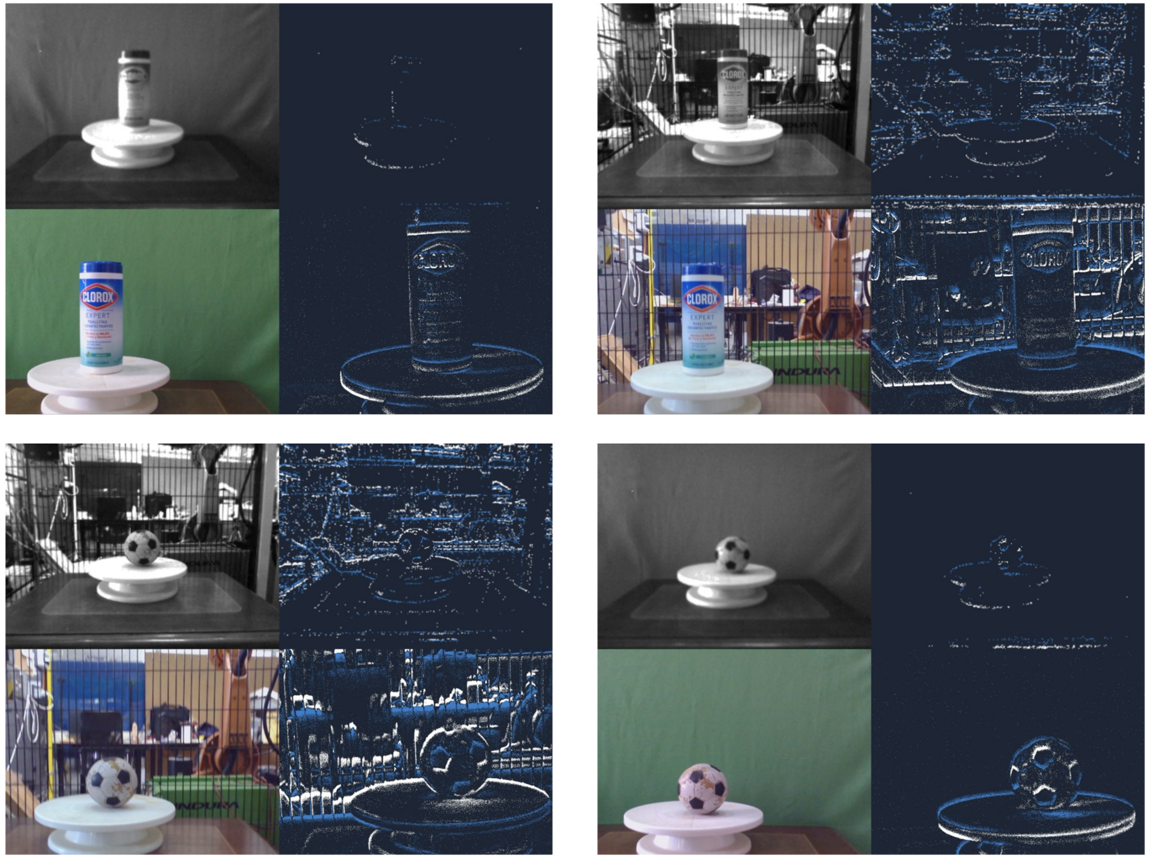


Relevant technical characteristics of vision sensors employed

Sensor	Lens	Data	Dimension	Field-of-view	Resolution	$\text{Rate}_{\text{Generation}}$	$\text{Rate}_{\text{Storage}}$
ASUS ROG Eye S	Monocular	RGB Frame	2.87cm high 8.1cm wide 1.65cm deep	78.0°	640x480	Fixed (30Hz)	Fixed (30Hz)
ZED-2	Stereo	RGB frame per lens	3.0cm high 17.5cm wide 3.3cm deep	H: 92-103° V: 61-71°	640x480	Fixed (15Hz)	Fixed (15Hz)
		Stereo RGB frame			1280x480		
DAVIS346	Monocular	Gray frame	4cm high 6.0cm wide 2.5cm deep	H: 29.9-113° V:22.7-99.7° D: 36.9-215°	346x260	Fixed (30Hz)	Fixed (30Hz)
		Events			346x260	Variable (1MHz)	Fixed (30Hz)
Prophesee EVK4	Monocular	Events	3.0cm high 3.0cm wide 3.6cm deep	H: 41.4° V: 23.6° D: 47.0°	1280x720	Variable (10KHz)	Variable (10KHz)

Results

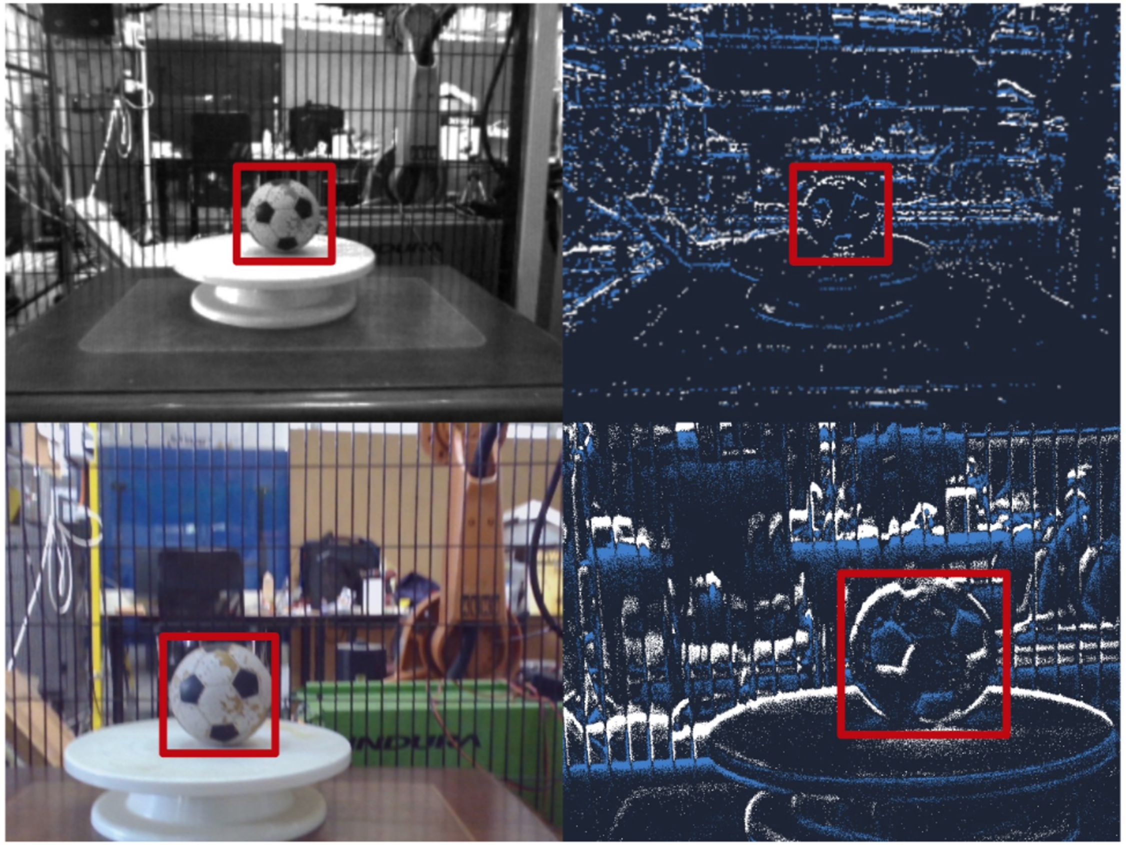
- Captured database
- 8 objects analyzed
- 4 sensors used
- 4 different scenarios
- 2 types of illumination (artificial and natural)
- 2 backgrounds (uniform and non-uniform)
- 5 captures x object (for each scenario)
- Each capture has a variable duration depending on the scenario exp (between 1 to 8 s)
- Total ~ 2,560 captures (RGB and events)

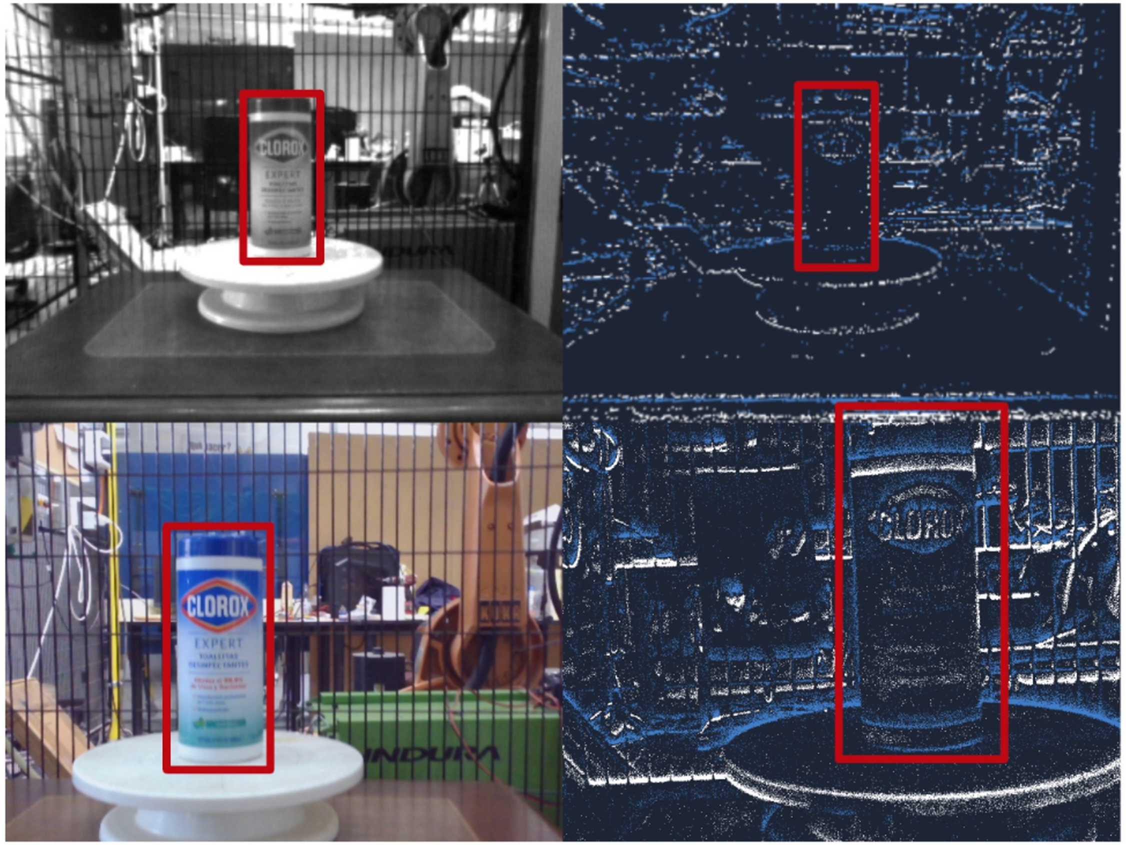


Annotations for segmentation, classification and detection tasks

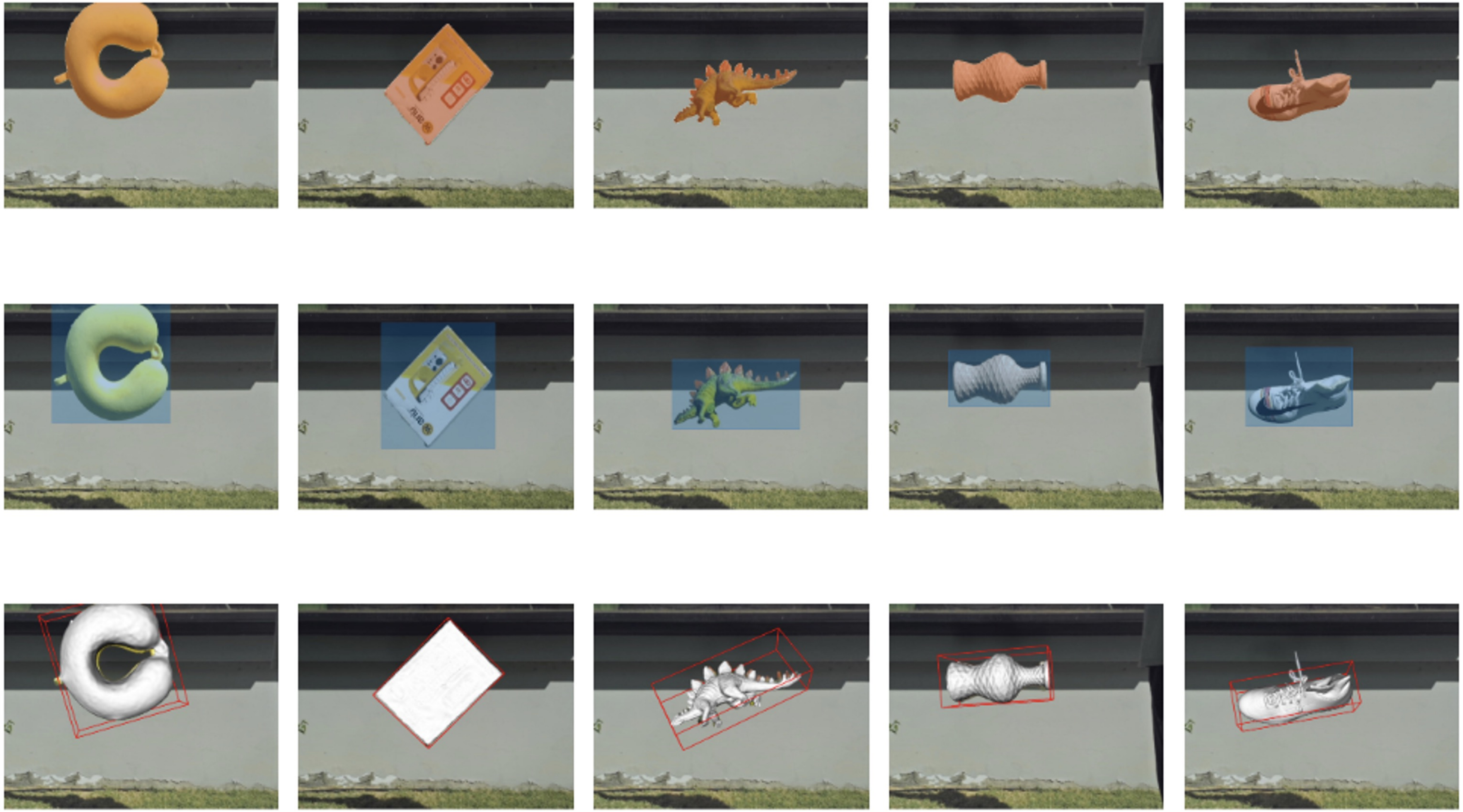
Results: annotations for segmentation, classification, detection, etc.

Annotations for conventional RGB cameras and event cameras

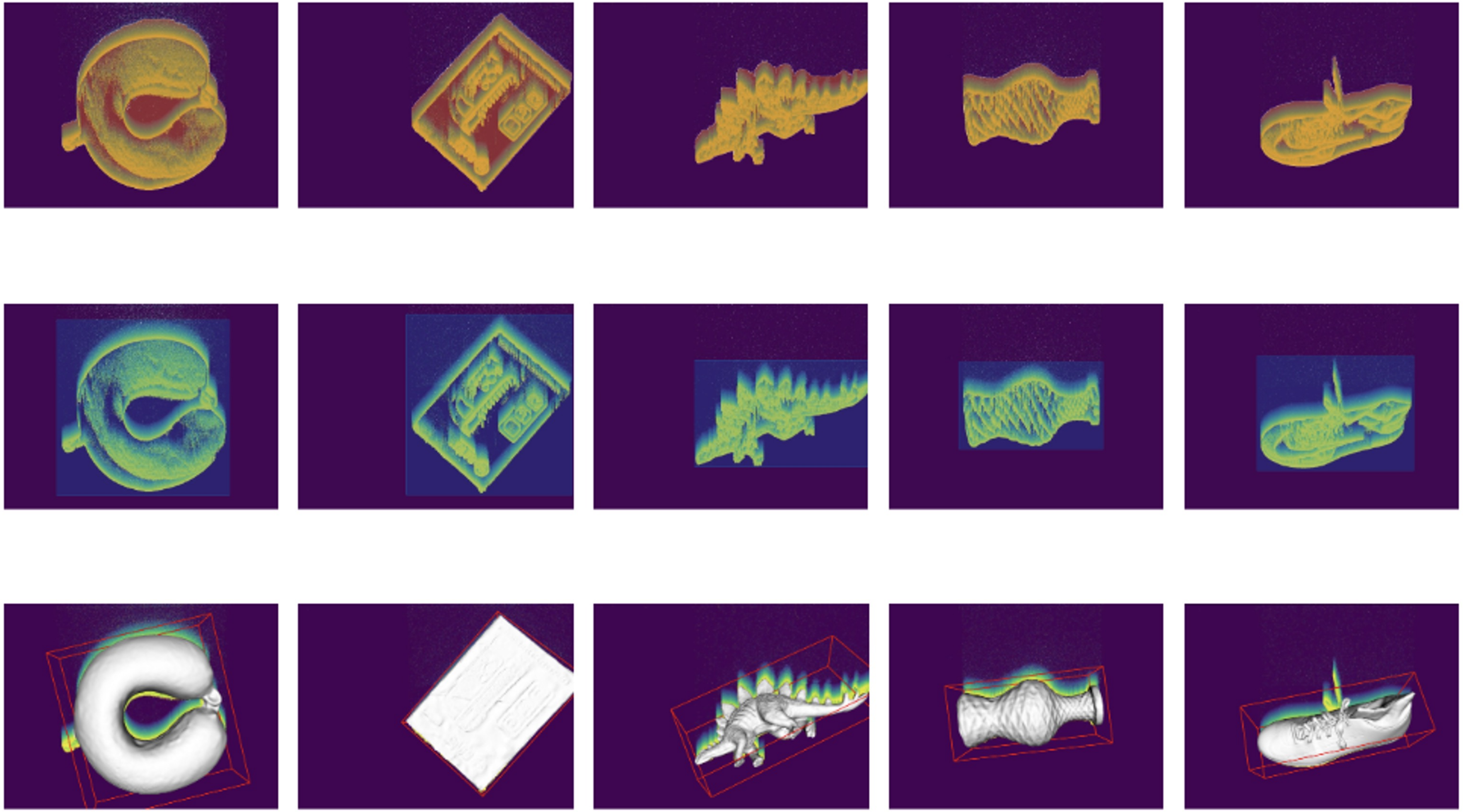




Moving6DPoSe-R dataset annotations



Moving6DPoSe-R dataset annotations



Moving6DPoSe-S: Synthetic Dataset

Resultados

Base de datos simulada
(con foco a domain randomization)

- 20 objetos analizados

- 4 sensores simulados
  - Cámara RGB
    - Webcam ASUS ROG Eye S
    - ZED-2
  - Cámara de eventos
    - DAVIS346
    - Prophesee EVK4

- Variaciones de luminosidad

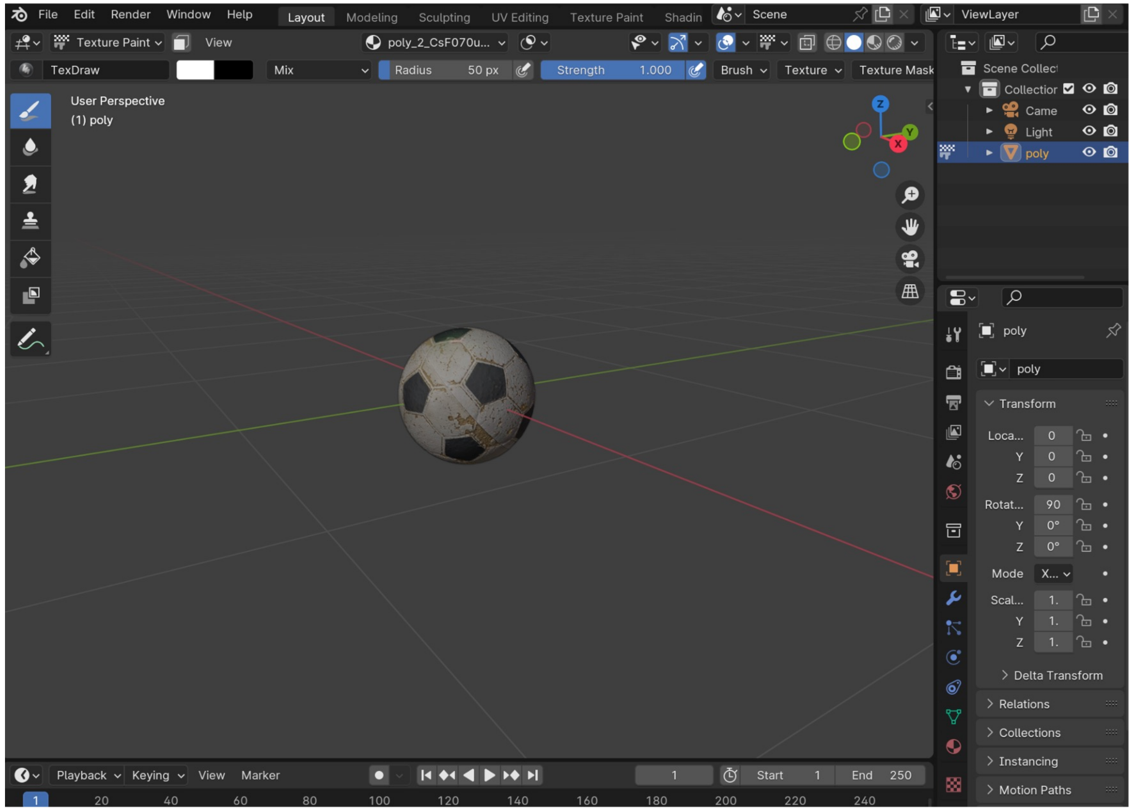
Cámara RGB simulada

Cámara de eventos simulada

Anotaciones para tareas de segmentación, clasificación y pose 6D

Integración de modelos escaneados a Blender

- Blender
- Objetos escaneados con PolyCam
- Librería *bpy (API de Blender para Python)*
- Simulador de eventos *IECBS, V2E*



Integración de modelos escaneados a Blender

(1) Almohada (2) Árbol (3) Avión (4) Boomerang (5) Caja

(6) Caja (7) Carro (8) Cilindro (9) Dinosaurio (10) Frisbee

(11) Jarrón (12) Cilindro (13) Contenedor (14) Paleta (15) Pelota

(16) Sombrero (17) Tarro (18) Taza (19) Toalla (20) Zapatilla

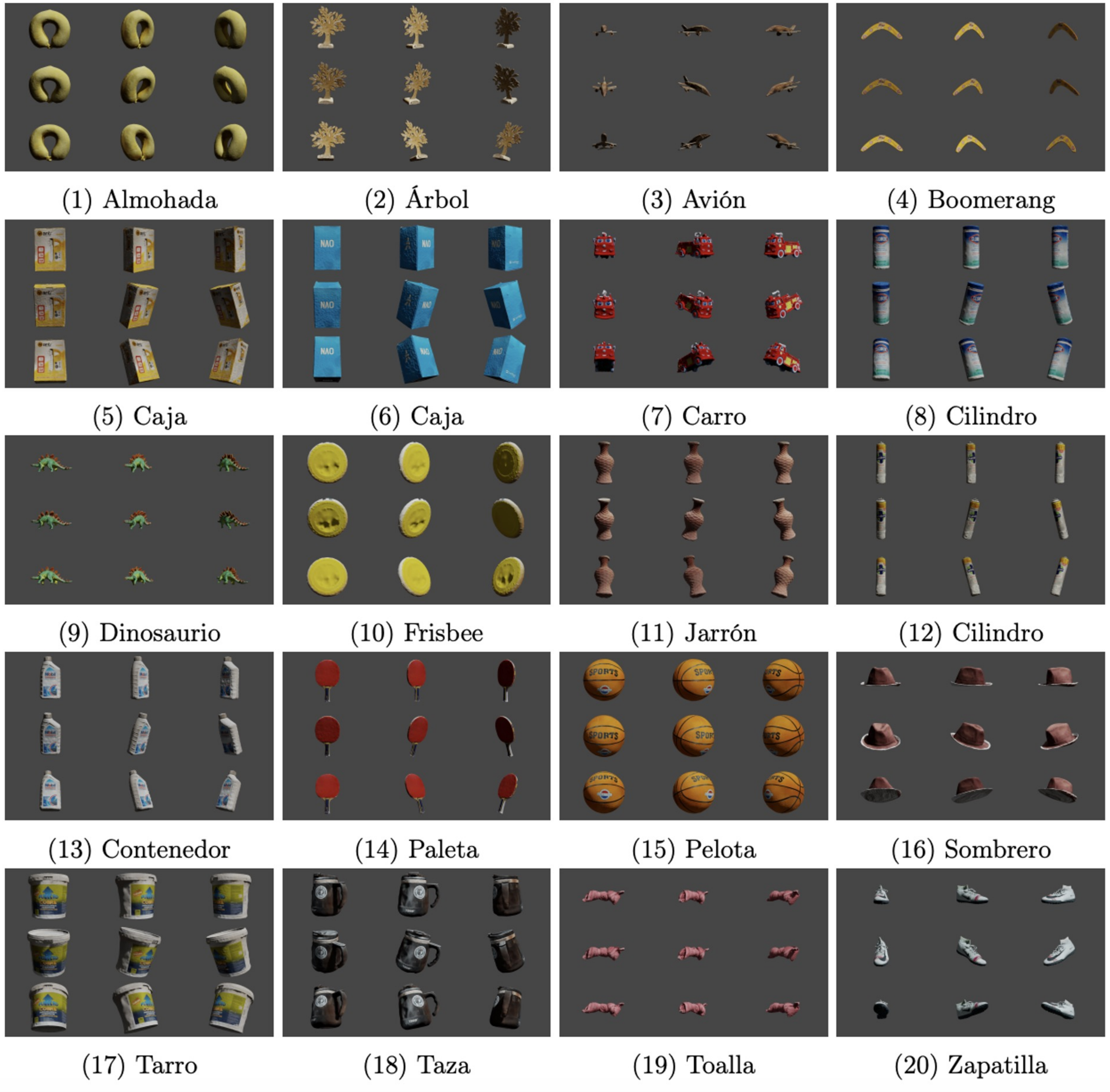
Simulación de objetos en movimiento

(1) Zapatilla

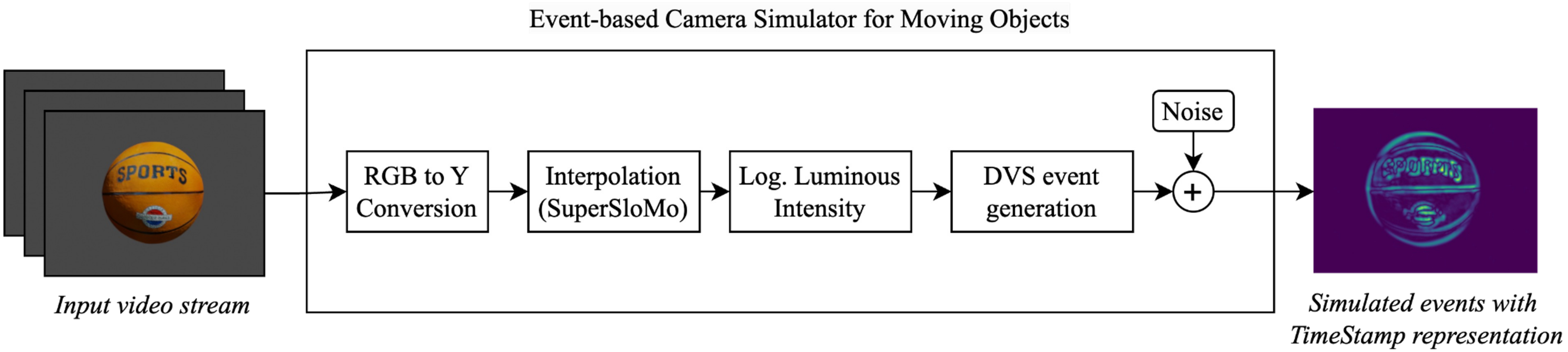
(2) Dinosaurio

(3) Avión

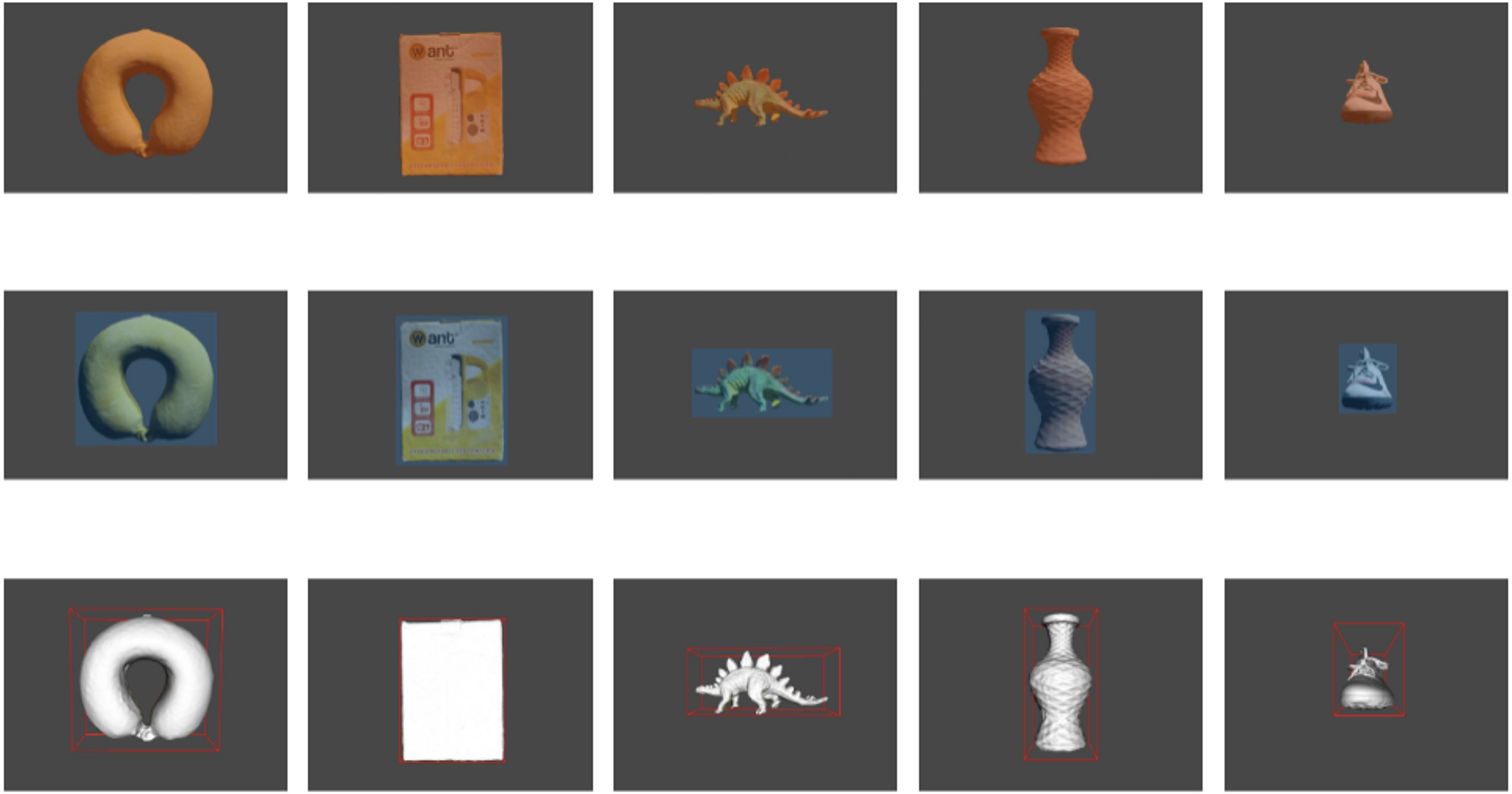
Resultados: base de datos con variación de perspectivas



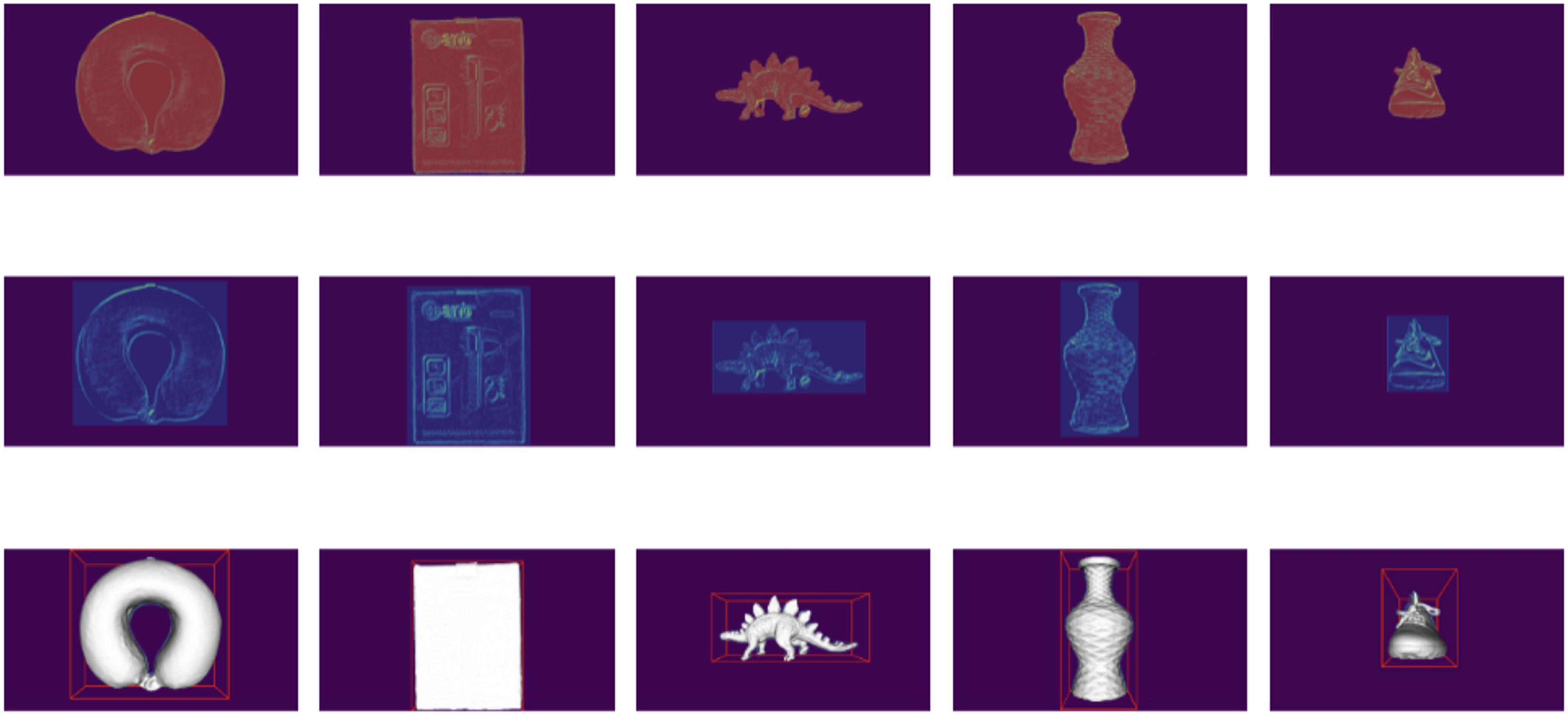
¿Cómo simulamos los eventos en Blender?



Moving6DPoSe-S dataset annotations



Moving6DPoSe-S dataset annotations



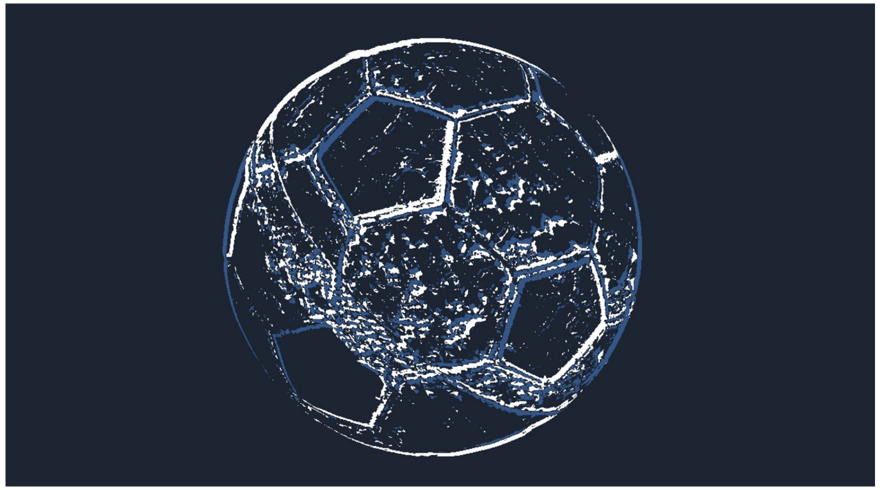
Results: bbdd with resolution variation (depending on sensor)

Webcam

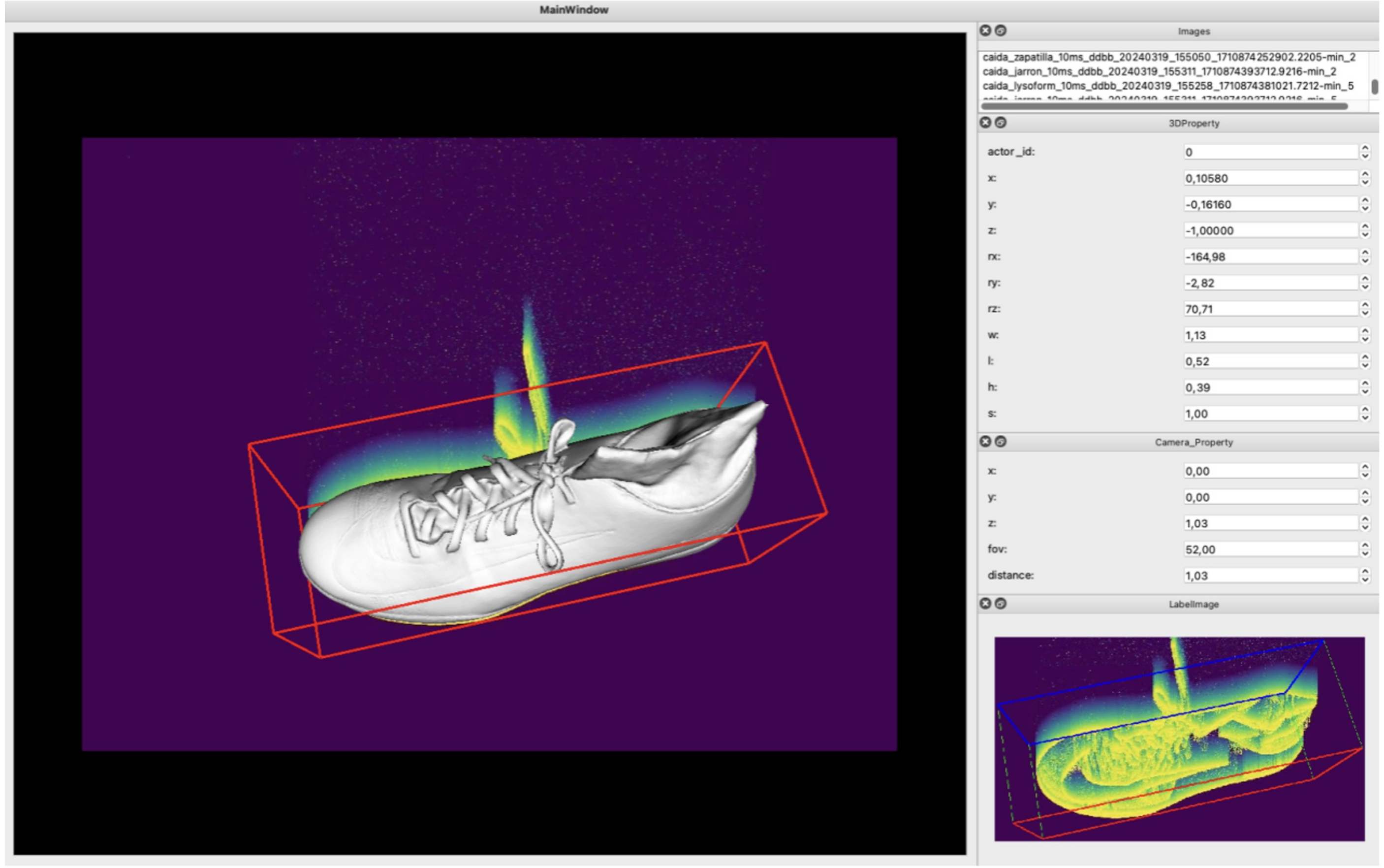
ZED-2

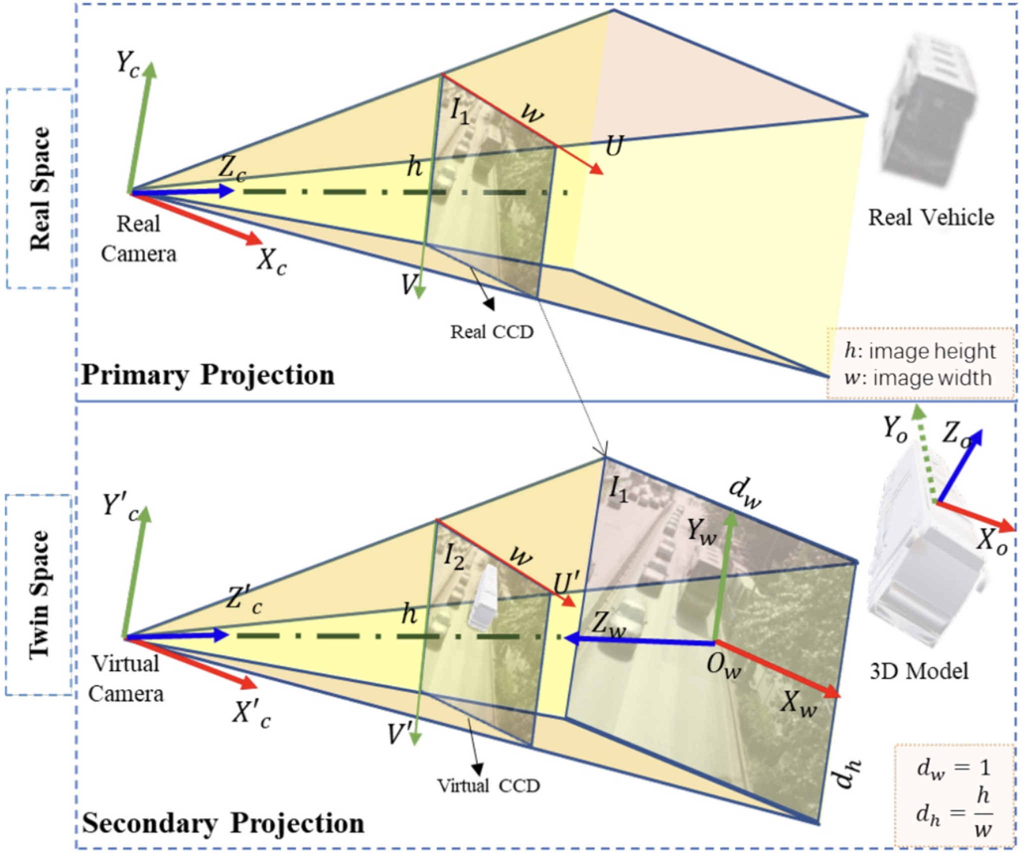
DAVIS346

Prophesee EVK4



Moving6DPoSe-R dataset annotations: 6D Pose





Moving6DPoSe-R dataset annotations: 6D Pose

(a) Frame-based 6D Pose (b) Event-based 6D Pose

Fig. 8: Moving6DPoSe-R - Frame and event-based 6D pose annotation of moving objects using LabelImg3D [29].

LACORO

Latin American Summer School on Robotics

9th - 13th December 2024
Rancagua, Chile

The Latin American Summer School on Robotics (LACORO) aims to make cutting-edge knowledge of Artificial Intelligence for Robotics Applications more accessible in the Southern Hemisphere. Moreover, we want to foster intercultural student collaboration within and outside the Americas.

Our aims are:

- to build a sustainable community of students, academics and professionals in Artificial Intelligence for Robotics, particularly Cognitive Inspired Aspects of AI.
- to foster intercultural student collaboration within and outside the Americas.
- to promote national and Latin American development in areas relevant to the region's economy, such as agriculture, manufacturing, mining, and retail.

LACORO

Latin American Summer School on Robotics

9th - 13th December 2024
Rancagua, Chile

Josh Pinskier
CSIRO Robotics, Australia

Javier Preciozzi
Universidad de la República & DigitalSense, Uruguay

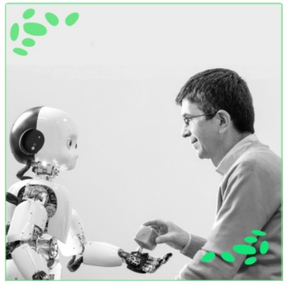
J. Matias Di Martino
Universidad Católica del Uruguay, Uruguay and Duke University, USA

Yinoussa Adagolodjo
Defrost team, Institut National de Recherche en Informatique et Automatique (INRIA)

Alice Smith
Department of Industrial and Systems Engineering, Auburn University, USA

Taihú Pire
CIFASIS, Rosario, Argentina

LECTURERS



Angelo Cangelosi
Department of Computer Science, University of Manchester (UK)

Josie Hughes
Institute of Mechanical Engineering, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Karinne Ramirez-Amaro
Department of Electrical Engineering, Chalmers University of Technology, Sweden

Miguel Torres Torriti
Departamento de Ingeniería Eléctrica, Pontificia Universidad Católica de Chile, Chile

Javier Ruíz Del Solar San Martín
Departamento de Ingeniería Eléctrica, Universidad de Chile, Chile

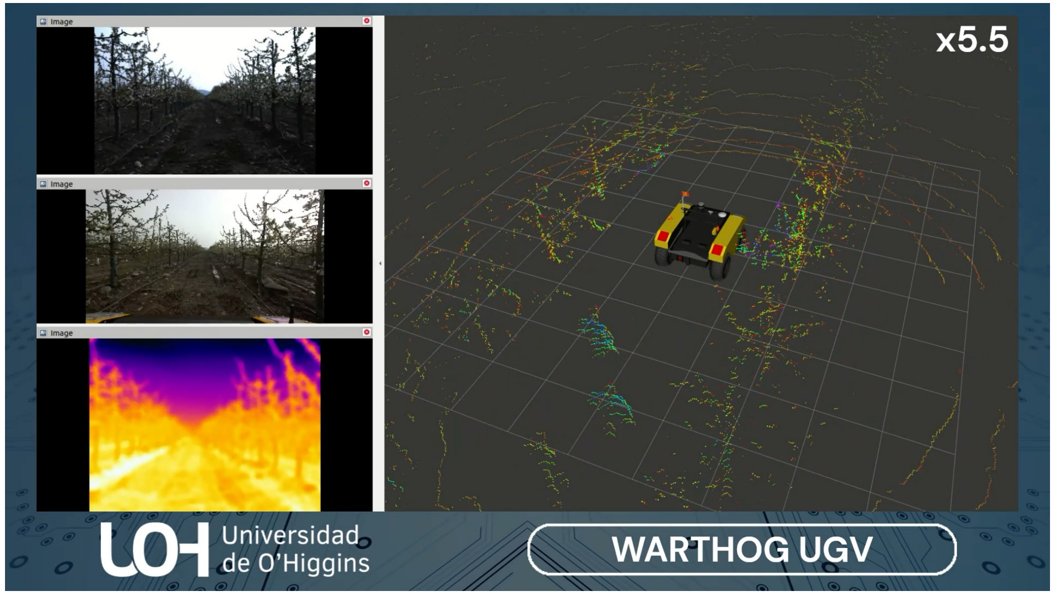
Karon MacLean
Department of Computer Science, University of British Columbia, Vancouver, Canada

Universidad de O'Higgins

Thank you for your attention

December 2022

December 2023



Rodrigo Verschae

Institute of Engineering Sciences,

Universidad de O'Higgins

Email: rodrigo@verschae.org

Web: rodrigo.verschae.org

Robotics and Intelligent Systems Laboratory (RIS LAB)
https://sites.google.com/uoh.cl/uoh-ris-lab