# Visual Inertial Odometry (VIO) based on Event Cameras

José Gomes

josepgomes@tecnico.ulisboa.pt

TÉCNICO LISBOA

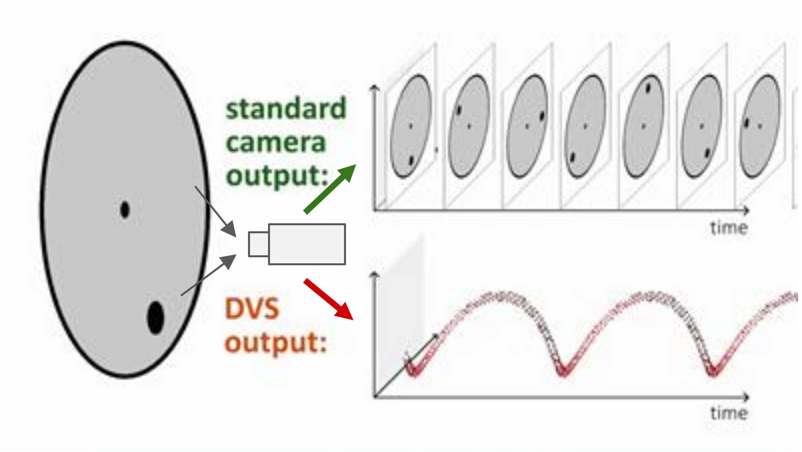
Institute for Systems and Robotics | LISBOA

LARSyS
Laboratory of Robotics and Engineering Systems

VIS LAB

# Event cameras

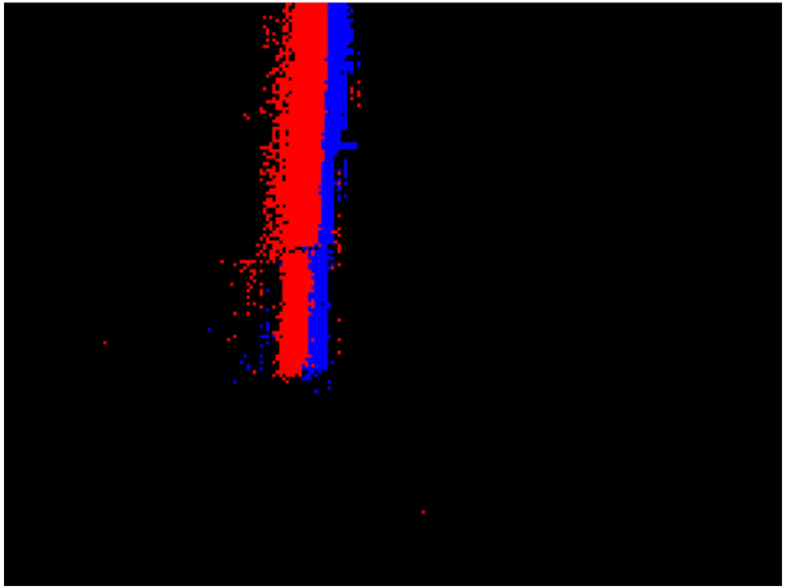
Based on the Dynamic Vision Sensor (DVS), respond to **changes in brightness** at each pixel.




| Camera models | |
|---|---|
| DVS128 | Events only |
| DVS240, DVS360 | Events and frames (one at a time) |
| DAVIS240, DAVIS360, newer models | Events and frames (simultaneously) |

*Note: all these models have embedded IMU*

# Event cameras, real data examples



Pen moving

Person waving

*Objective: use this data for the Visual Inertial Odometry system*

# Base tools for VIO using Event Cameras

## EKLT (feature tracker)
[Gehrig20]

Uses **frames** and **events** to track features

Features are tracked by comparing **image patches** (gradient) with **accumulations of events**.

## FUSION (pose estimator)
[Brossard17]

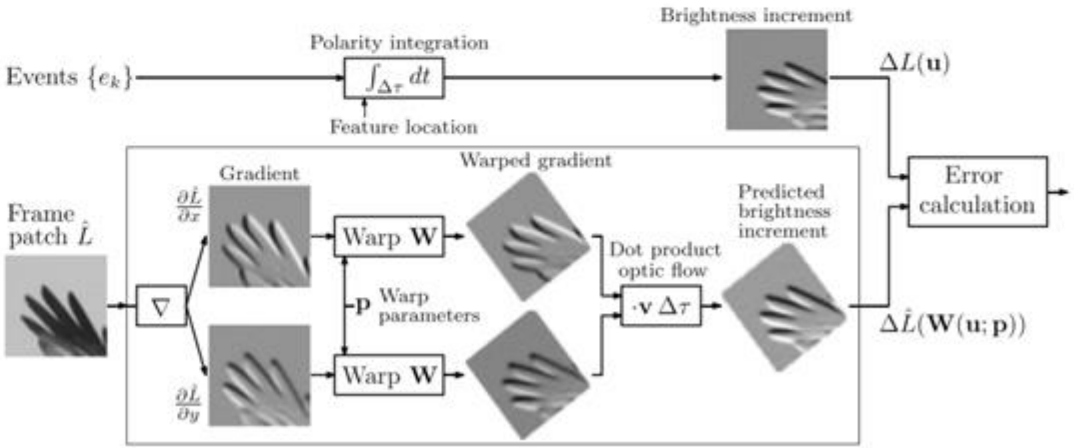
Combines **Visual** and **IMU** information for **pose estimation**.

Has a Lie group structure for the **UKF filter** encompassing pose, velocity and landmark estimation.

[Gehrig20] "EKLT: Asynchronous photometric feature tracking using events and frames", D. Gehrig, H. Rebecq, G. Gallego and D. Scaramuzza, International Journal of Computer Vision, 128(3), 601-618, 2020.

[Brossard17] Brossard, Martin, Silvere Bonnabel, and Axel Barrau. "Unscented Kalman filtering on Lie groups for fusion of IMU and monocular vision." Int. Conf. Robot. Automation (ICRA). 2017.

# Feature tracking EKLT



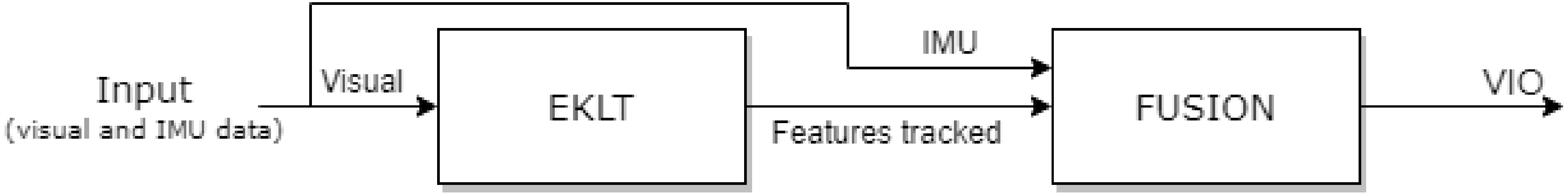

1) Identify features
2) Create patches around features
3) Match image template with event template
4) Update position estimate

$$\min_{\mathbf{p},\mathbf{v}} \left\| \frac{\Delta L(\mathbf{u})}{\|\Delta L(\mathbf{u})\|_{L^2(\mathcal{P})}} - \frac{\Delta \hat{L}(\mathbf{u};\mathbf{p},\mathbf{v})}{\|\Delta \hat{L}(\mathbf{u};\mathbf{p},\mathbf{v})\|_{L^2(\mathcal{P})}} \right\|^2_{L^2(\mathcal{P})}$$

p - warp parameters (position and rotation)
v - flow angle

*EKLT was designed for the DAVIS event camera*

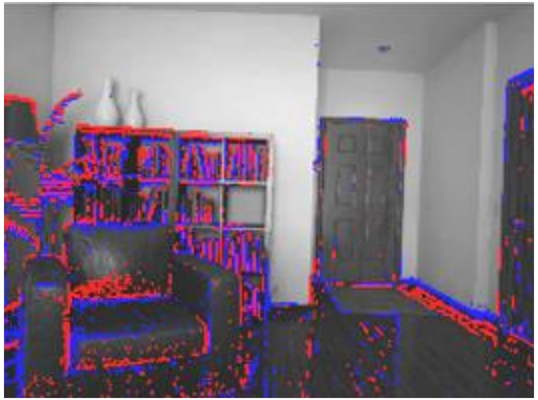
# Open loop integration




EKLT **tracks visual features**

FUSION receives **corresponded visual features** and combines with **IMU** to **estimate state** (camera pose and 3D landmarks).

# Open loop integration experiment: setup



*Simulated environment*

Experiment steps:

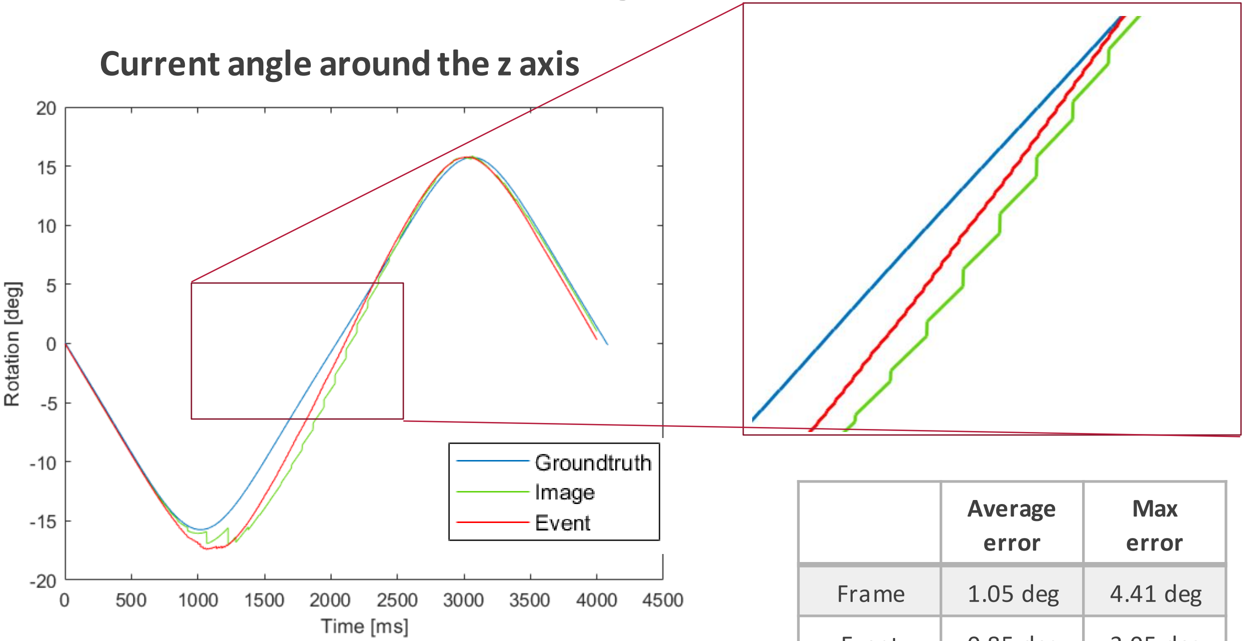
- **Render scenario** in Unreal Engine
- **Process with EKLT** the events saved in a ROS bag
- **Run FUSION** on tracked features

Simulating a DAVIS, the events camera for which the EKLT was developed.

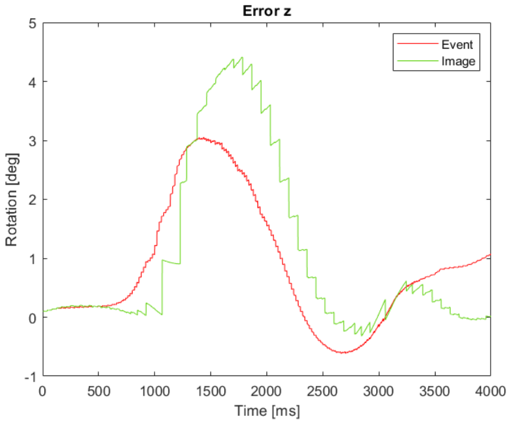
# Open loop integration experiment: compared results

## Motion: rotation on a single axis (z)

**Current angle around the z axis**




**Error in the estimated current angle**




| | Average error | Max error |
|---|---|---|
| Frame | 1.05 deg | 4.41 deg |
| Event | 0.85 deg | 3.05 deg |

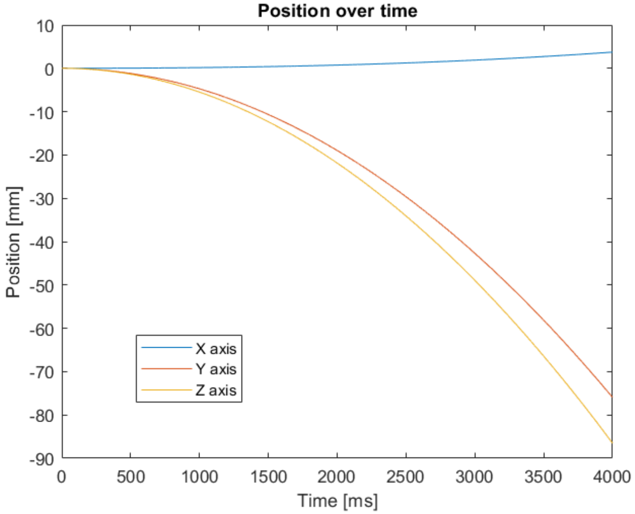
*Event camera visual data, added upon IMU data, improves performance and minimizes the average and absolute errors*

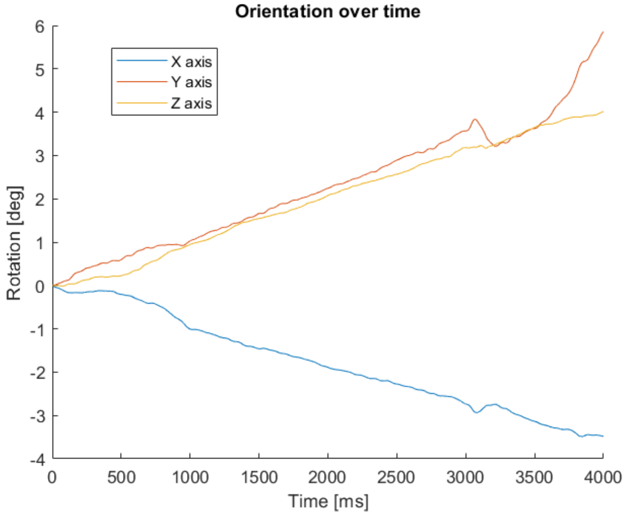
# Open loop integration experiment: Real world setup

**Visual tracking is lost early**

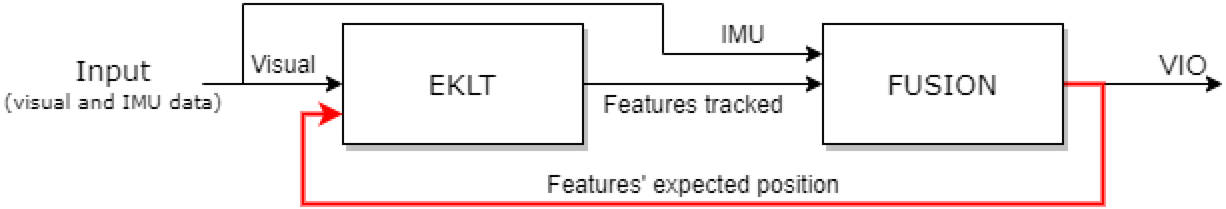
EKLT was designed for the DAVIS camera (frames and events always available).

We believe it can be adapted for the DVS by using the estimated ego-motion.



# Proposal: Closed Loop Integration of Sensor and Pose Filter




**Combining** EKLT with FUSION **in a closed loop integration** is expected to:

- Allow obtaining longer feature tracking with DVS cameras
- Find new features and track them despite periods without data
- Improve the estimation of camera position and orientation

LARSyS
Laboratory of Robotics
and Engineering Systems

Institute for Systems
and Robotics | LISBOA

VISLAB

Thank you for your attention

TÉCNICO
LISBOA

# Open loop integration experiment: Real world setup

Tracking **fails after a short while**

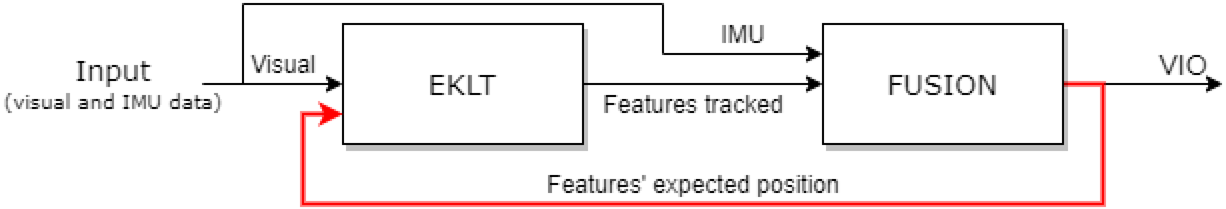
1) features get out of FOV
2) tracking quality degrades

**DVS camera** - long delay before new frame becomes available (and events are stopped - zero new data)

No simultaneous events and frames to replace features

Problem DVS vs DAVIS: Commutation between modes in the DVS takes too long for the tracking to restart. Sometimes artifacts occur.

# Proposal: Closed Loop Integration of Sensor and Pose Filter




**Combining** EKLT with FUSION **in a closed loop integration** is expected to:

- Allow obtaining longer feature tracking with DVS cameras
- Find new features and track them despite the long lack of events after acquiring a frame
- Improve the estimation of camera position and orientation