Visual Inertial Odometry (VIO) based on Event Cameras

José Gomes
josepgomes@tecnico.ulisboa.pt
Event cameras

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

<table>
<thead>
<tr>
<th>Camera models</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVS128</td>
<td>Events only</td>
</tr>
<tr>
<td>DVS240, DVS360</td>
<td>Events and frames (one at a time)</td>
</tr>
<tr>
<td>DAVIS240, DAVIS360, newer models</td>
<td>Events and frames (simultaneously)</td>
</tr>
</tbody>
</table>

*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)**


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

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

**FUSION (pose estimator)**


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

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

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

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

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).
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)

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

<table>
<thead>
<tr>
<th></th>
<th>Average error</th>
<th>Max error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame</td>
<td>1.05 deg</td>
<td>4.41 deg</td>
</tr>
<tr>
<td>Event</td>
<td>0.85 deg</td>
<td>3.05 deg</td>
</tr>
</tbody>
</table>

Error in the estimated current angle
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
Thank you for your attention
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