1 - Abstract

A new approach for reinforcement learning with event cameras.
- Operating directly on the event stream using EventNet[1] and our 3D IMU/IMU.
- No intermediate aggregation
- Continuous input observation stream and output action stream
- Wrapper code for simple application to any “Gymnasium style” RL environment

2 - Background - Event Cameras

- Event cameras are asynchronous visual sensors.
- Brightness changes cause immediate signals from the sensor, with no shutter based policing.
- Numerous advantages: lowpower, low bandwidth, high dynamic range, and low world-to-sensor latency
- Disadvantages: No images, unclear how to apply traditional computation vision tools.

3 - Background - Reinforcement Learning

- Long horizon strategic machine learning (e.g. game playing)
- No ground truth right or wrong answers
- Maximise reward function across game
- Traditionally iterative process
- Agent chooses actions based on current environmental state
- Environment executes actions and returns new state

4 - CERIL overview

- Modules asynchronously inserted into the rollout buffer
- Losses query the buffer and are computed on any entries found

5 - CERIL Details

5.1 - Continuous rollout generation

- Generic wrapper for OpenAI Gym environments
- Render environment every step
- EventCameraSimulator (ESM) turns discrete render into continuous-time event stream

5.2 - Continuous feature encoding

- Event Decay Neural Network (EDNN) on events
- Specialised spatio-temporal convolution
- CNN style spatial convolution kernel
- SNR style temporal decay (learned per neuron)
- Dense feature encoding from sparse events
- See our oral paper here at BMVC

5.3 - Losses

- Projection head loss: regularises vision system
- Requires that states are recoverable from features
- Continuous variant of Prioriseal Policy Optimisation
- Policy loss: integral of clipped advantage function
- Critic loss: integral of critic-reward disagreement
- Evaluated at discrete times based on control loop speed

6 - Evaluation

6.1 - Environments

- Pendulum: Apply torque to swing up & balance pole
- Dense Reward: vary vertically angle – torque use
- CartPole: Use linear actuated cart to balance inverted pole
- Keep-alive reward: +1 per step pole is upright

- Pong: Move a paddle up and down to deflect bouncing balls past an autonomous opponent paddle
- Sparse reward: +1 if agent wins a round, -1 if it loses

- MountainCar: Oscillate linear actuated cart to escape valley
- Terminates ASAP reward: -1 for every step in episode

6.2 - State-of-the-art comparison

<table>
<thead>
<tr>
<th>Approach</th>
<th>Input state format</th>
<th>Average reward</th>
<th>Keep-alive reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventNet</td>
<td>RGB + events</td>
<td>402.8</td>
<td>172.2</td>
</tr>
<tr>
<td>ESM</td>
<td>R + F + ESM</td>
<td>172.2</td>
<td>17.4</td>
</tr>
<tr>
<td>CERIL</td>
<td>ESM + ESM</td>
<td>402.8</td>
<td>172.2</td>
</tr>
</tbody>
</table>

Average rewards of different RL techniques

- Pendulum training curves
- CartPole training curves
- Pong training curves
- MountainCar training curves

CERIL performs very favourably compared to all other visual RL approaches
- Pong is challenging: long-term planning required vs short-term event aggregation
- Only CERIL can solve MountainCar

Funded by EPSRC grant “Reflexive robotics” (EP/S035761/1)