Introduction

Dienstag, 1. November 2022 12:02

- Convolutions (standard vision approach) not applicable to asynchronous data
    - Create synchronous approach by "leaking" information from past
        - Asynchronous (event-by-event) or Synchronous (batches) input, synchronous output (fixed rate of reconstruction)
        - Example: CNN accepting async event, processing events -> events/s = event throughput
            - Single-layer CNN about 1 million events/s (cameras have up to 10 million events/s)
        - Problem: event-rate above event throughput -> no real time possible
    - Proposed method: sync/async hybrid
        - Problem with async event-by-even: 1 output per input event
        - Idea: apply only correction factor to some previous convolution result, performed as multi-thread process -> does not hinder event-throughput
        - Computing of correction factor (fast) and convolution update (slow) decoupled
        - Small cost in precision

Reference Implementation

Dienstag, 1. November 2022 12:29

- Each event requires internal representation C* to be updated for kernelSize around event location $\boldsymbol{x}_k$ -> patch wise updates, out-of-sync (denoted by *)
- 3 components per update:
  - Integration of event polarity
  - Temporal decay
    - user-defined scene parameter α
    - Time between current and prev. update C*, stored in matrix T (last update time/pixel)
  - Convolution with kernel W
- Algorithm:
  1. $for\ (u,v) \in patch(x_i, y_i) do$
  2. $dt \leftarrow t_i - T(u,v);$
  3. $C^*(u,v) \leftarrow C^*(u,v)e^{-\alpha dt} + W(u,v)p_i s;$
  4. $T(u,v) \leftarrow t_i;$
  5. $e_i \leftarrow \{x_i, y_i, t_i, C^*(x_i, y_i)\};$

High-throughput implementation

Dienstag, 1. November 2022 12:29

- Produces async output $\tilde{e}_i(x_i, y_i, t_i, c_i)$, reduces per-event computation by decoupling main operations
- Output event gets convolution score from convolution image from a secondary process
- Event-throughput maximised, cause secondary process multi-threaded
- c_i is approximation, cause convolved image production delayed
    - Correction factor v added
- Event-throughput from $0.61 * 10^6 \frac{e}{s}$ to $9.77 * 10^6 \frac{e}{s}$ (significant increase)(using $kernelSize = 3$)
- Latency of the order of milliseconds on benchmark data set $\left(35 * 10^6 \frac{e}{s}\right)$ for high-throughput (seconds for reference)
    - Real-time possible, delay does not increase over time
- Average error from reference implementation < 0.5%
- Github: https://github.com/event-driven-robotics/high-throughput-convolutions
- Video: doi