

Event-based Robot Vision

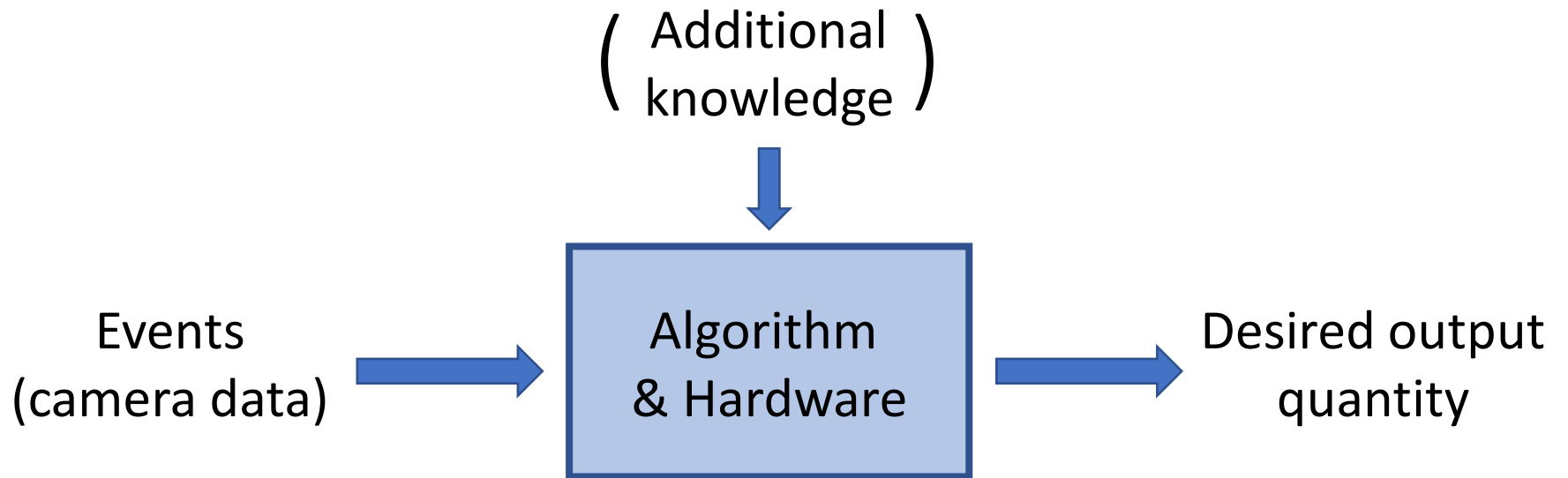
Prof. Dr. Guillermo Gallego
Chair: Robotic Interactive Perception

guillermo.gallego@tu-berlin.de

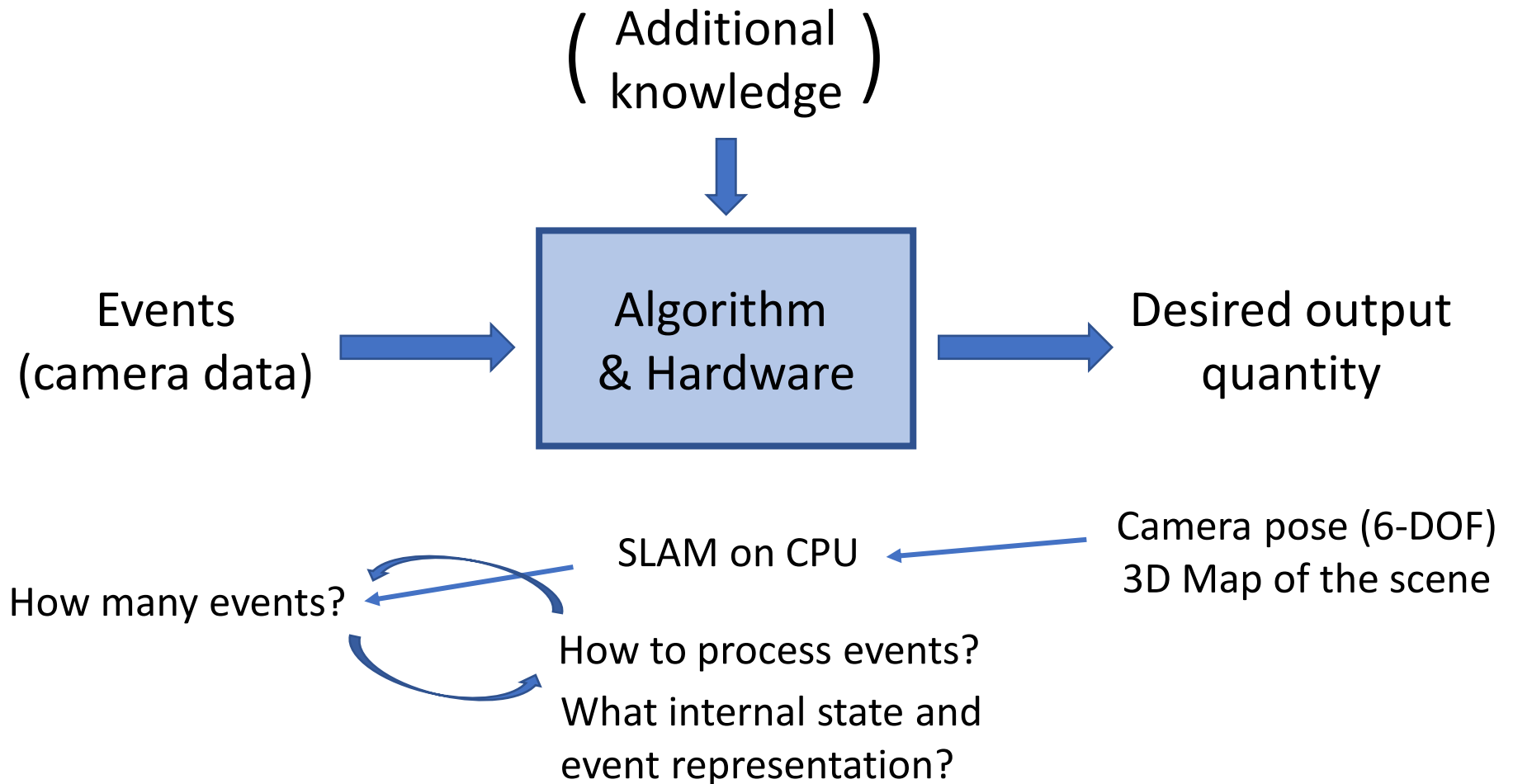
<http://www.guillermogallego.es>

How to Process the Events?

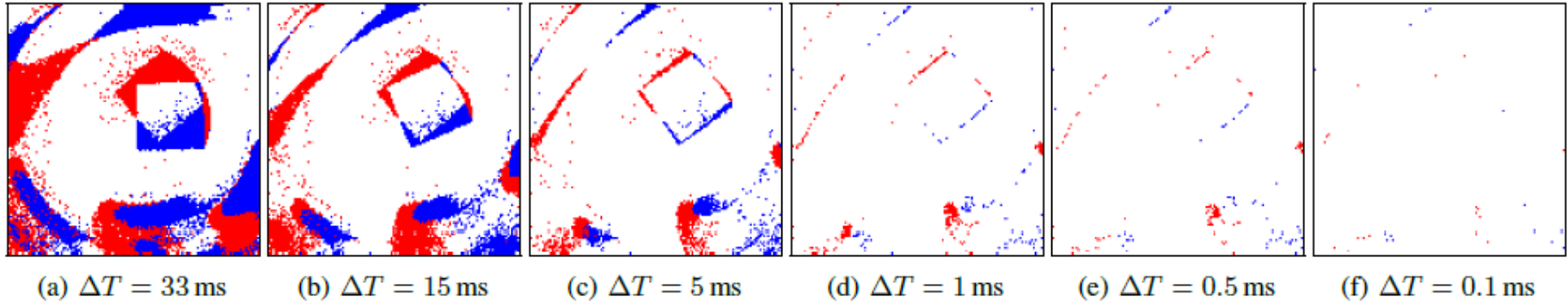
Overview



Overview



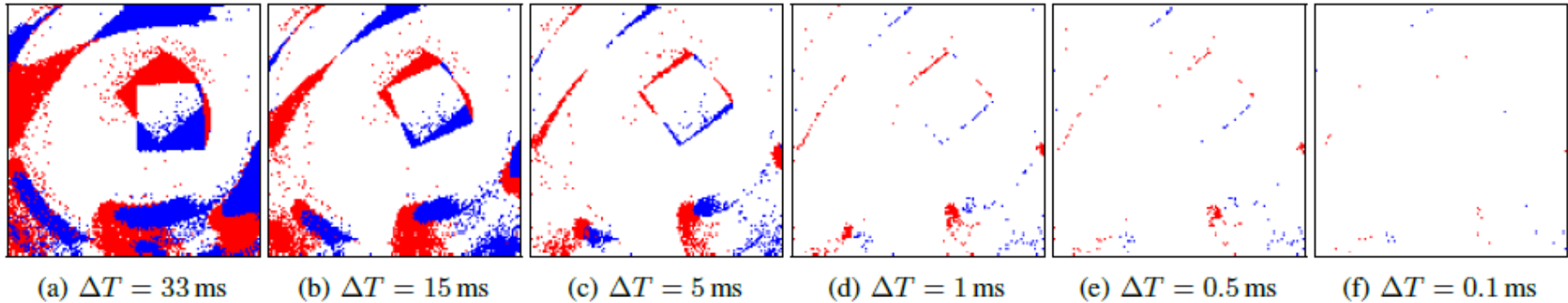
How many events should be used?



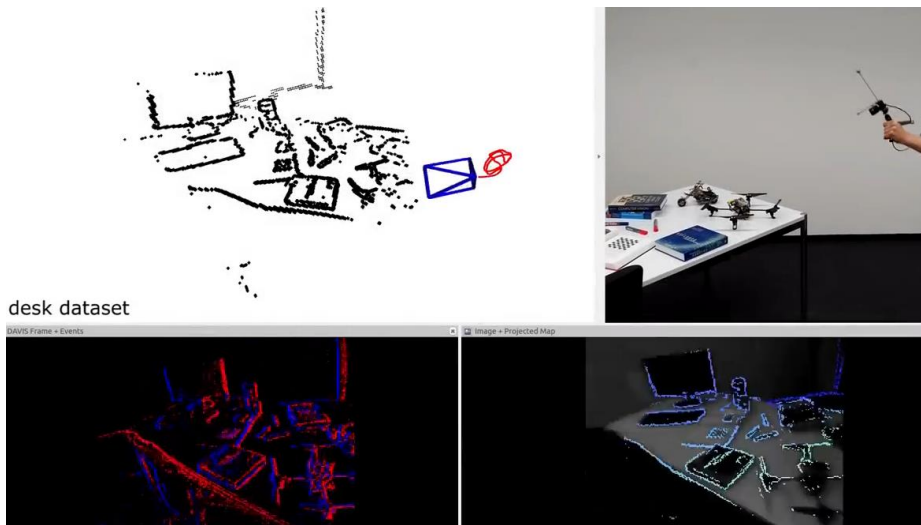
How can we estimate unknown parameters?

- **Event-by-event processing** (i.e., process **one event** a time)
 - **Pros**: low latency (in principle down to microseconds)
 - **Cons**: with high-speed motion \rightarrow millions of events per seconds \rightarrow GPU
- **Event-packet processing** (i.e., process the last **N events**)
 - **Pros**: N can be tuned to allow real-time performance on a CPU
 - **Cons**: no longer microsecond resolution (when is it needed?)

How many events should be used?



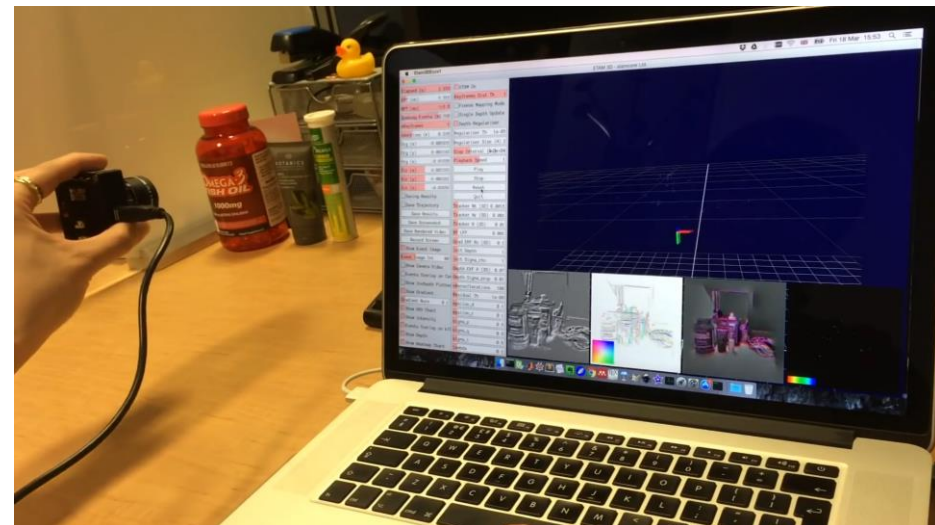
Event-packet



Rebecq et al., RA-L 2017 (CPU)

<https://youtu.be/bYqD2qZJlxE>

Event-by-event



Kim et al., ECCV 2016 (GPU)

<https://youtu.be/yHLYhdMSw7w>

Methods of Event Processing

Taxonomy of Methods

- **Event-by-event:**

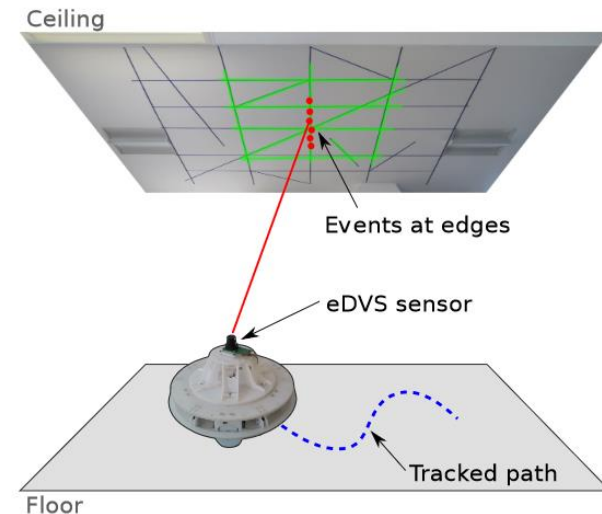
- Filters (deterministic or probabilistic – Bayes filter)
- ANNs - Spiking Neural Networks (SNNs)

- **Packets (groups) of events:**

- Optimization or variational methods
- Hand-crafted feature extractors
- Modern DNNs on grid-based representations of events

Event-by-event

- Process **one event at a time**
- How much information does an event carry?
- The vision system / algorithm needs to have **additional knowledge** (either from external information or built from past events) to assimilate (fuse with) each event.
- **Advantages:**
 - **Minimum latency**
 - Asynchronous & sparse
- **Disadvantages:**
 - Updating the system on a per-event basis may be expensive, depending on the task

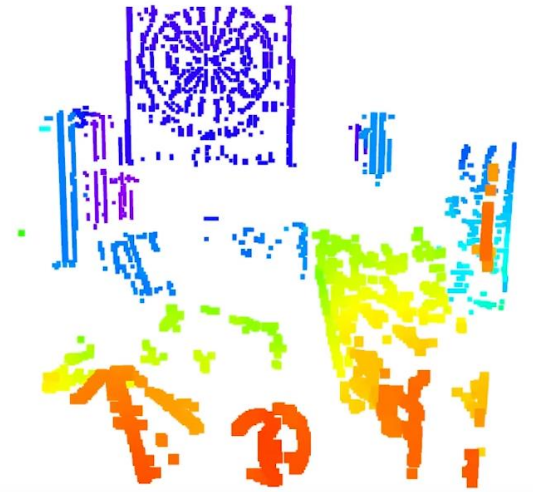


Event-by-event

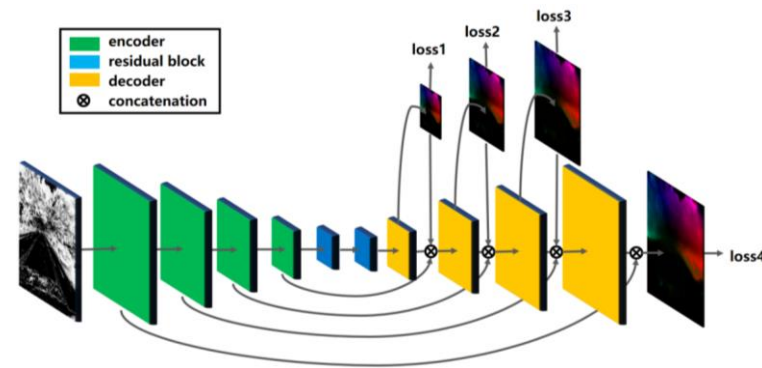
- Examples:
 - Background activity **filter** for event noise removal
 - Per-pixel **temporal filter** for image reconstruction (ACCV'18)
 - Parallel tracking and mapping using the **Bayes filter** (EKF and PFs): Weikersdorfer et al. ICVS 2013, Kim et al. BMVC 2014, Gallego et al. TPAMI 2018, etc.
 - **Multilayer ANNs** for object classification. Implemented in SNNs
- What's the **additional knowledge** on each system? (e.g., appearance information or scene “map” in SLAM)
- What's the inference **mechanism** used by each method to process the event and output a state update? (e.g., data fusion by EKFs, innovation)

Packets of events

- Process **events in a packet / group / window**
- Aggregate information from multiple events.
May not need additional knowledge.
- Events are processed differently depending on their representation. Specific methods for time surfaces, event frames, voxel grids, etc.
- **Advantages:**
 - Efficient event aggregation
 - May be asynchronous, with adaptive rate
 - Broader toolbox than event-by-event
- **Disadvantages:**
 - Introduce some latency



Packets of events



- Examples:
 - Conventional computer vision on **event frames or voxel grids**
 - **Classical** feature detection, extraction and tracking (e.g., Harris, KLT, etc)
 - **Modern** ANNs (CNNs, DNNs, encoder-decoder architectures, etc.)
 - Hierarchical feature extractors + shallow classifiers for **time surfaces**
 - Contrast / Focus maximization methods on **motion-compensated images**
- How is information in the event stream aggregated and filtered to produce a state update (the desired output)?
- How is additional knowledge, if any, considered?

References

Reading:

- Section 3 of Gallego et al., [Event-based Vision: A Survey](#), TPAMI 2020