EE5110/EE6110
Special Topics in Automation and Control
Segment B

From Frames to Events: Theory and Applications of Event-based Vision

Jiang Rui, Ph.D., BEng
rui.j@nus.edu.sg
Adjunct Lecturer
Department of Electrical and Computer Engineering

Semester 1, AY2021/2022

Course Structure

- **Lecture 1 – Event-based Vision System**
  - Conventional machine vision systems
  - Event-based sensors and cameras
  - Event data representation and compression
  - Event data simulation and dataset

Course Structure

- **Lecture 2 – Algorithms and Applications**
  - Detection and tracking
  - Optical flow estimation
  - Pose estimation and SLAM
  - Image reconstruction
  - Motion segmentation

Learning Objectives

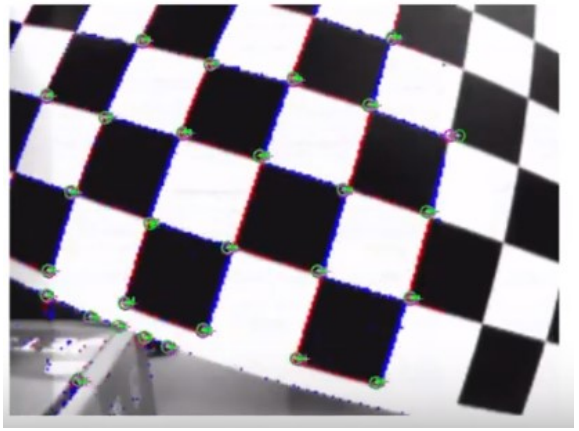
- In this lecture, you are expected to:

1. Describe and explain the basic concepts and methods in event-based computer vision algorithms and applications;

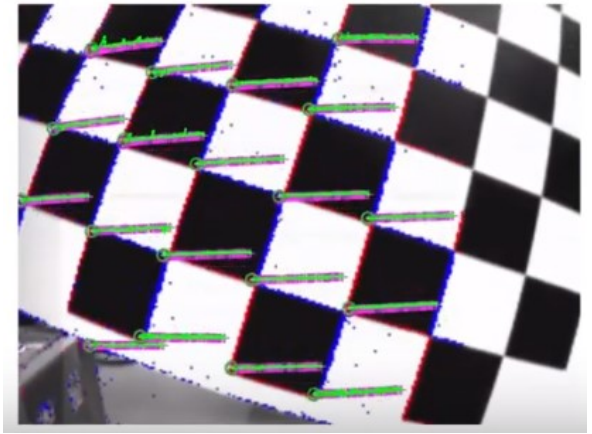
2. Understand the pros and cons by comparing the event-based methods with the frame-based counterparts.

Event-based Detection and Tracking

- Connection between detection and tracking
  - **Feature** detection:
    Extract those robust,
    representative events (“features”).
  - **Object** detection:
    Identify and find the interested objects.

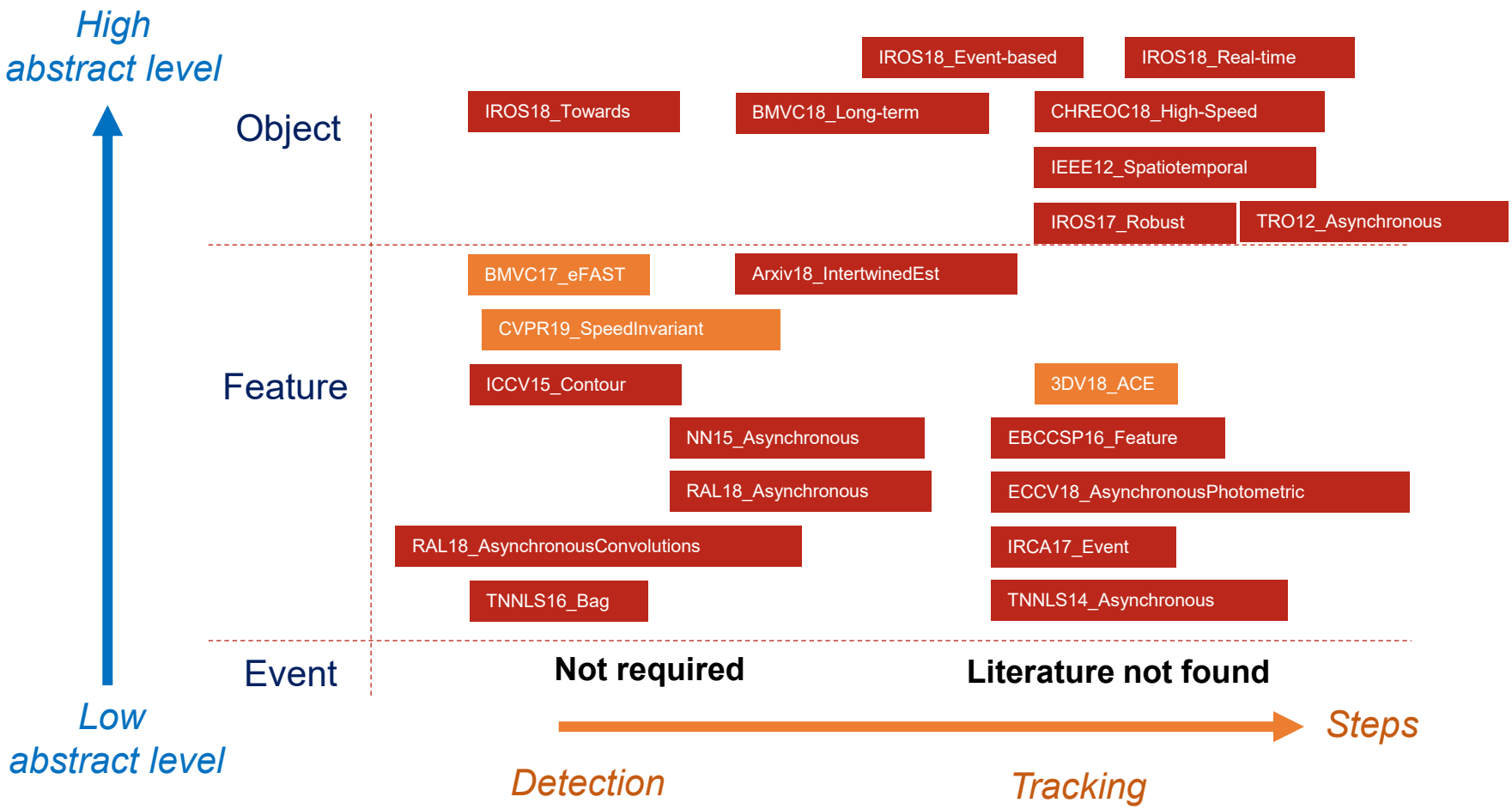


  - **Feature/object tracking:**
    Establish correspondences
    between features or objects.



Event-based Detection and Tracking

- Connection between detection and tracking



*Above references are for demonstration only, and do not reflect the recent progress of this field.

Event-based Detection and Tracking

- References' details

3DV18_ACE_An Efficient Asynchronous Corner Tracker for Event Cameras
Arxiv18_Event-Based Features Selection and Tracking from Intertwined Estimation of Velocity and Generative Contours
BMVC17_Fast Event-based Corner Detection
BMVC18_Long-term object tracking with a moving event camera
CHREOC18_High-Speed Object Tracking with Dynamic Vision Sensor
CVPR19_Speed Invariant Time Surface for Learning to Detect Corner Points with Event-Based Cameras
EBCCSP16_Feature detection and tracking with the dynamic and active-pixel vision sensor (DAVIS)
ECCV18_Asynchronous_Photometric Feature Tracking using Events and Frames
ICCV15_Contour Detection and Characterization for Asynchronous Event Sensors
IEEE12_Spatiotemporal Multiple Persons Tracking Using Dynamic Vision Sensor
IRCA17_Event-based Feature Tracking with Probabilistic Data Association
IROS17_Robust Visual Tracking with a Freely-moving Event Camera
IROS18_Event-based Moving Object Detection and Tracking
IROS18_Real-time clustering and multi-target tracking using event-based sensors
IROS18_Towards Event-Driven Object Detection with Off-the-Shelf Deep Learning
NN15_Asynchronous event-based corner detection and matching
RAL18_Asynchronous Corner Detection and Tracking for Event Cameras in Real Time
RAL18_Asynchronous Spatial Image Convolutions for Event Cameras
TNNLS14_Asynchronous Event-Based Multi-kernel Algorithm for High Speed Visual Features Tracking
TNNLS16_Bag of Events_An Efficient Probability-Based Feature Extraction Method for AER Image Sensors
TRO12_Asynchronous Event-based Visual Shape Tracking for Stable Haptic Feedback in Microrobotics

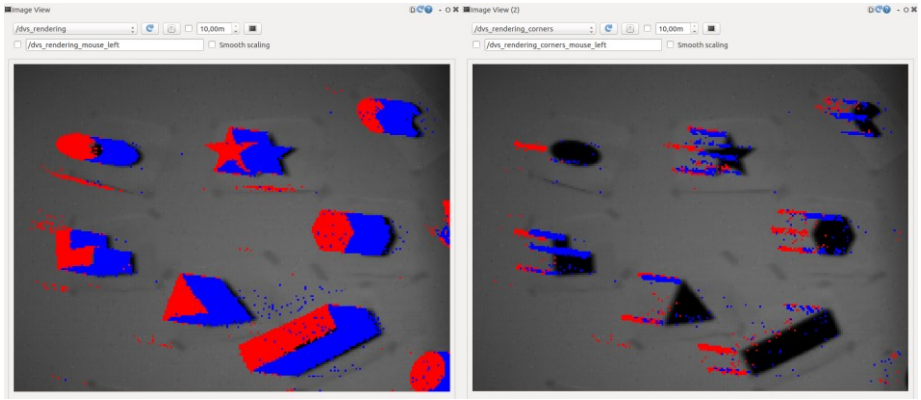
*Above references are for demonstration only, and do not reflect the recent progress of this field.

Paper: Fast Event-based Corner Detection

- "Event camera generates much less data than conventional camera. "

Is this always true?

- Not always! **Sometimes, more data from event camera…**
- That's why we need feature detection.
- Paper highlights:
  - A fast method for corner detection in an event stream
  - Efficient computing using only comparison operations



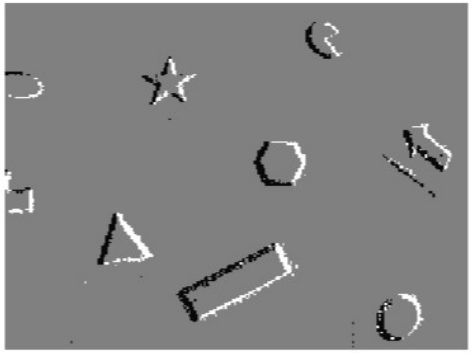
Reduce the event stream to a **corner event** stream!

PowerPoint Presentation (uzh.ch)

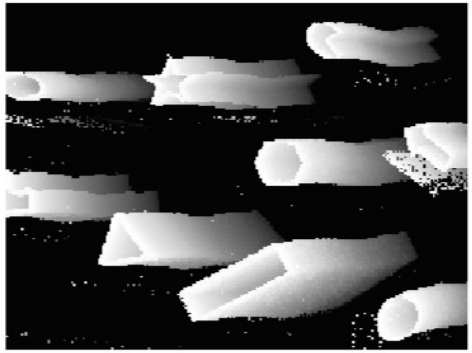
uzh-rpg/rpg_corner_events: Fast Event-based Corner Detection (github.com)

Paper: Fast Event-based Corner Detection

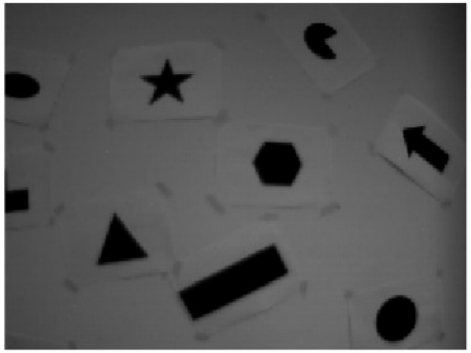
- Approach
  - Data representation: Surface of Active Events (a map with the timestamp of the latest event at each pixel)
  - Analyze timestamp distribution around current event
  - Detect corners by **searching for contiguous pixels with higher timestamps than the rest**



(a) Events over the last 33 ms. Positive events (white) and negative events (black)



(b) Surface of Active Events (SAE). Brightness represents time, from past (dark) to present (bright)



(c) Intensity Image (absolute brightness $I(x, y)$ in (1))

Figure 2: Signal used for corner detection: the Surface of Active Events (SAE).

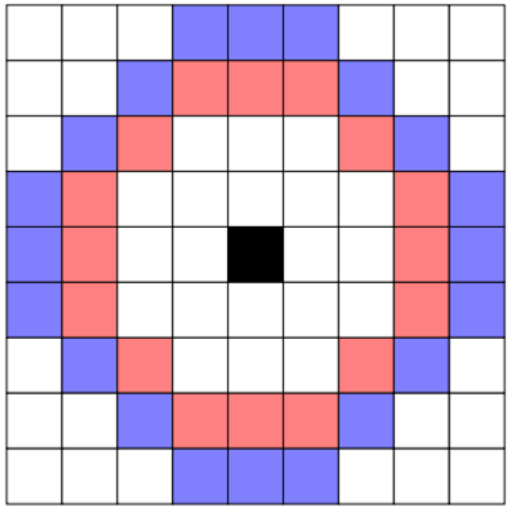
PowerPoint Presentation (uzh.ch)
uzh-rpg/rpg_corner_events: Fast Event-based Corner Detection (github.com)

Paper: Fast Event-based Corner Detection

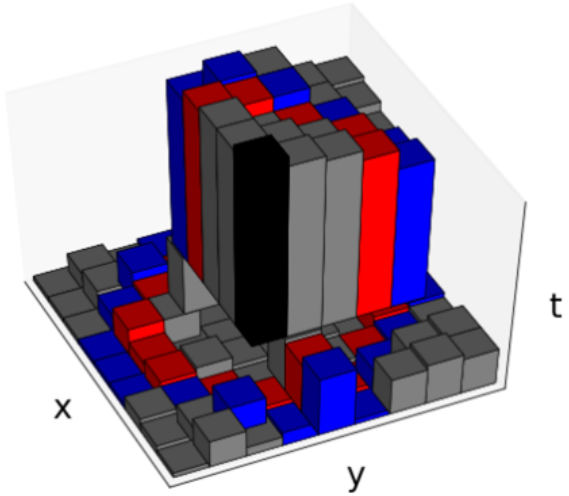
- Idea: The corner will create below local pattern. How to detect the pattern?
  - Along each circle, the algorithm searches for segments of contiguous pixels (arcs) that are higher (i.e., they have a more recent timestamp) than all other pixels on the circle.

Outer circle: search for a segment of length 4-8
Inner circle: search for a segment of length 3-6

} The center is a corner event!



(a) Circles of radius 3 and 4 pixels

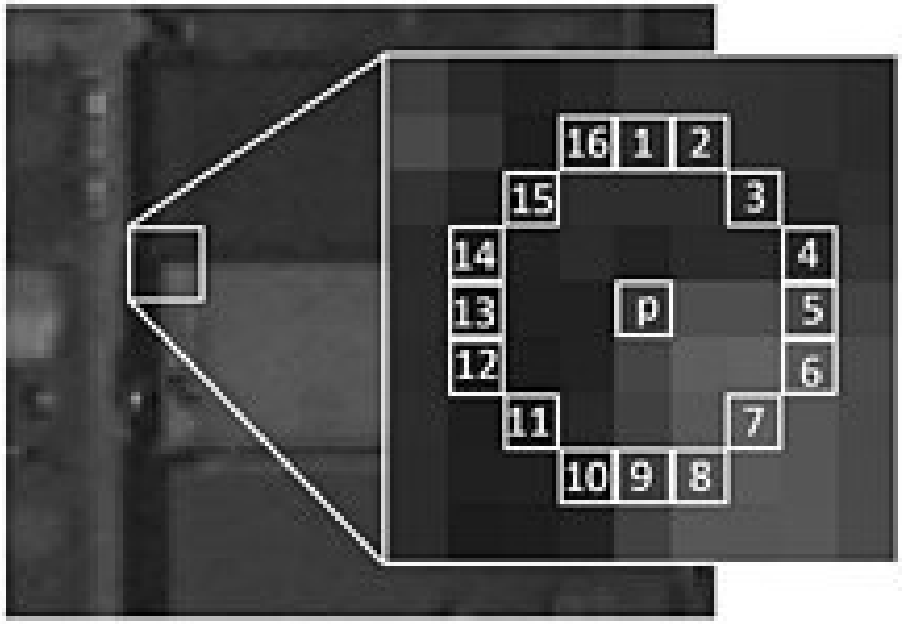


(b) Visualization of Surface of Active Events (SAE)

PowerPoint Presentation (uzh.ch)
uzh-rpg/rpg_corner_events: Fast Event-based Corner Detection (github.com)

Paper: Fast Event-based Corner Detection

- Idea similar to “FAST” feature in conventional computer vision.

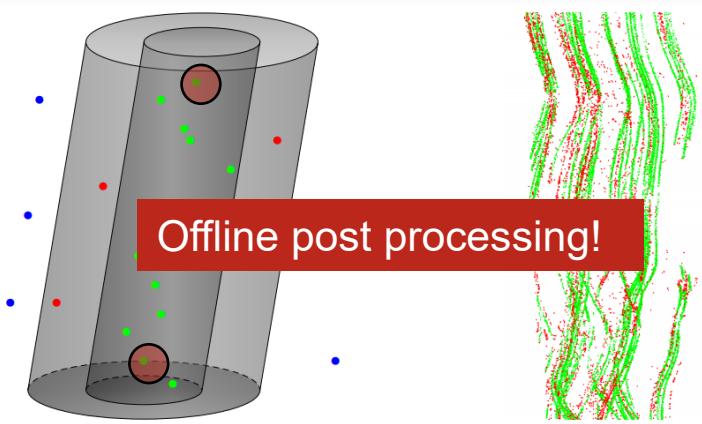


So it is also called “eFAST”!

Features from accelerated segment test - Wikipedia

Paper: Fast Event-based Corner Detection

- Find feature tracks
  - A feature track is composed of an inner and an outer oblique cylinder in space-time.
  - Exhaustively search for feature tracks by creating hypotheses using two corner events and checking whether there are **enough corner events in the inner cylinder** and **few corner events in the outer cylinder**.
  - Use 3 and 5 pixels for the inner and outer radius, respectively.
  - Minimum of 30 inner events, and maximum ratio of outer-to-inner events of 25%.



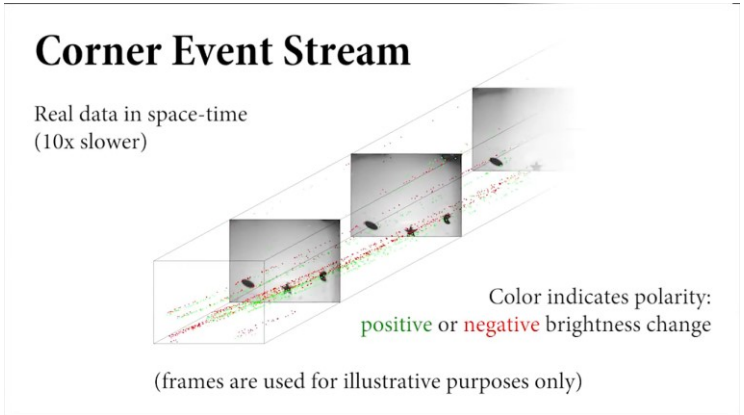
(a) Feature Track (b) Labeled Data

PowerPoint Presentation (uzh.ch)

uzh-rpg/rpg_corner_events: Fast Event-based Corner Detection (github.com)

Paper: Fast Event-based Corner Detection

- How fast/accurate is eFAST corner detection?



event reduction rate

success rate of corner tracking

Texture	Dataset	Harris [17] Red.	Harris [17] FT	Ours Red.	Ours FT
low	shapes_rotation	92.7	74.1	88.9	74.7
	shapes_translation	91.7	78.3	87.8	77.5
	shapes_6dof	90.6	77.1	87.0	76.8
medium	dynamic_rotation	95.1	53.3	96.4	46.4
	dynamic_translation	95.3	62.1	96.7	52.1
	dynamic_6dof	95.4	55.9	96.4	49.4
high	poster_rotation	92.6	35.3	95.7	30.5
	poster_translation	92.3	39.5	95.8	35.9
	poster_6dof	92.4	35.3	95.6	32.1
high	boxes_rotation	92.1	32.9	96.7	25.2
	boxes_translation	92.4	37.0	96.7	30.5
	boxes_6dof	92.7	34.4	96.8	26.7

Table 1: Performance of Harris and our detector expressed as reduction rate (Red.) of the event stream and matched Feature Tracks (FT). Values are given in percentages.

Method	Time per event [µs]	Max. event rate [e/s]
Harris [17]	11.6	86,230
Ours	0.78	1,275,000

Table 2: Runtime comparison per event and corresponding maximum event rate.

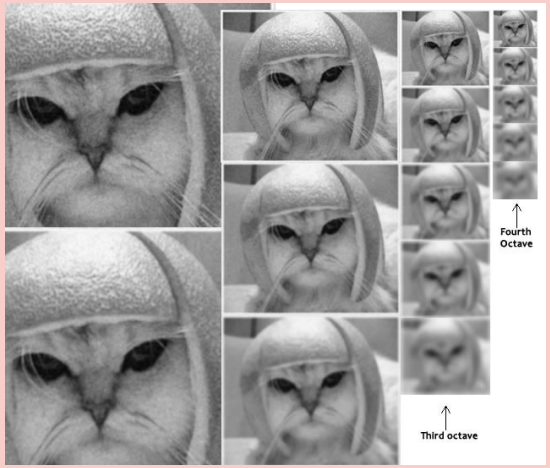
Fast event-based Harris corner detection exploiting the advantages of event-driven cameras | IEEE Conference Publication | IEEE Xplore
iros16.pdf (vvasco.github.io)

Recap: Feature Detection and Description

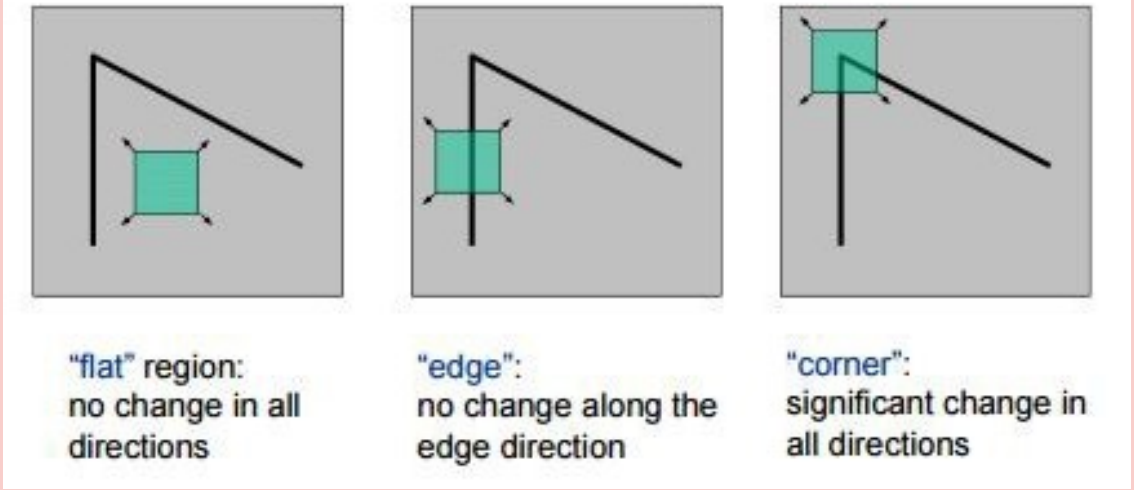
- In conventional computer vision, a feature usually comes with a **descriptor.**

Step 1: localize the feature

Keypoint localization → Orientation assignment → Keypoint description



Create a scale space to look for possible keypoints at different scales, and …

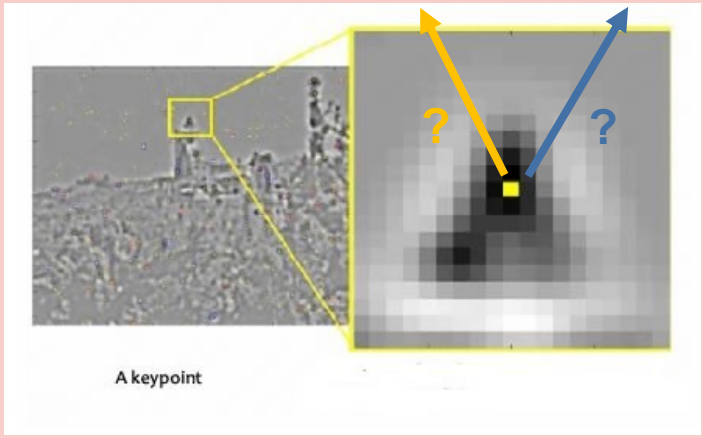


Check the gradients to select "corners" and reject "edges".

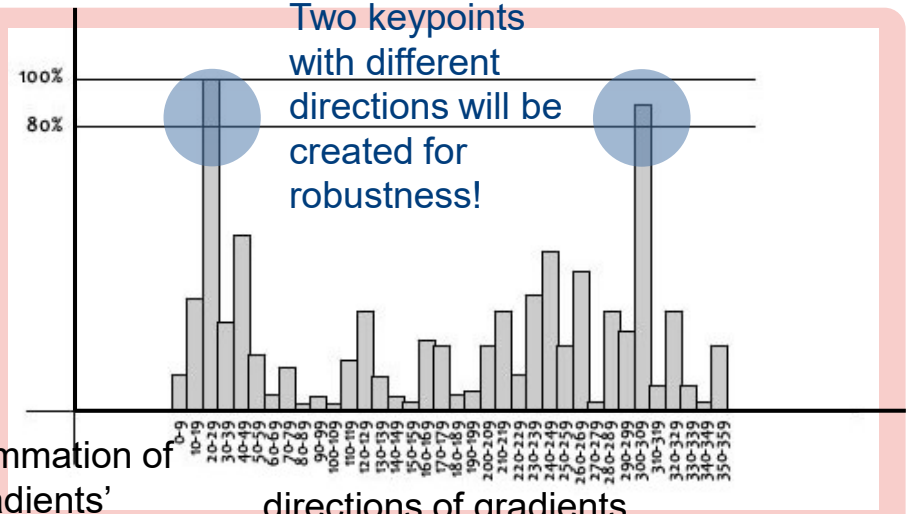
From feature descriptors to deep learning: 20 years of computer vision Eyes of Things
Introduction to SIFT(Scale Invariant Feature Transform) | by Deepanshu Tyagi | Data Breach | Medium

Recap: Feature Detection and Description

- **Step 2: find the orientation of the feature**



We want to find the orientation of the feature. Take a small patch near the feature point, and …



summation of gradients' magnitudes

directions of gradients

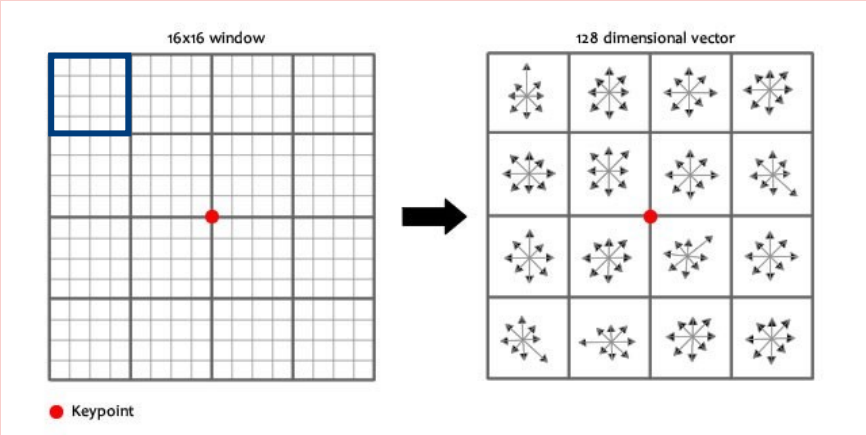
Compute gradients for all neighboring pixels, and create a histogram of gradient directions.

From feature descriptors to deep learning: 20 years of computer vision Eyes of Things
Introduction to SIFT(Scale Invariant Feature Transform) | by Deepanshu Tyagi | Data Breach | Medium

Recap: Feature Detection and Description

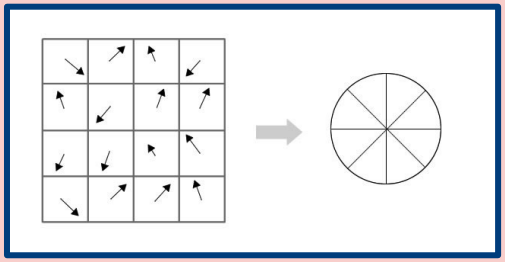
- **Step 3: compute the descriptor of the feature**

Keypoint localization → Orientation assignment → Keypoint description

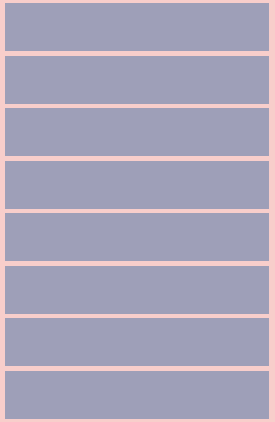


A 16x16 window around the keypoint is taken.
It is divided into 16 sub-blocks of 4x4 size.

For each sub-block, 8 bin orientation histogram is created.



feature vector



We will have 128 bin values to form a **128-dimensional feature vector.**

From feature descriptors to deep learning: 20 years of computer vision Eyes of Things
Introduction to SIFT(Scale Invariant Feature Transform) | by Deepanshu Tyagi | Data Breach | Medium

Recap: Feature Detection and Description

- We have just discussed **SIFT (Scale Invariant Feature Transform)**
  - Scale invariance:
    In the 1st step, we extract feature candidates in different scales.
  - Rotation invariance:
    In the 2nd step, direction of the feature is obtained, so that the descriptor contains relative rotation angle, which is compared to the direction of the feature, not the direction of the image.
  - Translation invariance:
    Obvious.
  - Illumination invariance:
    Normalization used to reduce the influence from lighting conditions and illuminance.

From feature descriptors to deep learning: 20 years of computer vision Eyes of Things
Introduction to SIFT(Scale Invariant Feature Transform) | by Deepanshu Tyagi | Data Breach | Medium

Paper: DART: Distribution Aware Retinal Transform for Event-based Cameras

- Highlights
  - Introduced a generic **visual descriptor** "DART", that encodes the structural context using log-polar grids for event cameras
  - DART is **robust to scale, rotation, and view-point variations**.

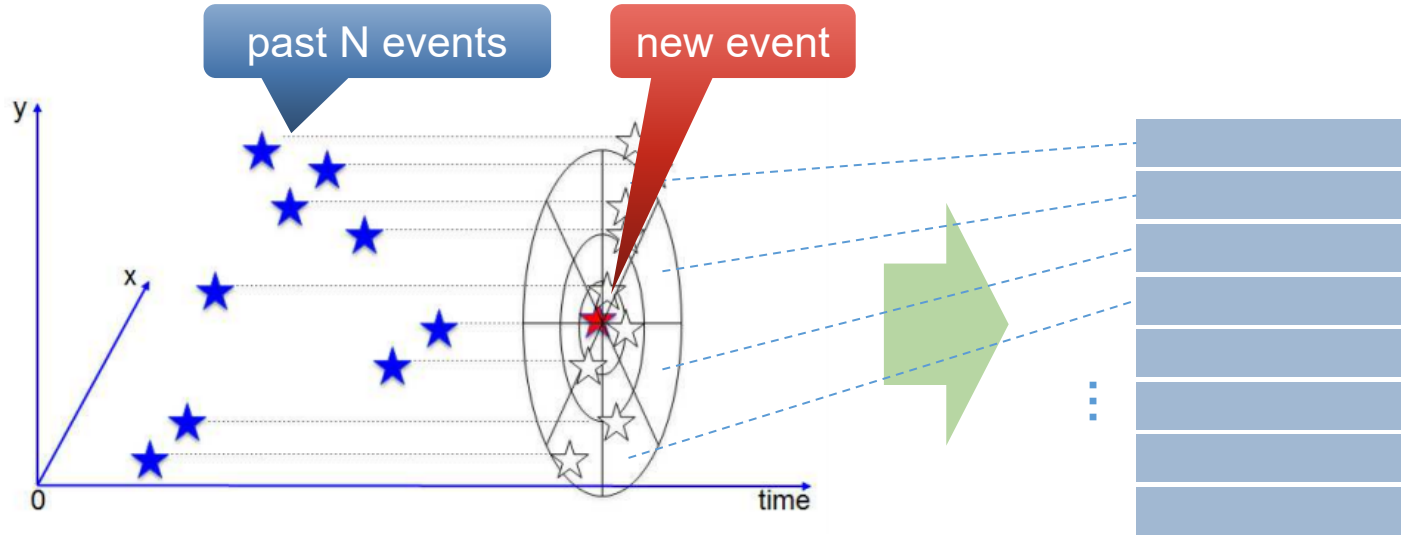
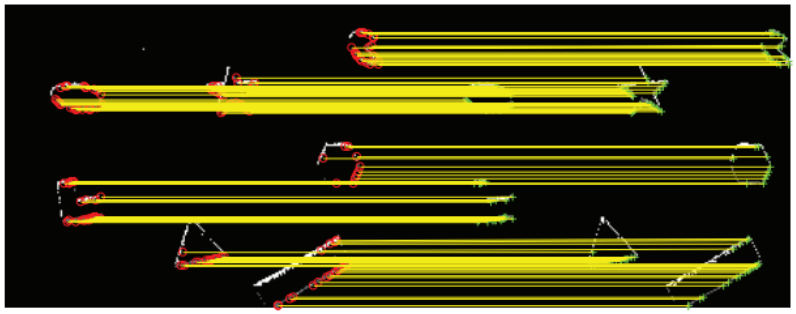


Fig. 1. The most recent event is indicated in red and the previous events are indicated in blue. White stars indicate the position of previous events mapped onto the log-polar bins at the current time.

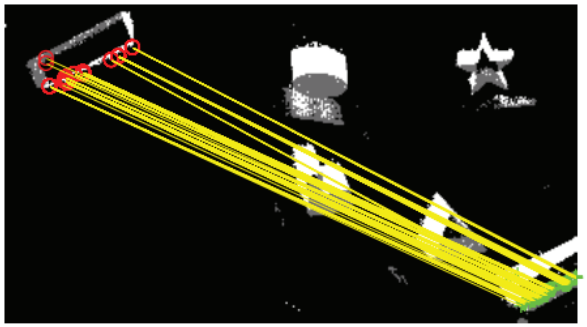
1710.10800.pdf (arxiv.org)

Paper: DART: Distribution Aware Retinal Transform for Event-based Cameras

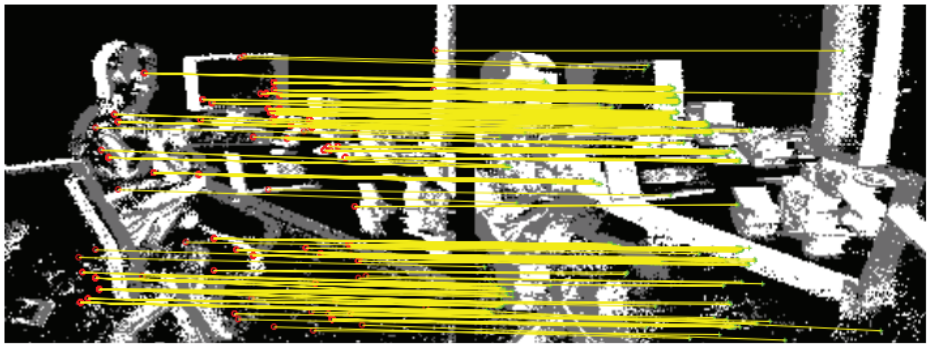
- Feature matching results



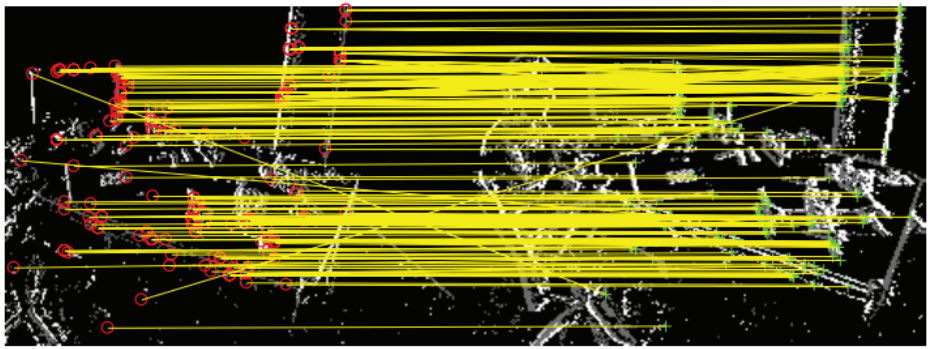
(a) Shapes data of temporally close sequences.



(b) Shapes data of temporally far sequences with occlusion.



(c) Indoor scene with different camera motion speeds.



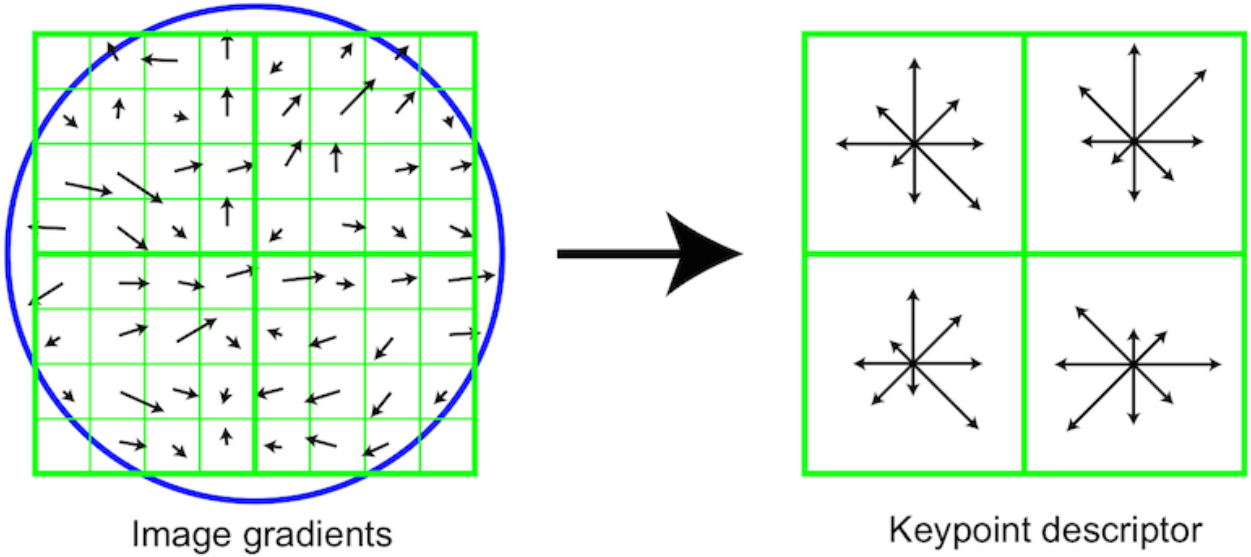
(d) Indoor scene with scale and rotation changes.

Red circles indicate the features in the left time-slice and the green crosses indicate the corresponding match in the right time-slice connected by yellow lines.

1710.10800.pdf (arxiv.org)

History and Future for Feature Detection and Description

- **Manually-crafted** features/descriptors v.s. **Learned** features/descriptors
  - **The rise of the local feature descriptors: ~1995 to ~2000**
    e.g., SIFT (1999, David Lowe @ University of British Columbia)



From comparing raw data (intensities) to comparing features!

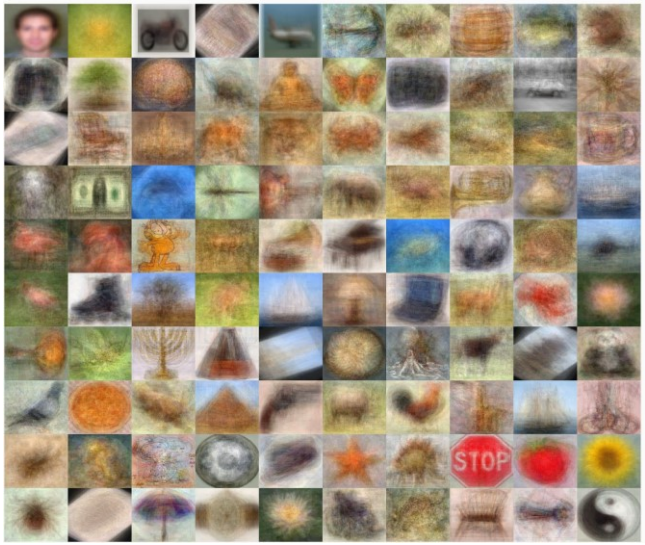
From feature descriptors to deep learning: 20 years of computer vision Eyes of Things
Introduction to SIFT(Scale Invariant Feature Transform) | by Deepanshu Tyagi | Data Breach | Medium

History and Future for Feature Detection and Description

- **The modern dataset: ~2000 to ~2005**

  Geometric problems (hmography estimation, ground-plane estimation, robotic vision, SfM) greatly benefited from robust image features such as SIFT.

  Images were going online. Datasets were being created.
  e.g., Caltech-101 (2003, Fei-Fei Li, et al. @ Caltech)



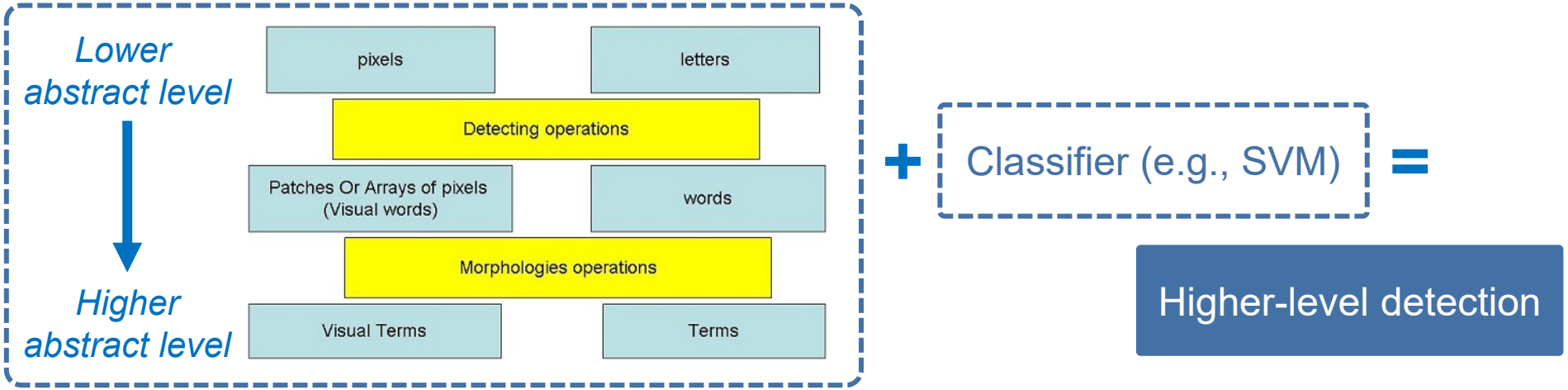
From feature descriptors to deep learning: 20 years of computer vision Eyes of Things
Introduction to SIFT(Scale Invariant Feature Transform) | by Deepanshu Tyagi | Data Breach | Medium

History and Future for Feature Detection and Description

- **Bins, Grids, and Visual Words (aka Machine Learning meets descriptors): ~2000 to ~2005**
  Raw SIFT: not powerful enough to provide object-level metric for matching

Statistical way: To combine more features and descriptors and compare the **feature set** instead of a single feature!

- e.g., Visual words: analogy text - image



From feature descriptors to deep learning: 20 years of computer vision Eyes of Things
Visual Word - Wikipedia

History and Future for Feature Detection and Description

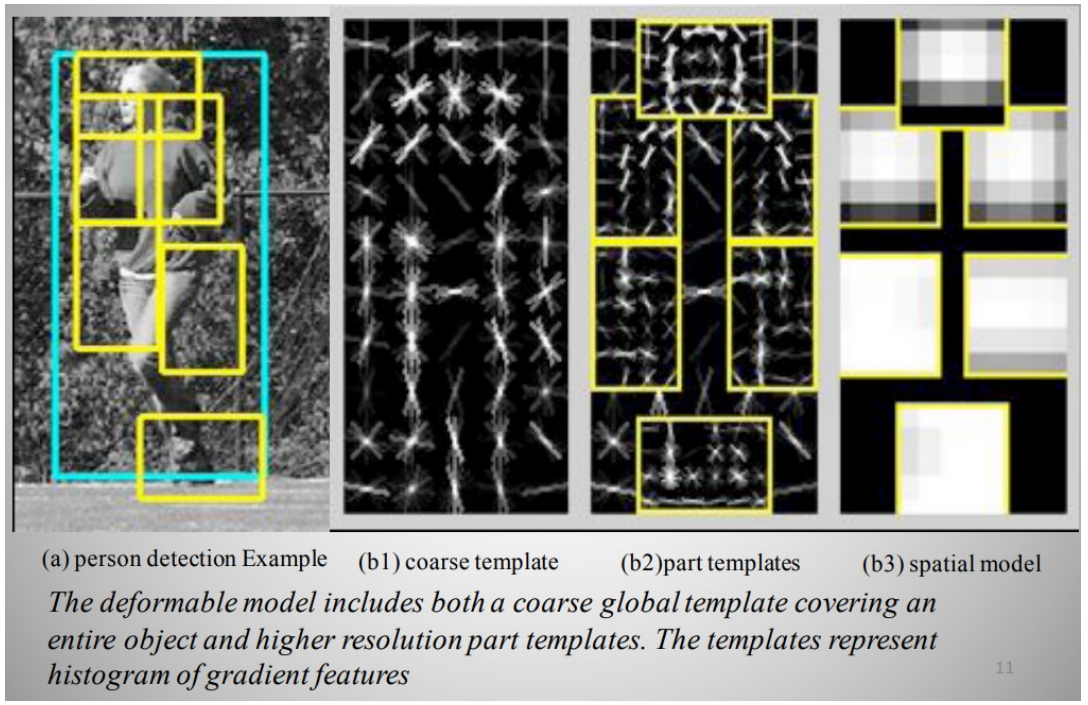
- **Object Templates (aka the reign of HOG and DPM): ~2005 to ~2010**

HOG: Histogram of oriented gradients

describe local object appearance and shape by the **distribution of intensity gradients** or edge directions.

Based on HOG, later we have DPM: **Deformable Part Model.**

Idea: To separate object into parts and make them movable!



Lec21-Deformable Part Models (ucf.edu)

History and Future for Feature Detection and Description

- **Big data, Convolutional Neural Networks and the promise of Deep Learning: ~2010 to ~2015 (until present)**

Bigger and bigger data!

+

Learning algorithms (since 1980s)

+

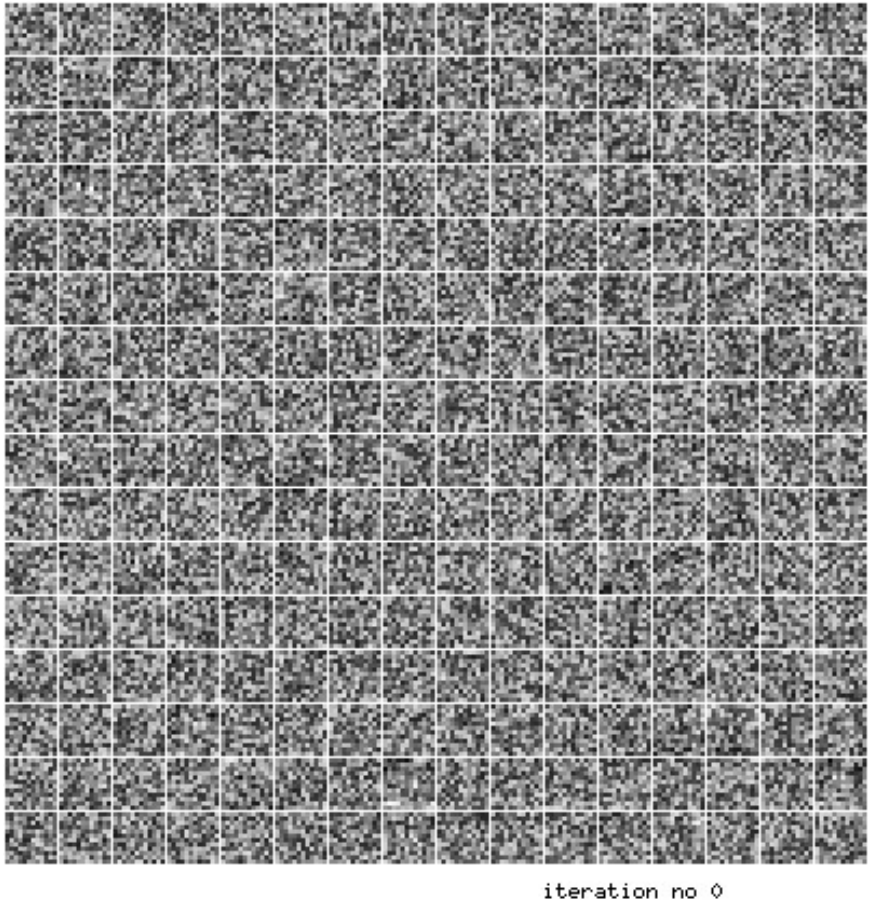
Hardware (GPU)

= ?

Simplified feature engineering, better performance…

What about for event-based vision?

Difficulties exist due to fundamentally different data type compared to conventional cameras!



"CBLL, Research Projects, Computational and Biological Learning Lab, Courant Institute, NYU"

Paper: Object Tracking on Event Cameras with Offline-Online Learning

- We have just discussed feature detection.
  What about object-level detection and tracking?

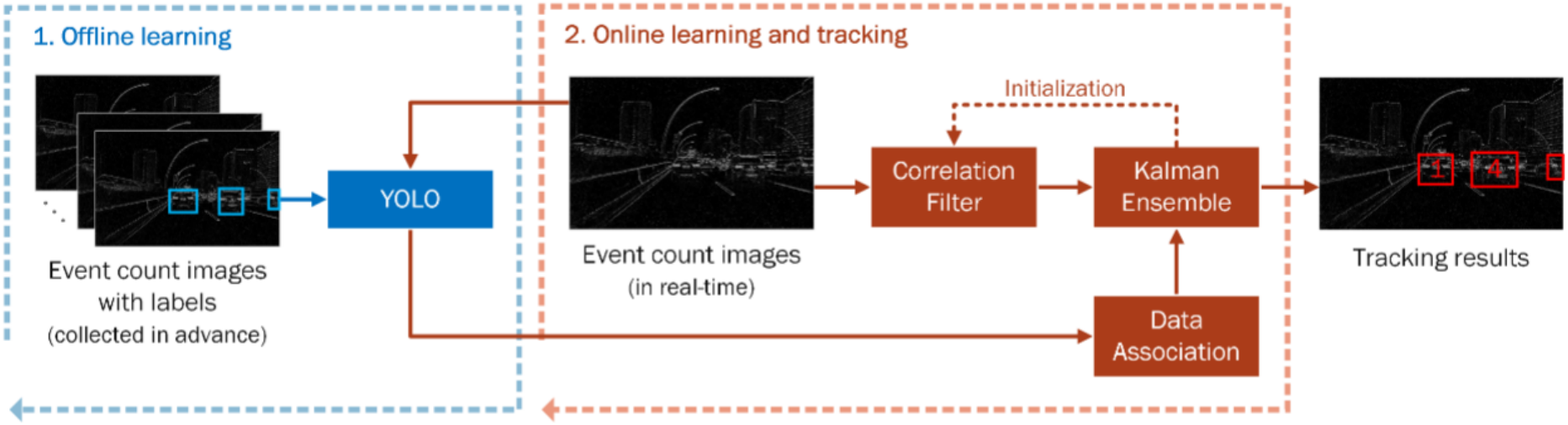
- Possible ways for object detection/tracking based on event cameras
  - **Detect and track objects on event frames** – straightforward and easy
  - Detect and track object in point clouds

- Highlights
  - Contains YOLO-Based offline-trained deep detector + Correlation Filter (CF)-based online-trained tracker, which complements each other;
  - A real-time fusion scheme designed based on the Kalman filter;
  - It is verified by experiments that: event count images contains enough information for object detection and tracking.

(PDF) Object Tracking on Event Cameras with Offline-Online Learning (researchgate.net)

Paper: Object Tracking on Event Cameras with Offline-Online Learning

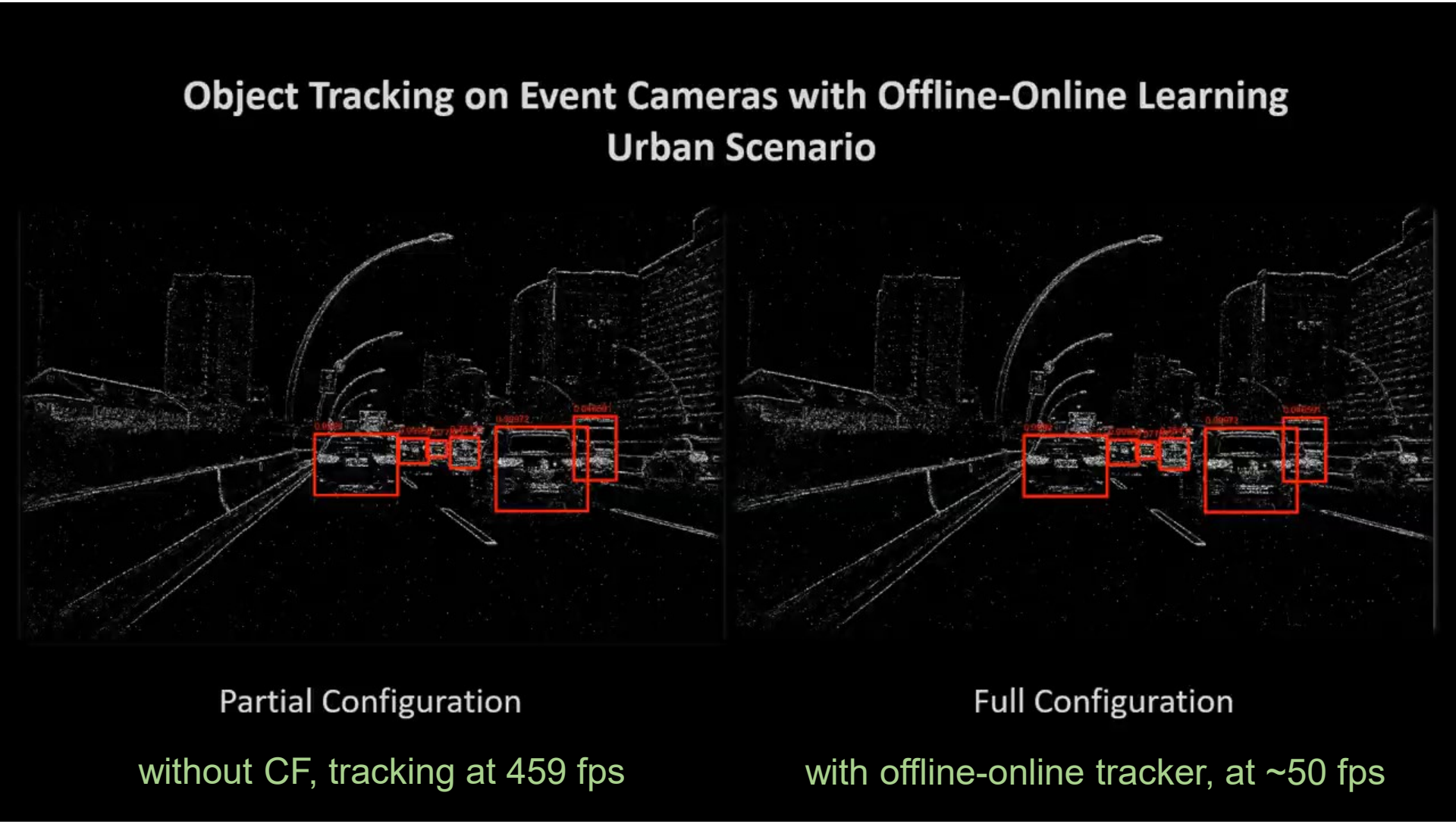
- Object tracking framework



- The 1st stage needs to be completed off-line before tracking such that a YOLO detector is trained;
- The correlation filter, which tracks single object, is initialized/reinitialized using results from the YOLO detector.
- The Kalman filter fuses measurements from both the detector and the tracker to achieve multi-object tracking.

(PDF) Object Tracking on Event Cameras with Offline-Online Learning (researchgate.net)

Paper: Object Tracking on Event Cameras with Offline-Online Learning



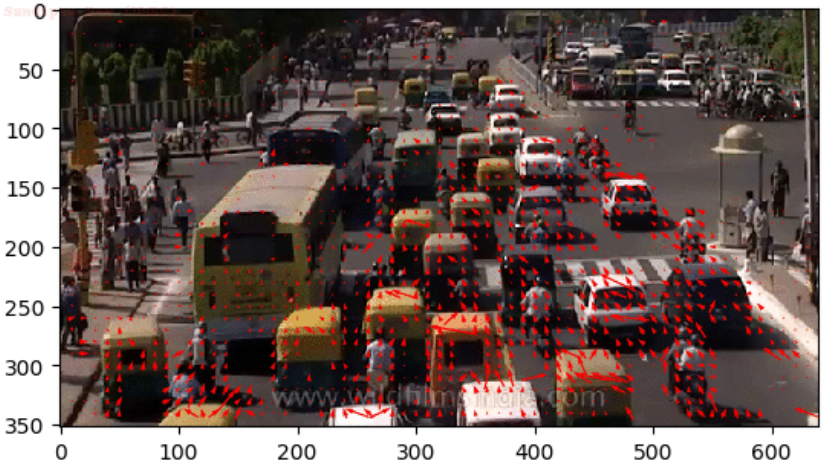
(PDF) Object Tracking on Event Cameras with Offline-Online Learning (researchgate.net)

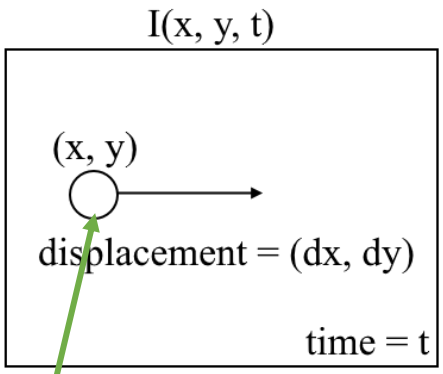
Summary – Detection and tracking

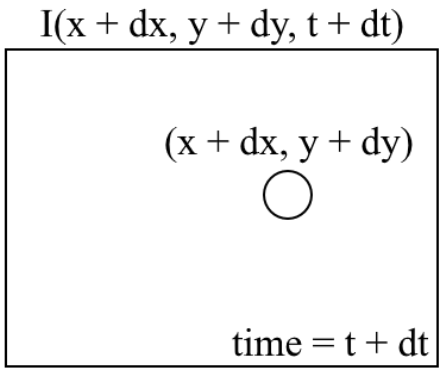
- Relation between detection and tracking
- Different abstraction levels of detection and tracking:
  - Event level → feature level → object level
- Why we need event-based feature detection?
  - To reduce the data rate, aiming at fast event data processing
- Feature detection and description
  - Descriptors required to “identify” features
  - From hand crafted to learned features
- Object-level detection and tracking
- Examples and papers

Optical Flow Estimation

- **Optical flow field** vs. **motion field**
  - Motion field reflects **object's real motion** in 3-D space
  - Optical flow field is the **distribution of apparent velocities** of movement of **brightness pattern** in an image (2-D space)





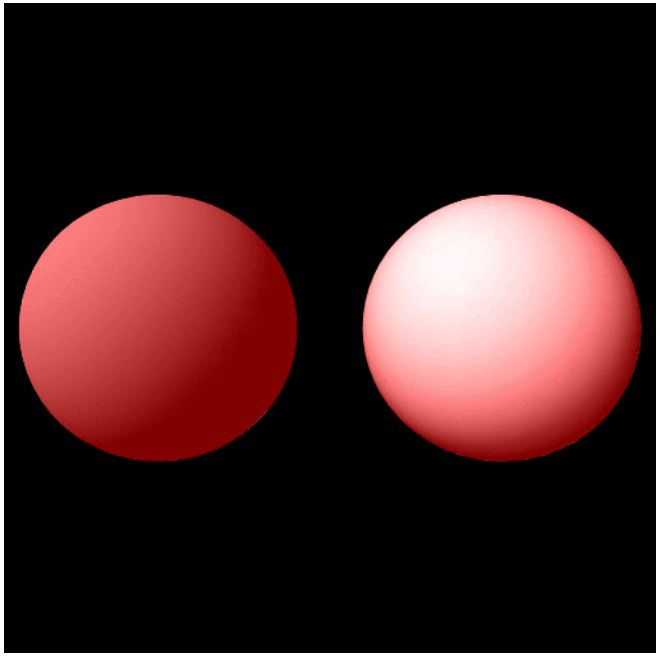


"brightness pattern"

Introduction to Motion Estimation with Optical Flow (nanonets.com)

Optical Flow Estimation

- Does the following scenario exist?
  - There is no motion field or optical flow;
  - There is both motion field and optical flow;
  - There is motion field but no optical flow;
  - There is optical flow but no motion field.



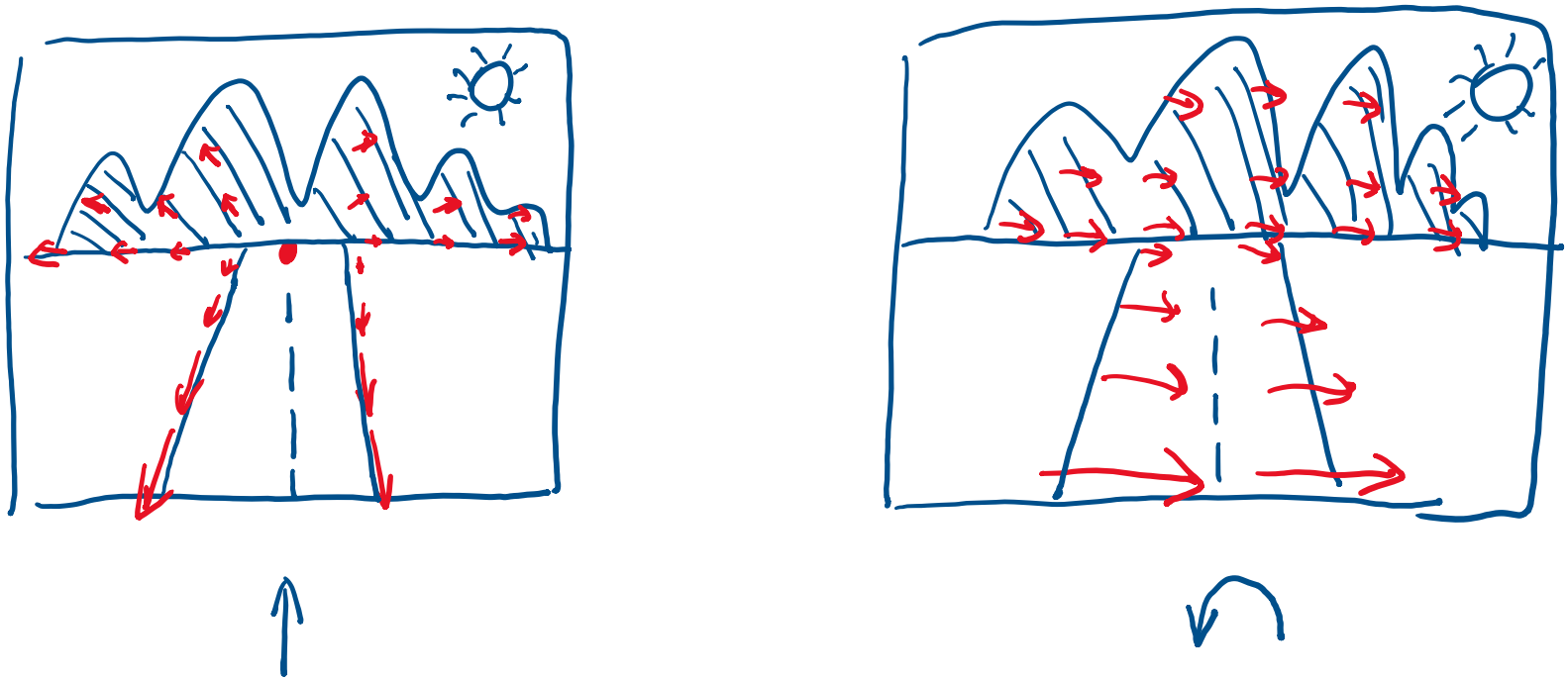
Can you tell whether it is because of the motion of the sphere, or the light source?

Information loss may be due to:

1. 3D to 2D projection
2. Non-informative texture on objects

Information in Optical Flow

- Motion information: translation and rotation



We always want to recover motion (2D/3D)!

Information in Optical Flow

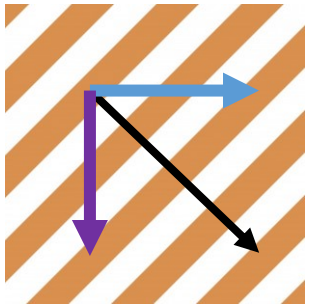
- Why motion recovery not easy?
  - Flow depending on scene/background, but **scene unknown**

  → usually recover motion together with scene (reconstruction)

  - Ego-motion coupled with translation and rotation, **different motion may lead to same flow**

  → internal measurements (IMU, other sensors)

  → build theoretically-robust model for motion and scene



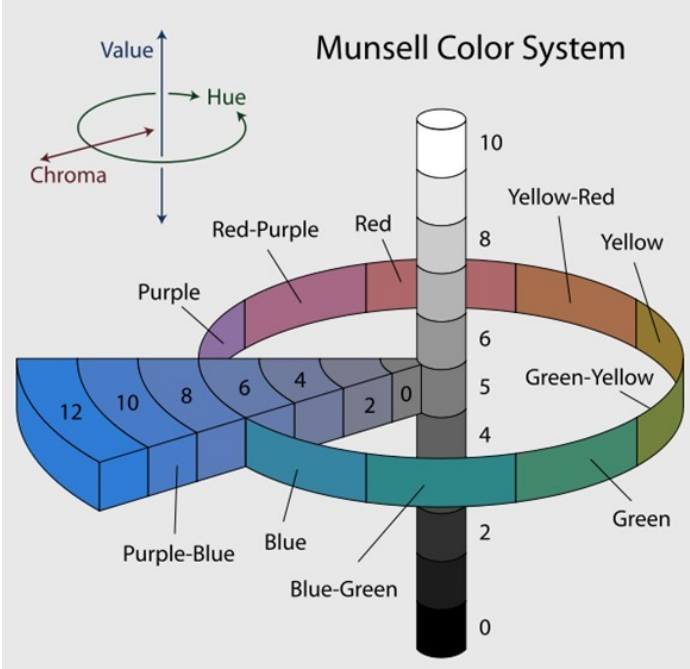
  - 3-D to 2-D projection, **irreversible information loss**

  → camera needs to move, observe from multi-angles to reconstruct 3-D

Estimating flow is the first step.

Visualization of Optical Flow

- HSV instead of RGB representation
  - Hue
  - Saturation/Chroma
  - Value



Frame-based Flow Estimation

- The problem formulation:

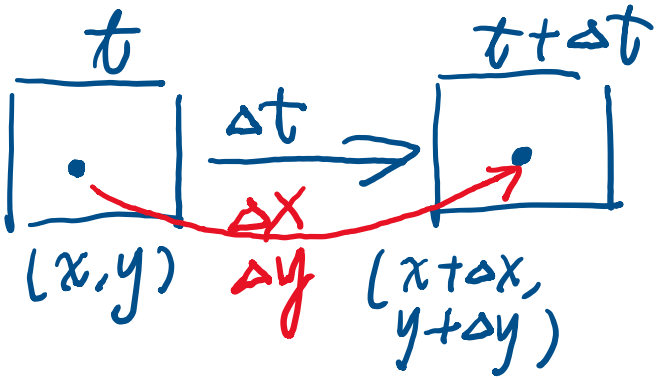
  Optimization of a global energy function

$$E_{\text{Global}} = E_{\text{Data}} + \lambda E_{\text{Prior}}.$$

  - **Data term:** how consistent the optical flow is with the input images;
  - **Prior term:** it favors certain flow fields over others (e.g. smoothly varying flow fields).

Flow Estimation: Data Term

- Brightness consistency → **optical flow constraint equation**



$$I(x,y,t) = I(x+\Delta x, y+\Delta y, t+\Delta t)$$
$$= I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + \text{H.o.T.}$$

$$\Rightarrow \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0$$

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = 0$$

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0$$

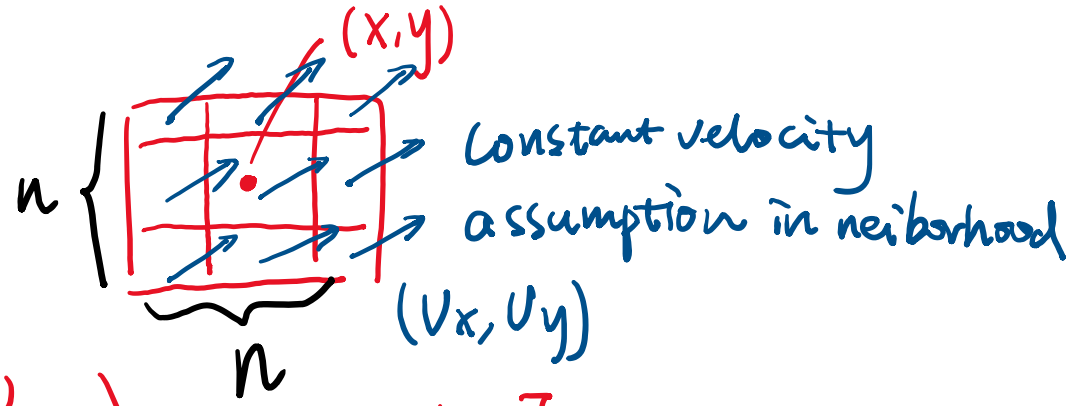
$$I_x V_x + I_y V_y = -I_t$$

Above equation leads to one error per pixel…

How to get the global constraints for the whole image?

Flow Estimation: Data Term

- Aggregating error over the local patch (**Lucas-Kanade** method)



$$m=n^2 \text{ eqns} \quad \begin{pmatrix} \frac{\partial I(x_1,y_1)}{\partial x} & \frac{\partial I(x_1,y_1)}{\partial y} \\ \vdots & \vdots \\ \frac{\partial I(x_m,y_m)}{\partial x} & \frac{\partial I(x_m,y_m)}{\partial y} \end{pmatrix} \begin{pmatrix} V_x \\ V_y \end{pmatrix} = \begin{pmatrix} -I_{t_1} \\ \vdots \\ -I_{tm} \end{pmatrix}$$

Use the **Least Squares Method** to solve above **overdetermined system**.
Only **local** and **sparse flow** can be obtained.

Flow Estimation: Data Term

- Aggregating error over the whole image
  - L2 norm (used in **Horn and Schunck** algorithm)

$$E_{\text{Data}} = \sum_{x,y} \left[u \frac{\partial I}{\partial x} + v \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} \right]^2.$$

  - L1 norm

$$\|\mathbf{E}\|_1 = \sum_{x,y} |E_{x,y}| \approx \sum_{x,y} \sqrt{\|E_{x,y}\|^2 + \epsilon^2}$$

Flow Estimation: Prior Term

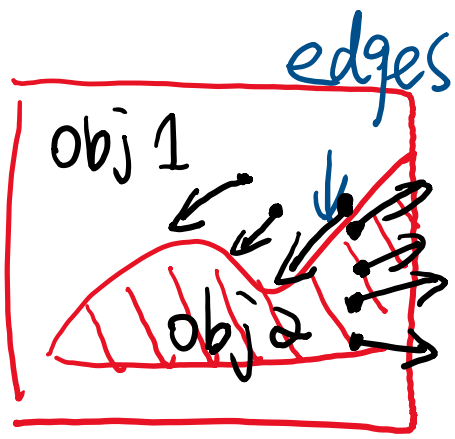
- Spatial and temporal smoothness
  - Favor small first-order derivatives (gradients) of the flow field (used in **Horn and Schunck** algorithm)

$$E_{\text{Prior}} = \sum_{x,y} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right].$$

  - Also, possible to add temporal smoothness terms $\frac{\partial u}{\partial t}$ and $\frac{\partial v}{\partial t}$.

Flow Estimation: Prior Term

- Different weights for different locations. **Why?**
  - Flow may not be smooth at edges.
  - We may want to reduce the weight of the prior at edges (with high $|\nabla I|$).



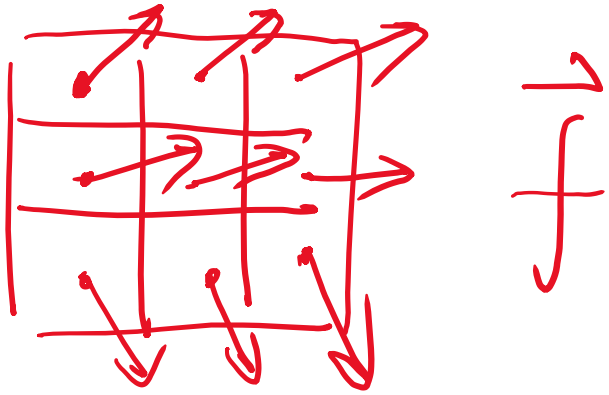
edges: may not satisfy the flow smoothness assumption

To do that, design the prior term as

$$E_{\text{Prior}} = \sum_{x,y} w(\nabla I)\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right].$$

Flow Estimation: Optimization

- Recap: **How to formulate the flow estimation problem?**



$\vec{f}$

Recall the flow est. prob.:

$\arg\min_{\vec{f}} \{E_{Global}\}$

vector field

The calculus of variations will help to solve the above problem.

- Hard to find the analytical solution due to various E_{Global}
- Numerical and iterative solution

Event-based Flow Estimation

- **Lucas-Kanade variants**

  No frames/intensities… How to obtain spatial and temporal gradients?

  1. For each neighborhood, use events to compute temporal gradients and approximate spatial gradients
  2. Find the local flow based on L-K method

  + Straightforward and easy to implement

  - Noisy results, needs further filtering

  - Approximation on spatial gradients not accurate

  - Only sparse flow available

$$\begin{cases} \frac{\partial e(x,y,t)}{\partial x} \sim \sum_{t-\triangle t}^{t} e(x,y,t) - \sum_{t-\triangle t}^{t} e(x-1,y,t) \\ \frac{\partial e(x,y,t)}{\partial y} \sim \sum_{t-\triangle t}^{t} e(x,y,t) - \sum_{t-\triangle t}^{t} e(x,y-1,t) \end{cases}$$

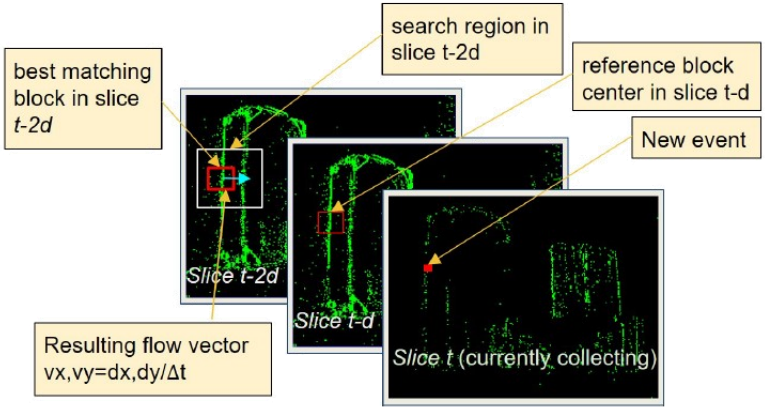
$$\frac{\partial e(x,y,t)}{\partial t} \sim \frac{\sum_{t-\triangle t}^{t_1} e(x,y,t) - \sum_{t-\triangle t}^{t} e(x,y,t)}{t - t_1}$$

$$\text{grad}^t(I) \begin{pmatrix} v_x \\ v_y \end{pmatrix} + \frac{\partial I}{\partial t} = 0$$

Benosman, Ryad, et al. "Asynchronous frameless event-based optical flow." Neural Networks 27 (2012): 32-37.

Event-based Flow Estimation

- **Block matching**
  - Idea: match **block,** instead of a single pixel



+ More robust as block may contain more information

- Not good for complex scenes due to repetitive textures and contours

- Defining block features is much more difficult than defining point features!

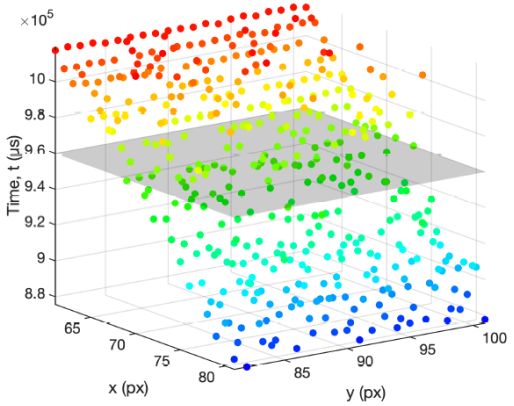
http://www.bmva.org/bmvc/2018/contents/papers/0280.pdf

Liu, Min, and Tobi Delbruck. "Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors." (2018).

Event-based Flow Estimation

- **Local plane fitting**
  1. Create **event time surface**
  2. Use a local plane to fit the time surface
  3. Gradient (normal vector) of the plane indicates the flow

  - Noisy results that requires further filtering



- **Energy minimization/optimization** (deep learning)

  \+ Global method

  \+ Able to provide dense flow

  - High computational complexity

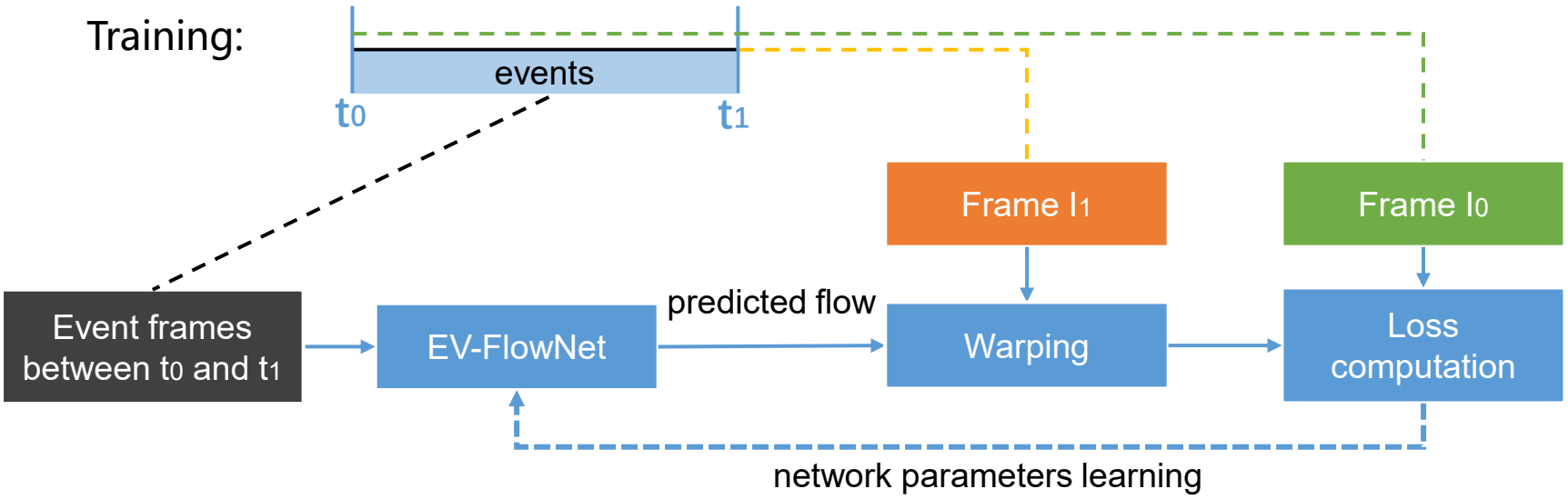
  - Data-driven for deep learning

$$\min_{\mathbf{u},L}\int_{\Omega}\int_{T}\Big(\lambda_1\|\mathbf{u}_{\mathbf{x}}\|_1+\lambda_2\|\mathbf{u}_t\|_1+\lambda_3\|L_{\mathbf{x}}\|_1+$$
$$\lambda_4\|\langle L_{\mathbf{x}},\delta_t\mathbf{u}\rangle+L_t\|_1+\lambda_5\,h_\theta(L-L(t_p))\Big)\mathrm{d}t\,\mathrm{d}\mathbf{x}$$
$$+\int_{\Omega}\sum_{i=2}^{|P(\mathbf{x})|}\|L(t_i)-L(t_{i-1})-\theta\rho_i\|_1\mathrm{d}\mathbf{x},$$

Ieng, Sio Hoi, João Carneiro, and Ryad Benjamin Benosman. "Event-Based 3D Motion Flow Estimation Using 4D Spatio Temporal Subspaces Properties." Frontiers in neuroscience 10 (2017): 596.
Benosman, Ryad, et al. "Event-based visual flow." IEEE transactions on neural networks and learning systems 25.2 (2013): 407-417.
Low, Weng Fei, et al. "SOFEA: A Non-iterative and Robust Optical Flow Estimation Algorithm for Dynamic Vision Sensors." CVPR Workshops. 2020.
Pan, Liyuan, Miaomiao Liu, and Richard Hartley. "Single image optical flow estimation with an event camera." 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.
Bardow, Patrick, Andrew J. Davison, and Stefan Leutenegger. "Simultaneous optical flow and intensity estimation from an event camera." CVPR. 2016.

Paper: EV-FlowNet

- **Self-Supervised** Optical Flow Estimation for Event-based Cameras
  - Self-supervised: no ground-truth flow required
  - Groundtruth frames required **only in training stage**

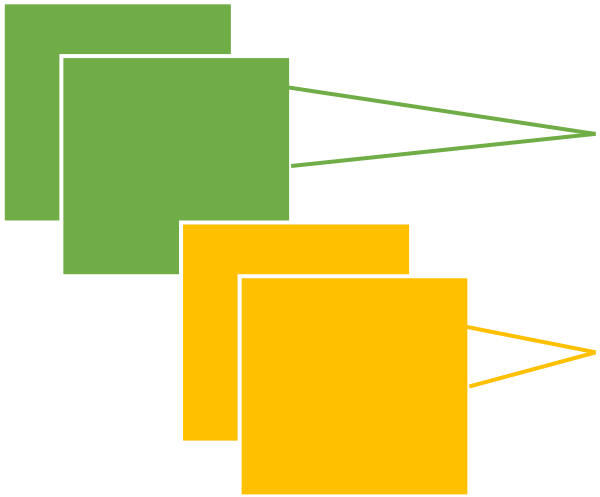


Inference: The input only contains events.

[1802.06898] EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras (arxiv.org)

Paper: EV-FlowNet

- Event representation
  - **Image form representation** instead of the point cloud form
  - Input to the network: **4 channel image**, same resolution as the camera



Event timestamp images (+/-) after normalization to [0, 1]

Event count images:
Number of +/- events that have occurred at each pixel

[1802.06898] EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras (arxiv.org)

Paper: EV-FlowNet

- Self-supervised loss

Total loss

- **Photometric loss**: minimize the difference in intensity between the warped second image and the first image

$$\ell_{\text{photometric}}(u, v; I_t, I_{t+1}) = \sum_{x,y} \rho(I_t(x, y) - I_{t+1}(x + u(x, y), y + v(x, y)))$$

where ρ is the Charbonnier loss function for outlier rejection.

- **Smoothness loss**: regularize the output flow by minimizing the difference in flow between neighboring pixels horizontally, vertically and diagonally.

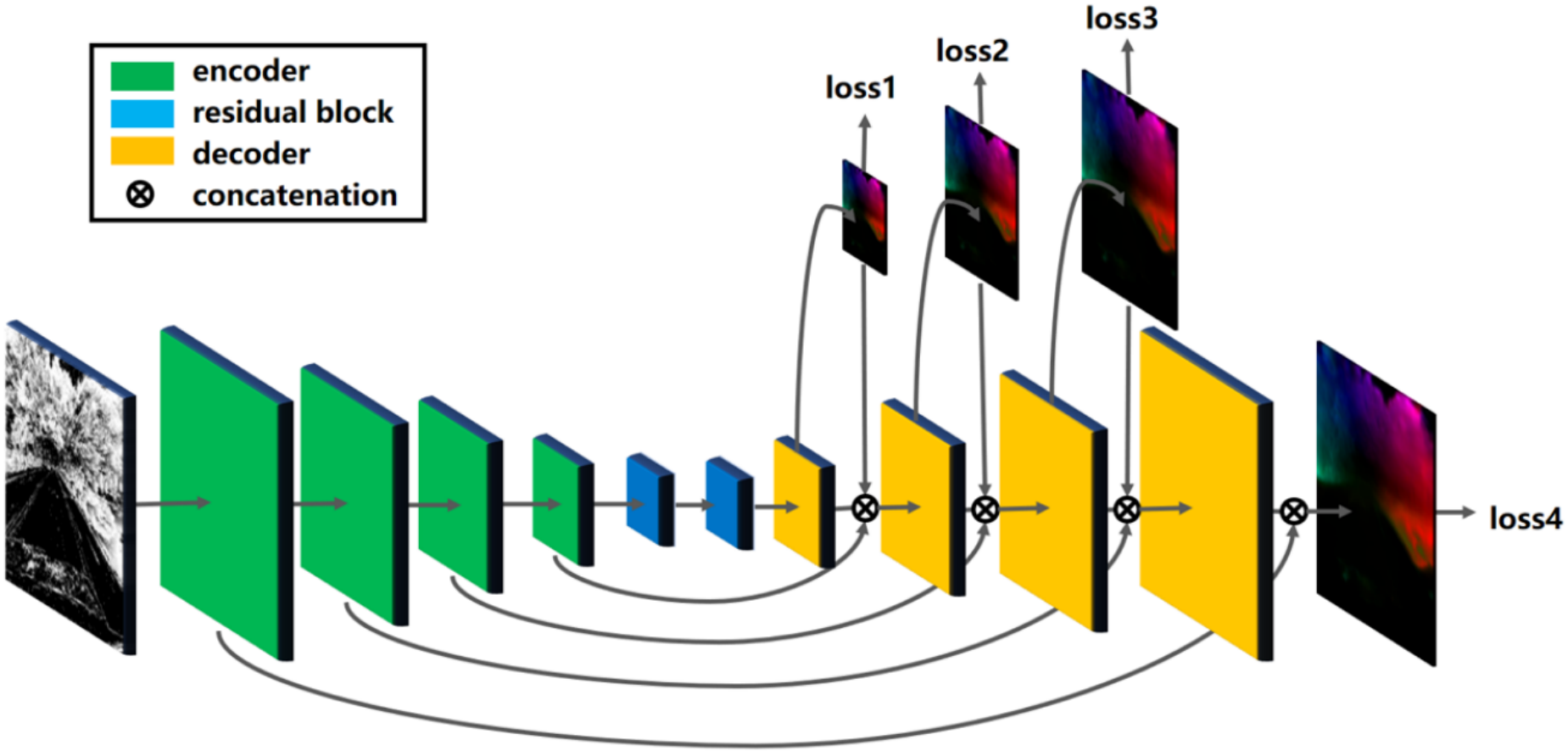
$$\ell_{\text{smoothness}}(u, v) = \sum_{x,y} \sum_{i,j \in \mathcal{N}(x,y)} \rho(u(x, y) - u(i, j)) + \rho(v(x, y) - v(i, j))$$

where N is the set of neighbors around (x, y).

[1802.06898] EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras (arxiv.org)

Paper: EV-FlowNet

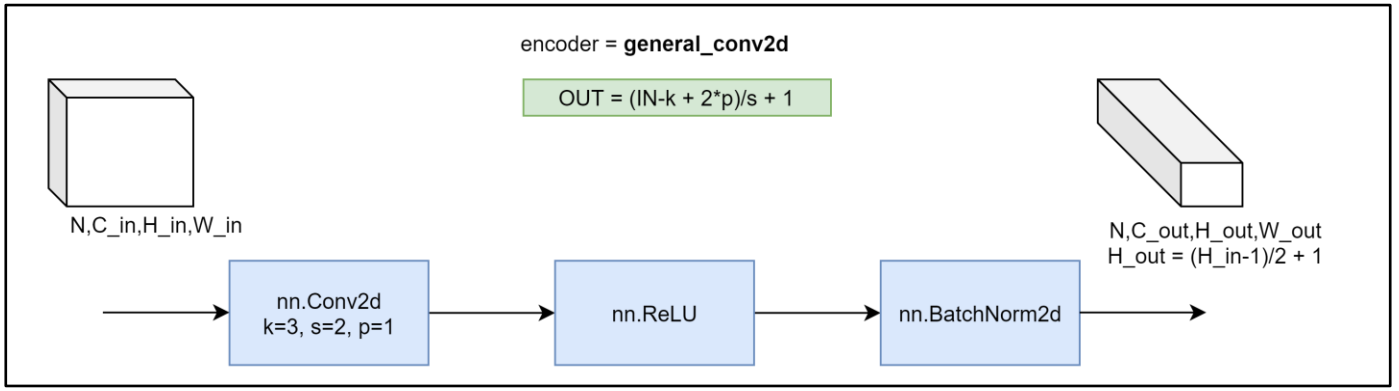
- EV-FlowNet architecture

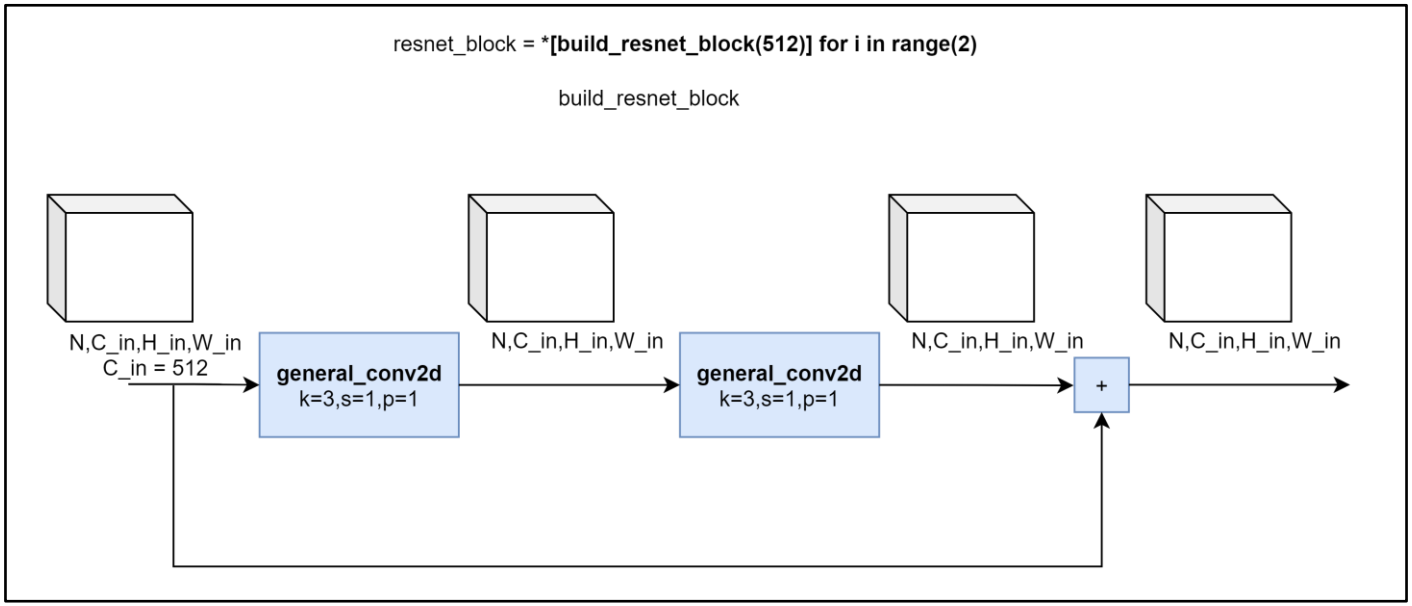


[1802.06898] EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras (arxiv.org)

Paper: EV-FlowNet

- EV-FlowNet architecture in details: sub-modules

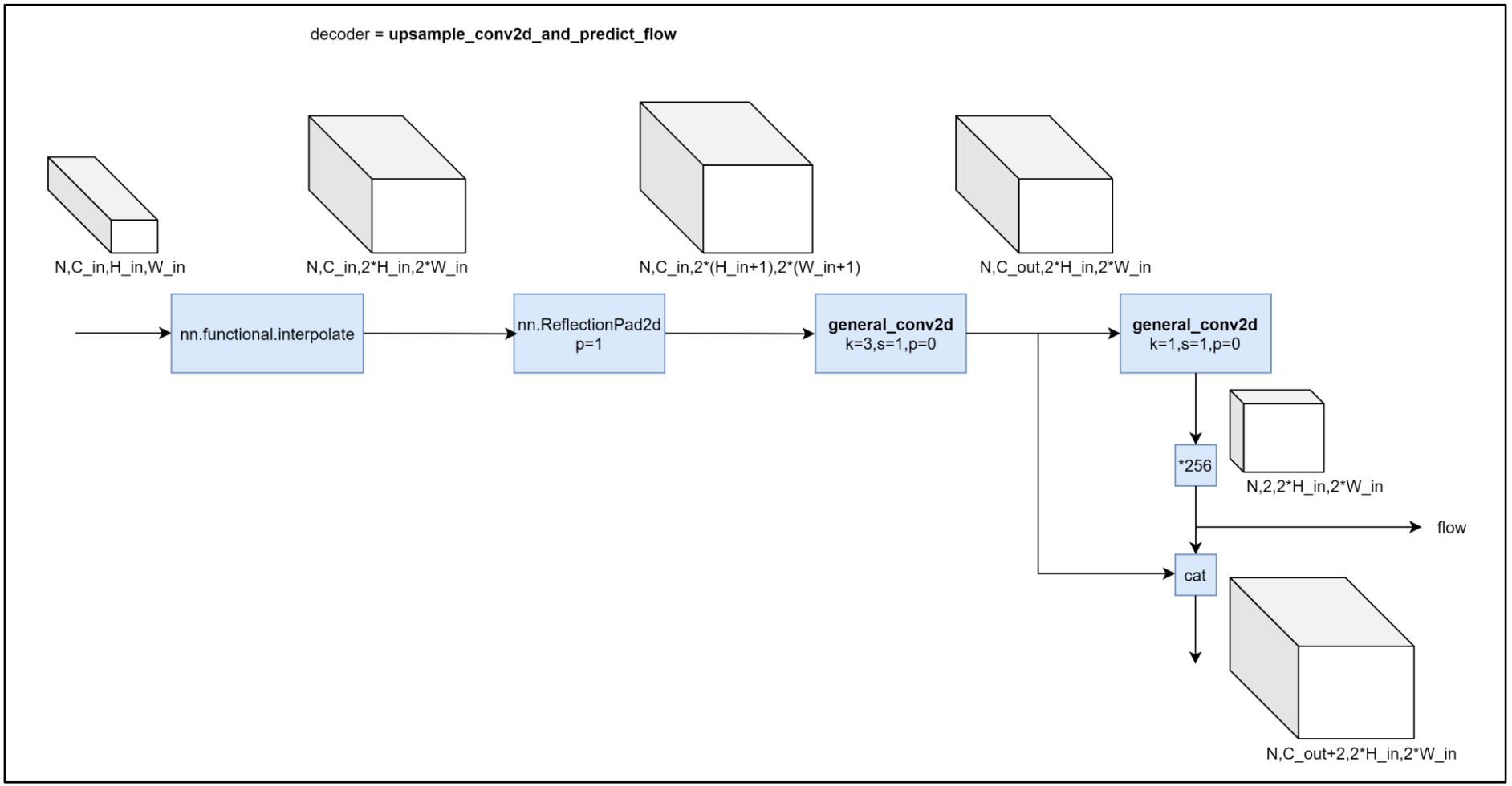




[1802.06898] EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras (arxiv.org)

Paper: EV-FlowNet

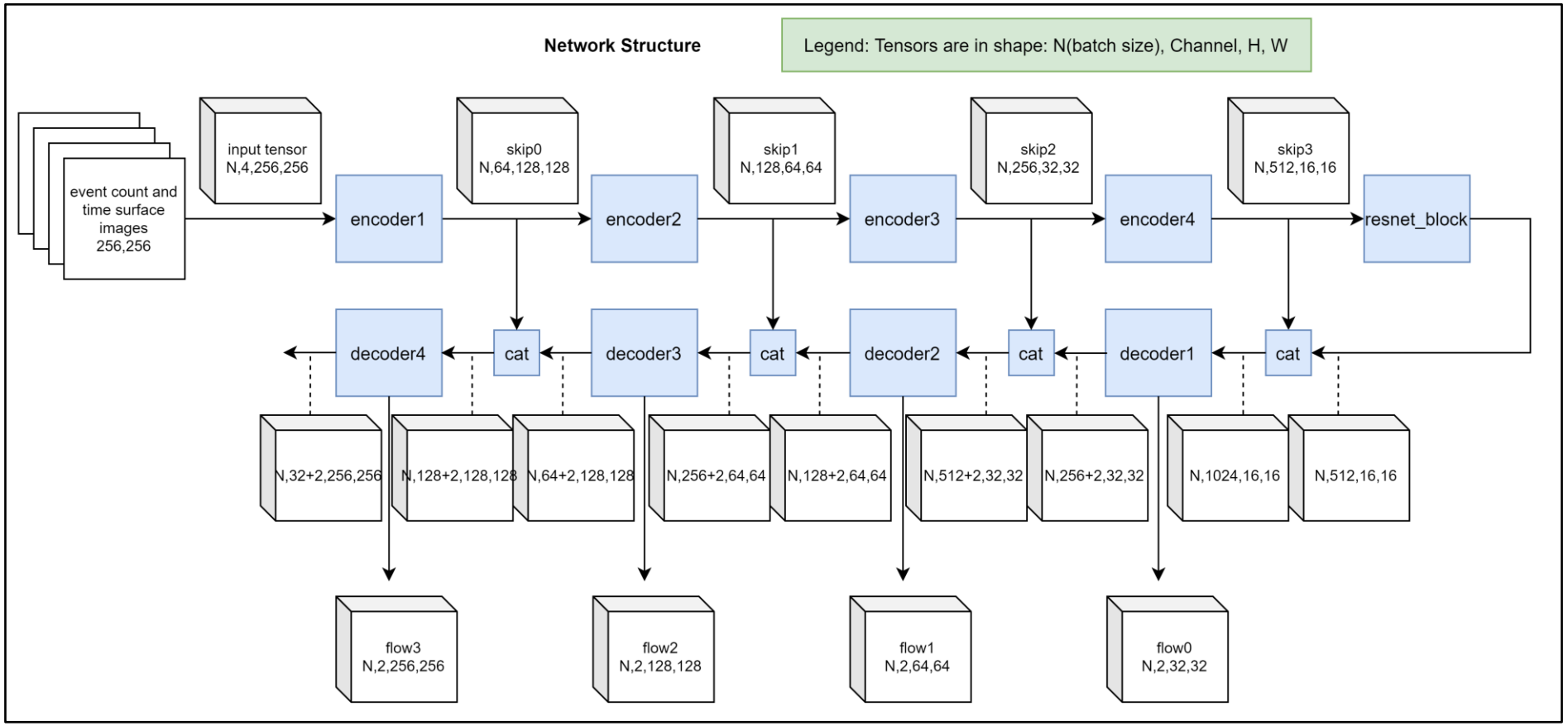
- EV-FlowNet architecture in details: sub-modules



[1802.06898] EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras (arxiv.org)

Paper: EV-FlowNet

- EV-FlowNet architecture in details: framework



[1802.06898] EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras (arxiv.org)

Paper: EV-FlowNet

EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras

Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, Kostas Daniilidis

Penn Engineering | GRASP Laboratory
General Robotics, Automation, Sensing & Perception Lab

[1802.06898] EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras (arxiv.org)

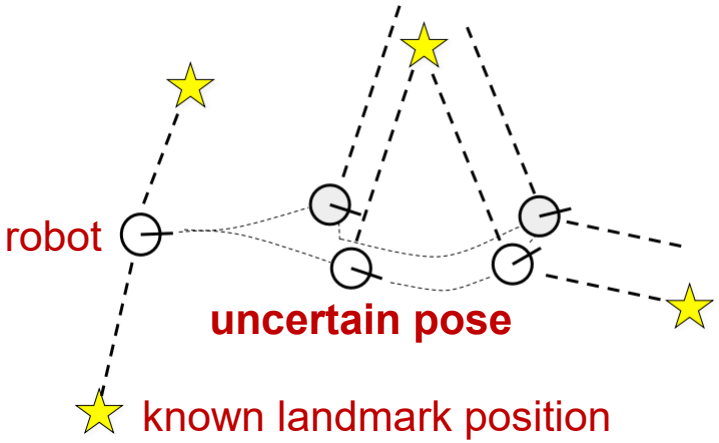
Summary – Optical flow estimation

- Optical flow field vs. motion field
- Information in optical flow: Why it is necessary to obtain optical flow?
- Visualization of optical flow
- Frame-based optical flow estimation
- Event-based flow estimation
  - Lucas-Kanade variants
  - Block matching
  - Local plane fitting
  - Energy minimization/optimization (with/without deep learning)

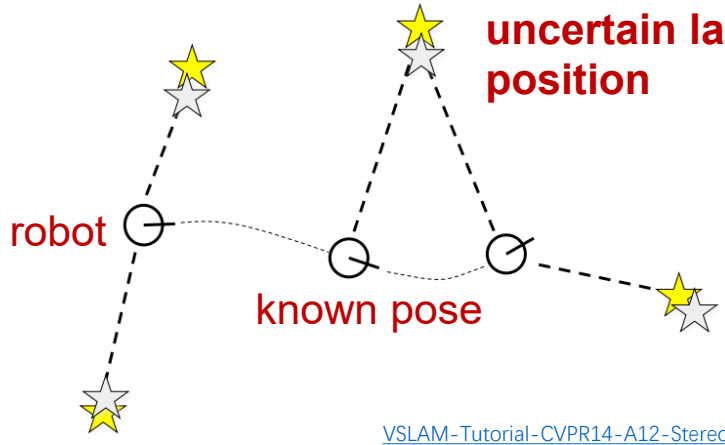
Pose Estimation and SLAM

- Problem
  - Pose estimation:
    Estimating **robot's poses only**.
  - SLAM (Simultaneous Localization And Mapping):
    Computing the **robot's poses** and the **map of the environment** at the same time.

Example of localization/pose estimation

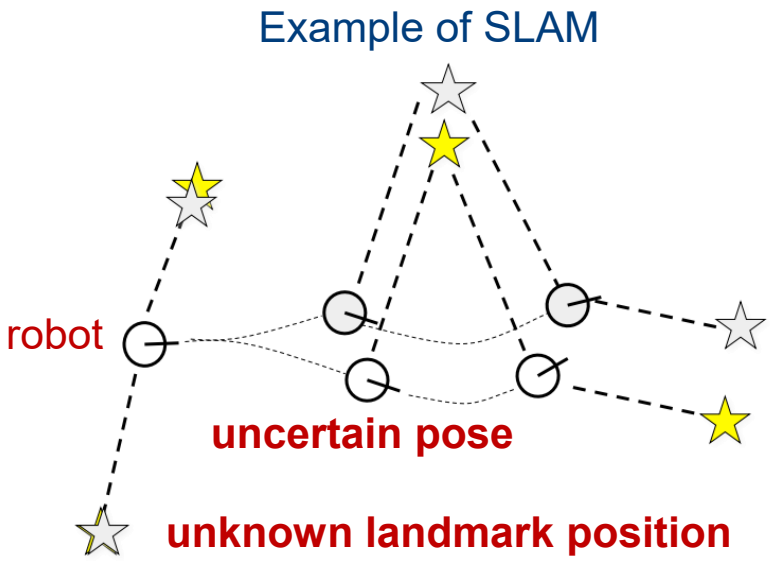


Example of mapping



VSLAM-Tutorial-CVPR14-A12-StereoVO.pdf (cmu.edu)
Slide 1 (toronto.edu)
Robot Mapping - WS 2013/14 - Arbeitsgruppe: Autonome Intelligente Systeme (uni-freiburg.de)

Pose Estimation and SLAM



- SLAM is a chicken-or-egg problem:
  - a map is needed for localization and
  - a pose estimate is needed for mapping

Robot Mapping - WS 2013/14 - Arbeitsgruppe: Autonome Intelligente Systeme (uni-freiburg.de)

Pose Estimation and SLAM

- Definition of the SLAM Problem

  Given

  - The robot's controls

$$u_{1:T} = \{u_1, u_2, u_3 \ldots, u_T\}$$

  - Observations

$$z_{1:T} = \{z_1, z_2, z_3 \ldots, z_T\}$$

  Wanted

  - Map of the environment

$$m$$

  - Path of the robot

$$x_{0:T} = \{x_0, x_1, x_2 \ldots, x_T\}$$

Robot Mapping - WS 2013/14 - Arbeitsgruppe: Autonome Intelligente Systeme (uni-freiburg.de)

Pose Estimation and SLAM

- Definition of the SLAM Problem

Given

- The robot's controls

$$u_{1:T} = \{u_1, u_2, u_3 \ldots, u_T\}$$

- Observations

$$z_{1:T} = \{z_1, z_2, z_3 \ldots, z_T\}$$

Wanted

- Map of the environment

$$m$$

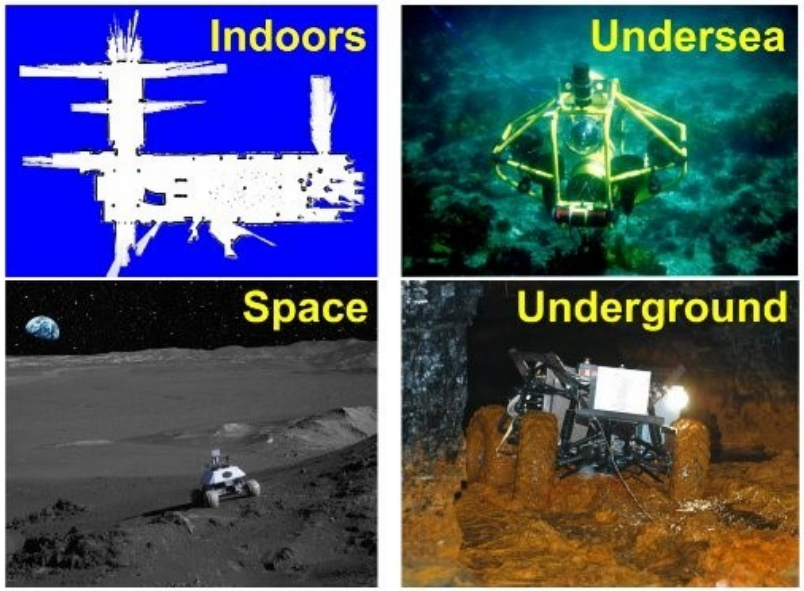
- Path of the robot

$$x_{0:T} = \{x_0, x_1, x_2 \ldots, x_T\}$$

Robot Mapping - WS 2013/14 - Arbeitsgruppe: Autonome Intelligente Systeme (uni-freiburg.de)

Pose Estimation and SLAM

- Why we need SLAM since we have the GPS and the Google Map?



- What we want for SLAM solutions?
  - Fast control feedback;
  - Environment flexibility, e.g., working in high dynamic range scenes;
  - Low power consumption and hardware complexity.

research-ar-lecture-series-slides-by-hanme-kim-visual-slam.pdf (qualcomm.com)

Pose Estimation and SLAM

- What sensors have been used?
  - Passive sensors
    - Monocular
    - Stereo
    - Omnidirectional
  - Active sensors
    - Lidar
    - Time-of-flight
    - RGB-Depth
  - Hybrid sensors
    - Uses multiple sensors

UEye Camera


Point Grey Stereo Cam

Bubl Omnicam


Velodyne Lidar

Mesa TOF Cam

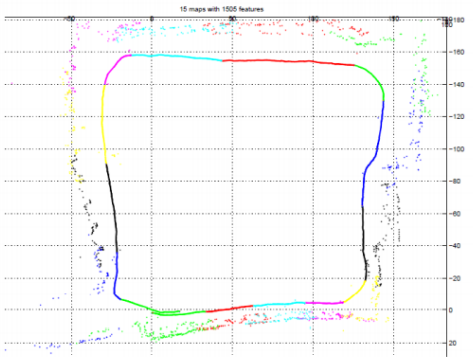
Kinect RGB-D

Slide 1 (toronto.edu)

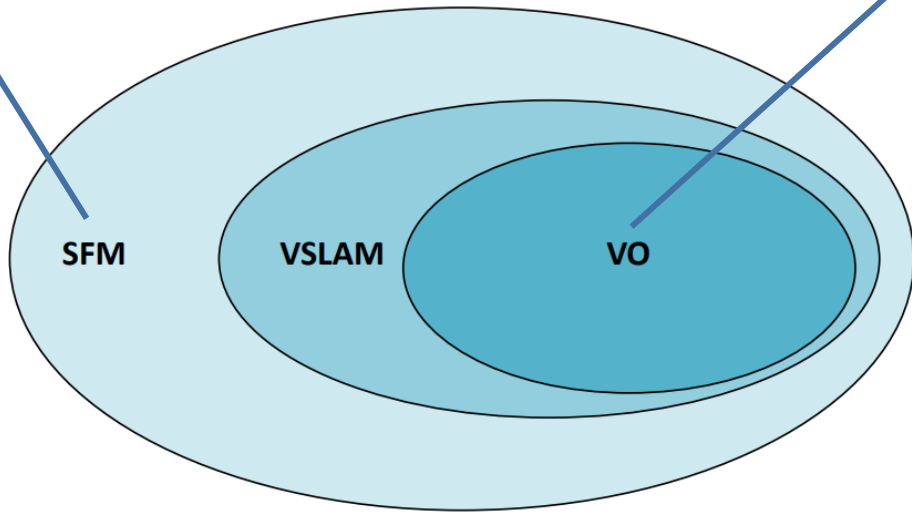
Visual Odometry (VO), VSLAM, and SFM

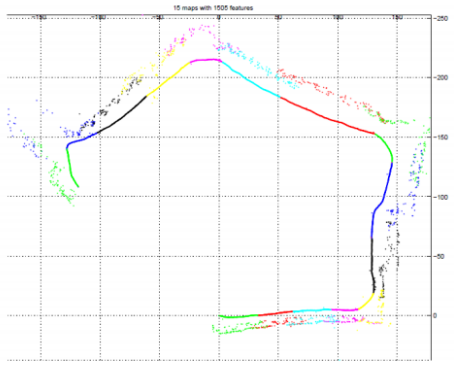
3-D reconstruction and 6-DOF pose estimation from **unordered image sets**

Estimating the 3D motion of the camera **sequentially** and **in real time**



Visual SLAM





Visual odometry

- VO: Focusing on **local consistency** of the trajectory
- SLAM: Focusing on **global consistency** of the trajectory

VO can be used as a building block in a SLAM system.

Robots that know what they do (upenn.edu)
Robots that know what they do (uzh.ch)

Visual Odometry (VO)

- VO Working Principle

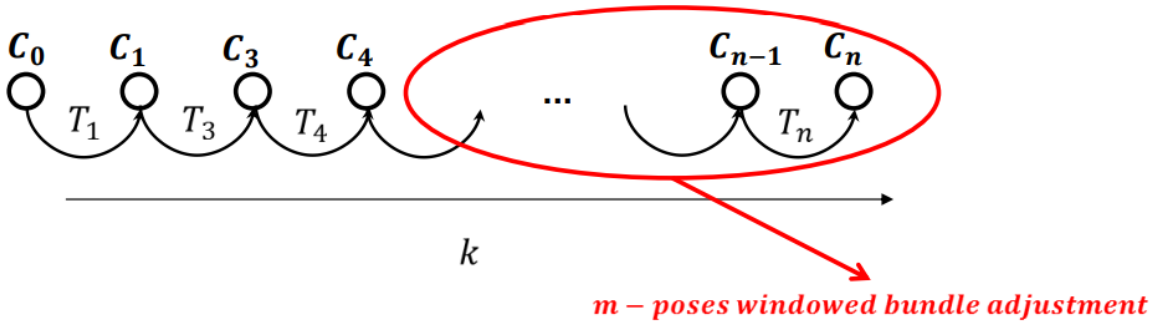
1. Compute the relative motion T_k from images I_{k-1} to image I_k

$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix}$$

2. Concatenate them to recover the full trajectory

camera pose $C_n = C_{n-1}T_n$ transformation

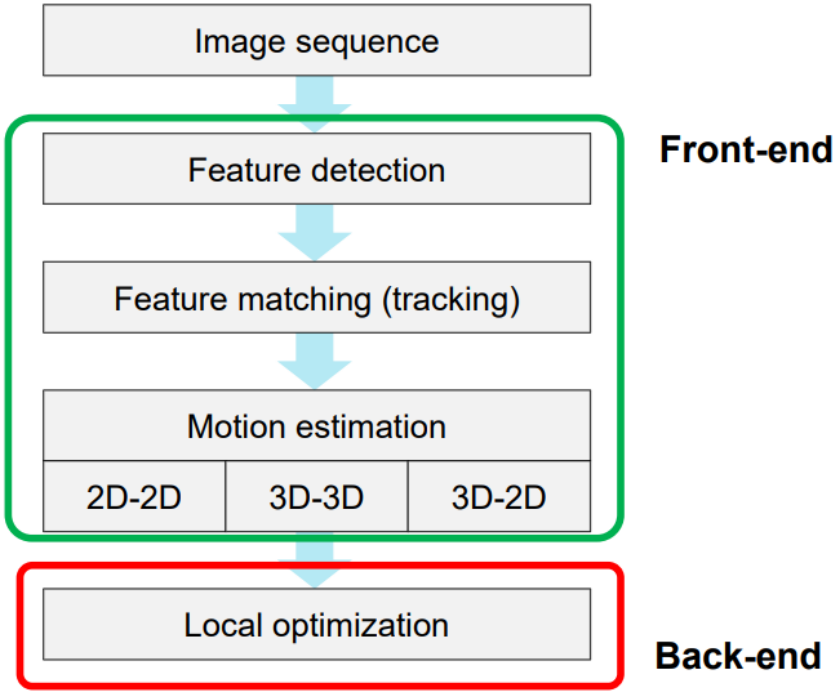
3. An optimization over the last m poses can be done to refine locally the trajectory (Pose-Graph or Bundle Adjustment)



(5) (PDF) Visual Odometry [Tutorial] (researchgate.net)

Front-End and Back-End

- VO systems contains the front-end and the back-end.
- The **front-end** is responsible for
  - Feature extraction, matching, and outlier removal
  - Loop closure detection
  - Motion estimation between two frames
- The **back-end** is responsible for the pose and structure optimization (e.g., with libraries iSAM, g2o, Google Ceres)

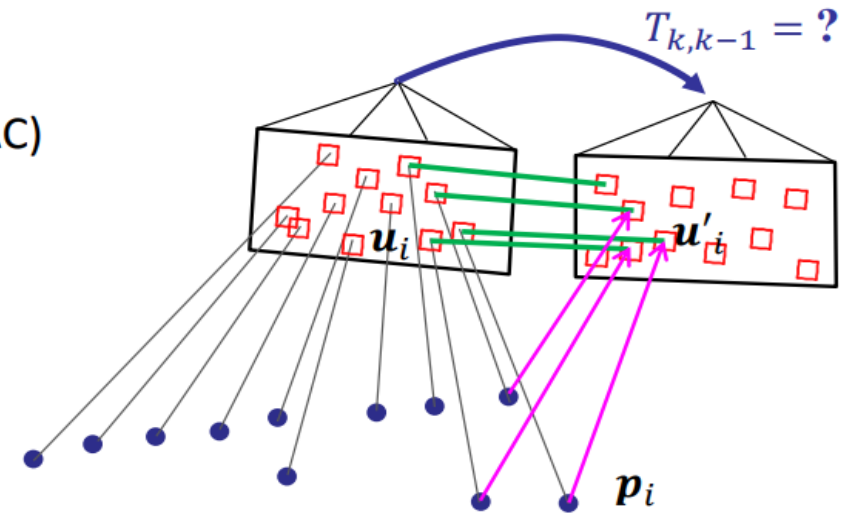


(5) (PDF) Visual Odometry [Tutorial] (researchgate.net)

Estimating the Relative Motion $\mathbf{T_k}$

Feature-based methods

1. Extract & match features (+RANSAC)
2. Minimize **Reprojection error** minimization



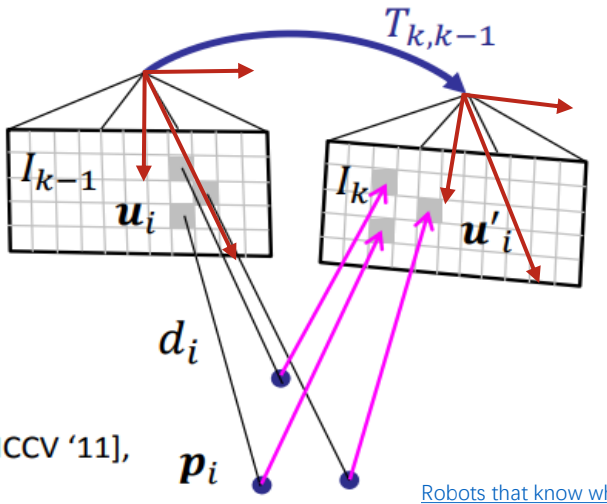
$$T_{k,k-1} = \arg\min_T \sum_i \| u'_i - \pi(p_i) \|^2_\Sigma$$

Direct methods

1. Minimize **photometric error**

$$T_{k,k-1} = \arg\min_T \sum_i \| I_k(u'_i) - I_{k-1}(u_i) \|^2_\sigma$$

where $u'_i = \pi\left(T \cdot (\pi^{-1}(u_i) \cdot d)\right)$



[Jin,Favaro,Soatto'03] [Silveira, Malis, Rives, TRO'08], [Newcombe et al., ICCV '11], [Engel et al., ECCV'14], [Forster et al., ICRA'14]

Robots that know what they do (upenn.edu)

Solving Relative Motion $\mathbf{T}_k$ by Optimization

- Graph Representation

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{<i,j>\in\mathcal{C}} \underbrace{\mathbf{e}_{ij}(\mathbf{z}_{ij}, \mathbf{x}_{ij})^\top \mathbf{\Omega}_{ij} \mathbf{e}_{ij}(\mathbf{z}_{ij}, \mathbf{x}_{ij})}_{\mathbf{F}_{ij}}$$

- To linearize:

$$\mathbf{e}_{ij}(\mathring{\mathbf{x}}_{ij} + \Delta\mathbf{x}_{ij}) \simeq \mathbf{e}_{ij}(\mathring{\mathbf{x}}_{ij}) + \mathbf{J}_{ij}\Delta\mathbf{x}_{ij}$$

error function

$$\mathbf{e}_{ij}(\mathbf{z}_{ij}, \mathbf{x}_i, \mathbf{x}_j) = \mathbf{z}_{ij} \ominus \mathbf{h}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$$

$\mathbf{x}_i$: state of vertex i;
$\mathbf{h}_{ij}(\mathbf{x}_i, \mathbf{x}_j)$: measurement equation;
$\mathbf{z}_{ij}$: actual measurement.

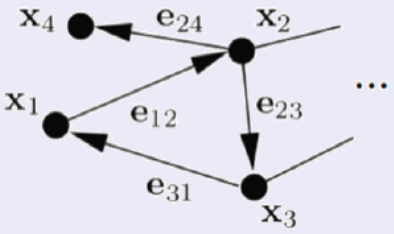
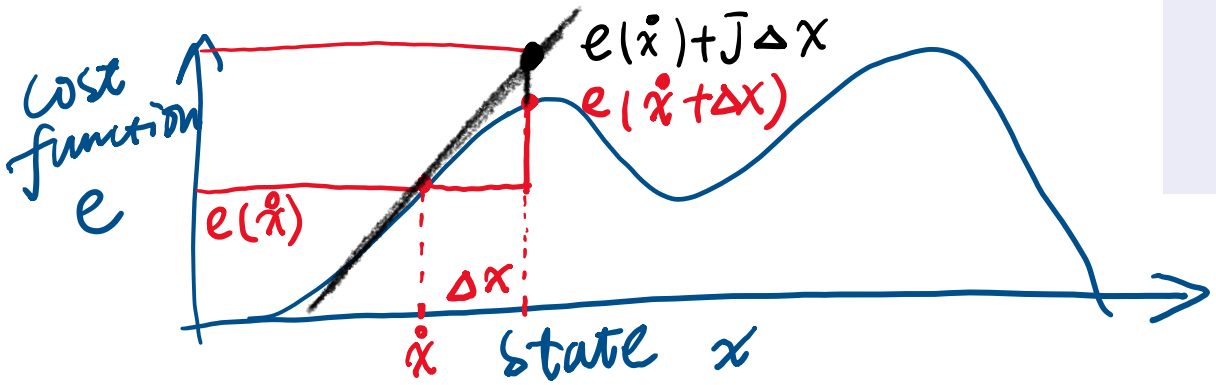
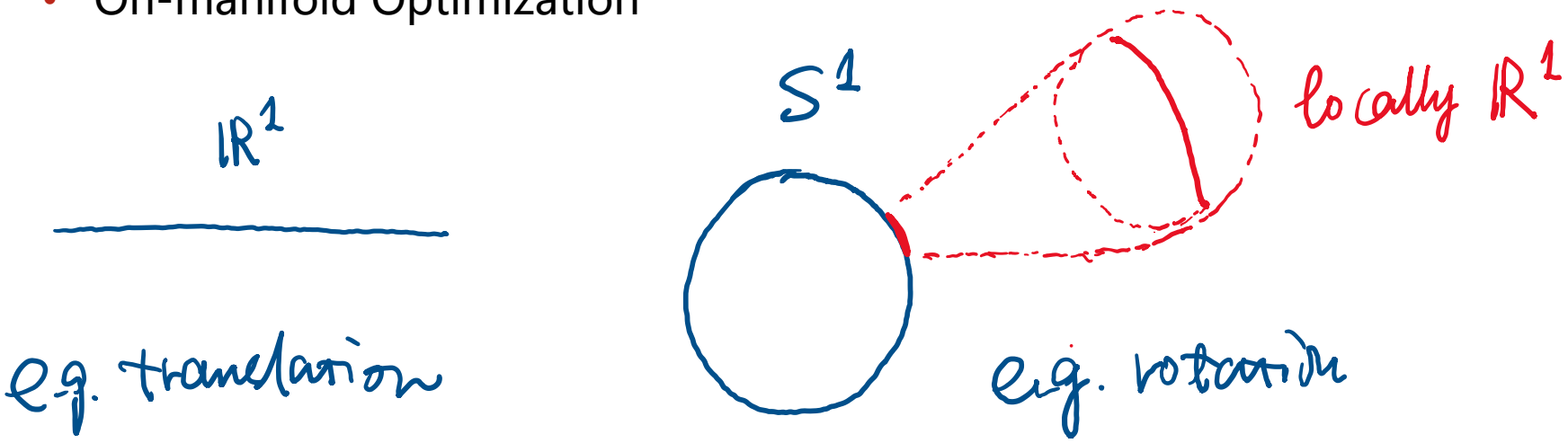


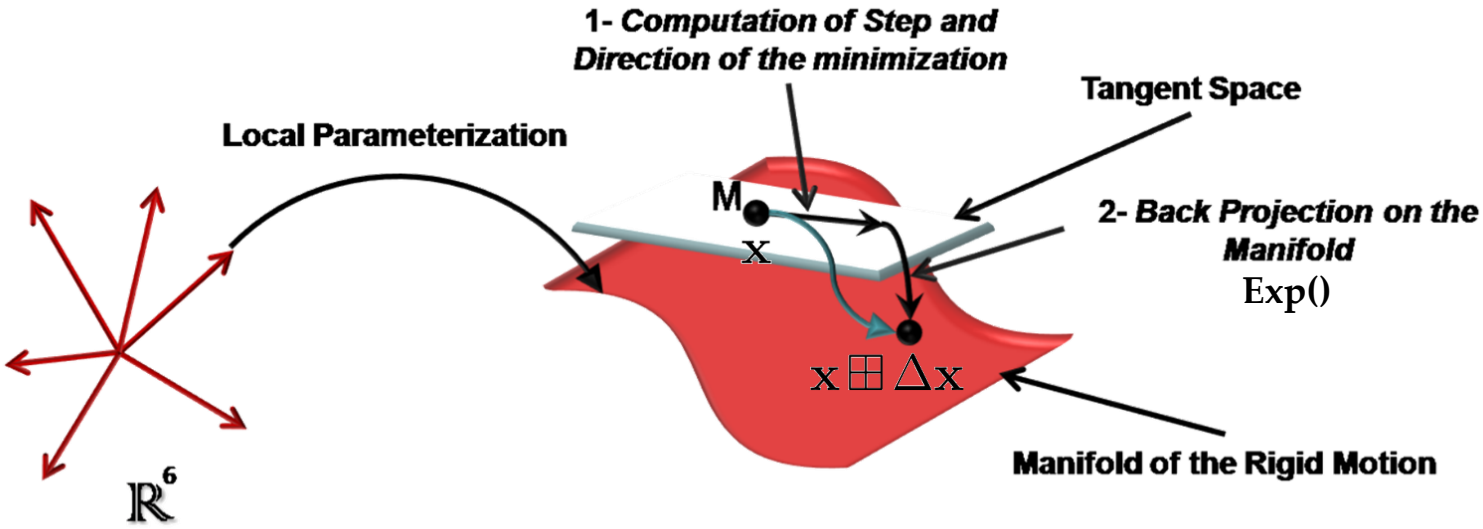
Figure: Pose graph representation [1].



Solving Relative Motion T_k by Optimization

- On-manifold Optimization





Pose Estimation and SLAM with an Event Camera

- Challenges
  - Event streams cannot be fed directly to existing methods designed for standard cameras!

- Ideas
  - Classification by data type:

    event frames v.s. **event point clouds**
  - Classification by way of processing:

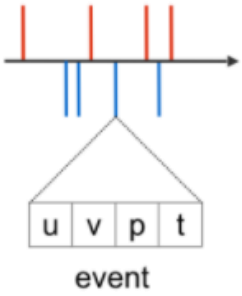
    batch-based v.s. **event-based**

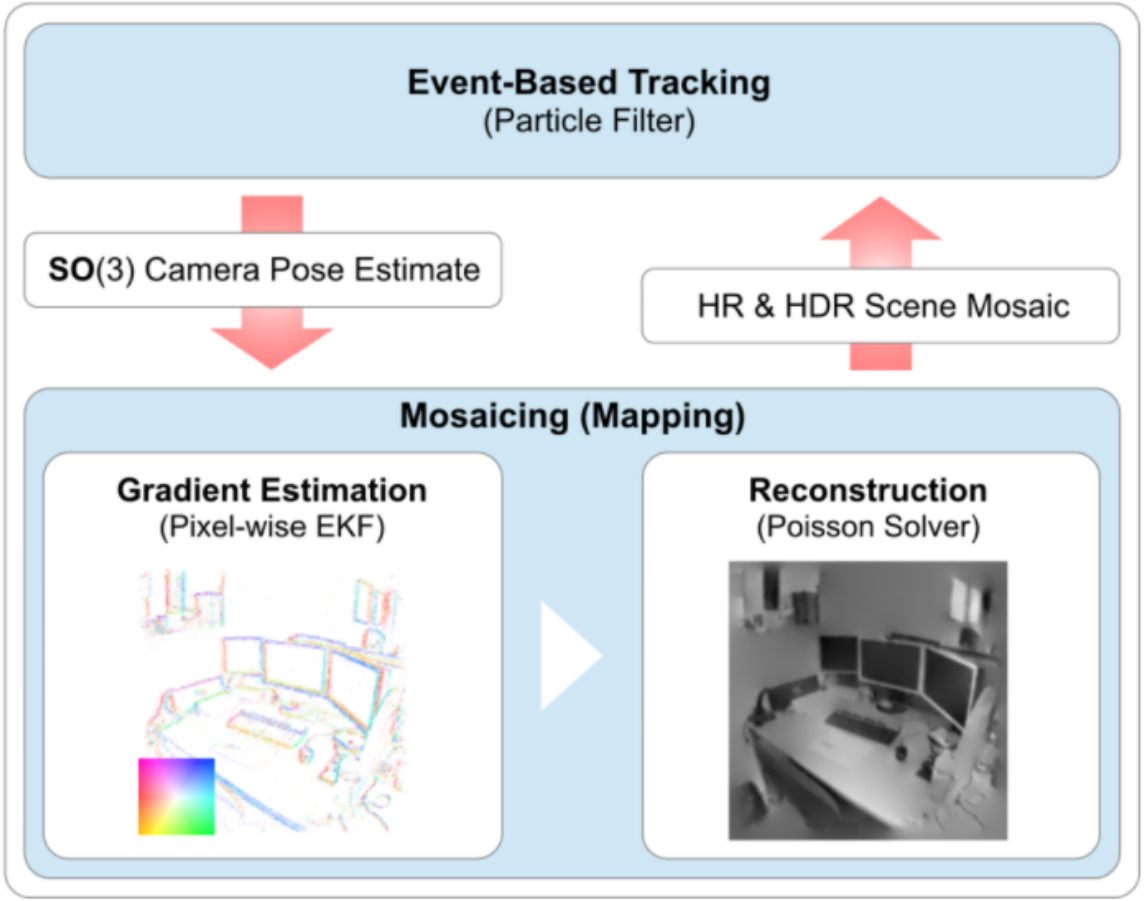
- Practical considerations (besides accuracy)
  - Latency
  - Computation complexity / power consumption

Yi Zhou, Event-based Visual Odometry: A Short Tutorial. PowerPoint Presentation (tub-rip.github.io)

Paper: Simultaneous Mosaicing and Tracking with an Event Camera

- Proposed algorithm
  - Simplified problem: only considers 3-DoF camera ego-motion
  - No frame involved





Simultaneous Mosaicing and Tracking with an Event Camera (bmva.org)

Paper: Simultaneous Mosaicing and Tracking with an Event Camera

- **Event-based camera tracking based on a particle filter**

Use N particles to represent camera pose: $\{\mathbf{p}_1^{(t)}, \mathbf{p}_2^{(t)}, \ldots, \mathbf{p}_N^{(t)}\}$

3D rotation group $\mathrm{R}_i^{(t)} \in \mathbf{SO}(3)$ $w_i^{(t)}$

Two-step process:

1) Motion prediction: constant pose motion model

$$\mathrm{R}_i^{(t)} = \mathrm{R}_i^{(t-\tau)} \exp\left(\sum_{k=1}^{3} \mathbf{n}_k \mathrm{G}_k\right)$$

Lie group generators

Gaussian distributed noise vector

Simultaneous Mosaicing and Tracking with an Event Camera (bmva.org)

Paper: Simultaneous Mosaicing and Tracking with an Event Camera

- **Event-based camera tracking based on a particle filter**

2) Measurement update using Bayes rule

$$w_i^{(t)} = P(z|\mathrm{R}_i^{(t)})w_i^{(t-\tau)}$$

How to define measurement *z*?

Difference on log intensity at the **same triggered pixel**

$$z = \log(\mathrm{M}(\mathbf{p}_m^{(t)})) - \log(\mathrm{M}(\mathbf{p}_m^{(t-\tau_c)})) ,$$

$$\text{where } \mathbf{p}_m^{(t)} = \pi\left(\mathrm{R}_i^{(t)}\mathrm{K}^{-1}\dot{\mathbf{p}}_c\right) .$$

K is the camera intrinsics;

$\pi(\mathbf{p}) = \frac{1}{p_2}(p_0, p_1)^\top$ is the projection function.

$$\dot{\mathbf{p}}_c = (u, v, 1)^\top$$

Simultaneous Mosaicing and Tracking with an Event Camera (bmva.org)

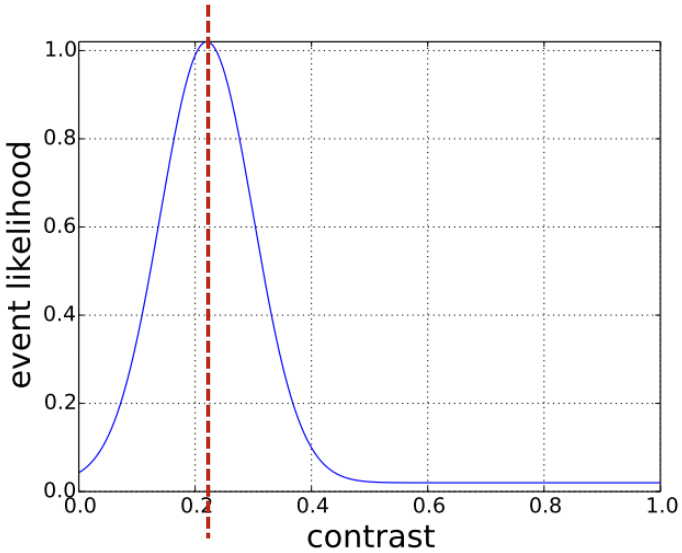
Paper: Simultaneous Mosaicing and Tracking with an Event Camera

- **Event-based camera tracking based on a particle filter**

2) Measurement update using Bayes rule

$$w_i^{(t)} = P(z|\mathbb{R}_i^{(t)})w_i^{(t-\tau)}$$

How to define conditional probability ***P***?



Theoretically, should be triggering threshold

Simultaneous Mosaicing and Tracking with an Event Camera (bmva.org)

Paper: Simultaneous Mosaicing and Tracking with an Event Camera

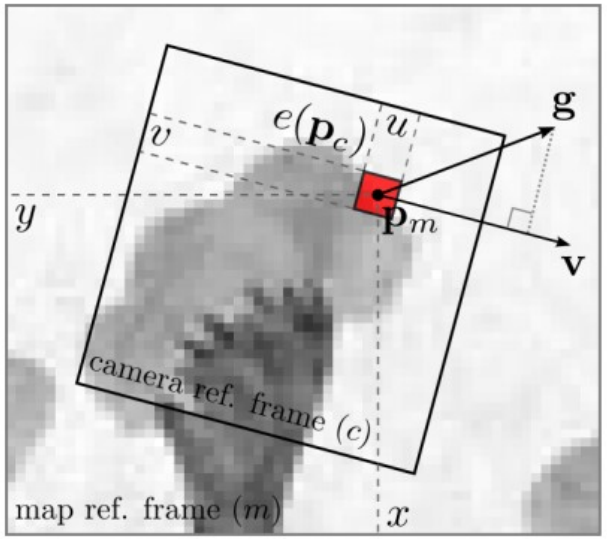
- **Mapping - Gradient estimation based on pixel-wise EKF**

Two-step process:

1) Process equation

Define the gradient vector as state vector

Apply the constant gradient process model



2) Measurement equation

$$\left(\mathbf{g}^{(t)} \cdot \mathbf{v}^{(t)}\right) \tau_c = \pm C$$

Define the measurement z as the instantaneous event rate at this pixel, we have the measurement equation:

$$z^{(t)} = \frac{1}{\tau_c}$$

$$h^{(t)} = \frac{\mathbf{g}^{(t)} \cdot \mathbf{v}^{(t)}}{C}$$

Simultaneous Mosaicing and Tracking with an Event Camera (bmva.org)

Paper: Simultaneous Mosaicing and Tracking with an Event Camera

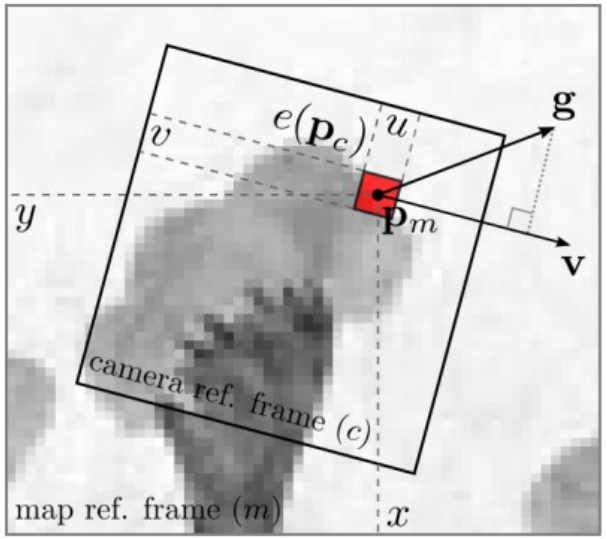
- **Mapping - Gradient estimation based on pixel-wise EKF**

Two-step process:

2) Measurement equation

How to find the velocity of the camera at a pixel $\mathbf{p}_m$?

$$\mathbf{v}^{(t)} = \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \frac{\mathbf{p}_m^{(t)} - \mathbf{p}_m^{(t-\tau_c)}}{\tau_c}$$



We can now iterate to estimate the pixel gradient based on EKF process.

Simultaneous Mosaicing and Tracking with an Event Camera (bmva.org)

Paper: Simultaneous Mosaicing and Tracking with an Event Camera

- **Mapping - Reconstruction from gradients**

Since we already obtained the gradients, how to reconstruct the intensity?

By optimization! Design the cost function as:

$$J(\mathrm{M}) = \int\int (M_x - g_x)^2 + (M_y - g_y)^2 dxdy \ .$$

where M_x, M_y are **true gradients** in x and y directions (we don't have);

g_x, g_y are **estimated gradients** in x and y directions (we have);

M denotes the intensity that we aim to solve.

Minimizing $J(M)$ leads to the well known Poisson equation:

$$\nabla^2 \mathrm{M} = \frac{\partial}{\partial x} g_x + \frac{\partial}{\partial y} g_y \ .$$

where $\nabla^2 \mathrm{M} = \frac{\partial^2 \mathrm{M}}{\partial x^2} + \frac{\partial^2 \mathrm{M}}{\partial y^2}$ is the Laplacian.

Fortunately, many methods are available to solve the above Poisson equation!

Simultaneous Mosaicing and Tracking with an Event Camera (bmva.org)

Paper: Simultaneous Mosaicing and Tracking with an Event Camera

Simultaneous Mosaicing and Tracking with an Event Camera

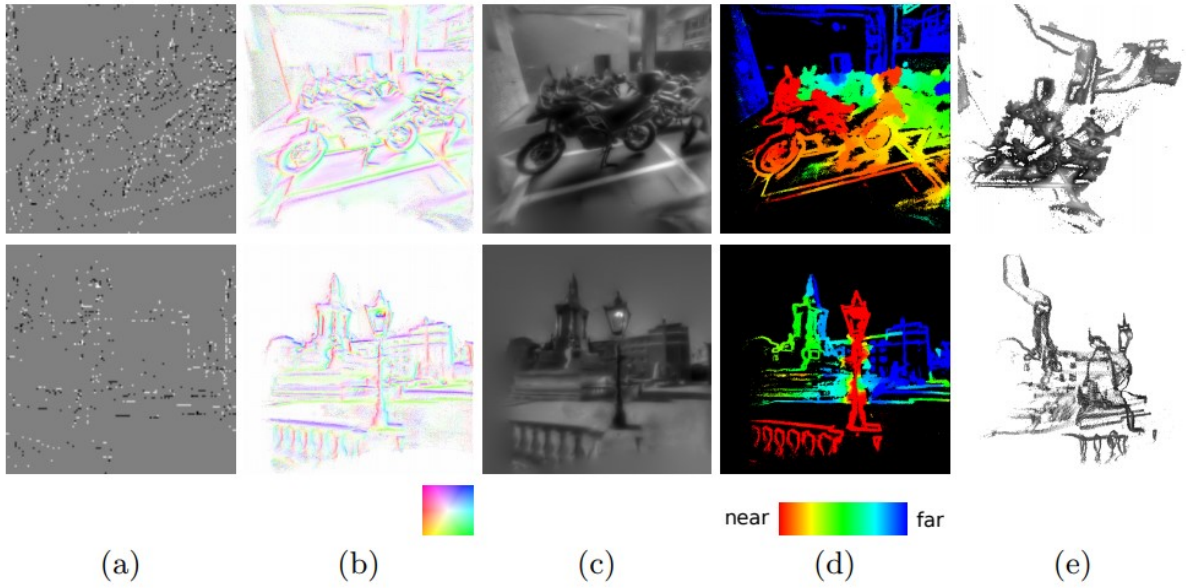
Hanme Kim [1], Ankur Handa [2], Ryad Benosman [3],
Sio-Hoi Ieng [3], Andrew J. Davison [1]

[1] Imperial College London
[2] University of Cambridge
[3] UPMC Univ Paris 06

Simultaneous Mosaicing and Tracking with an Event Camera (bmva.org)

Paper: Simultaneous Mosaicing and Tracking with an Event Camera

- The authors have expanded their work into 6-DoF pose estimation
  - Based on 3 decoupled probabilistic filters, estimating 6-DoF **camera motion**, scene log **intensity gradient**, and scene **inverse depth**, respectively.



(a) visualisation of the input event stream; (b) estimated gradient keyframes; (c) reconstructed intensity keyframes with super resolution and high dynamic range properties; (d) estimated depth maps; (e) semi-dense 3D point clouds.

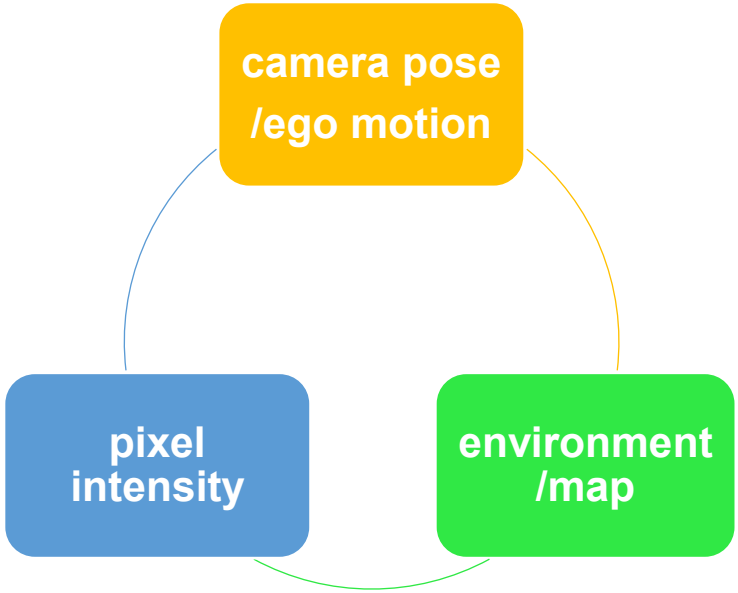
RESEARCH - Hanme Kim
kim_etal_eccv2016.pdf (ic.ac.uk)

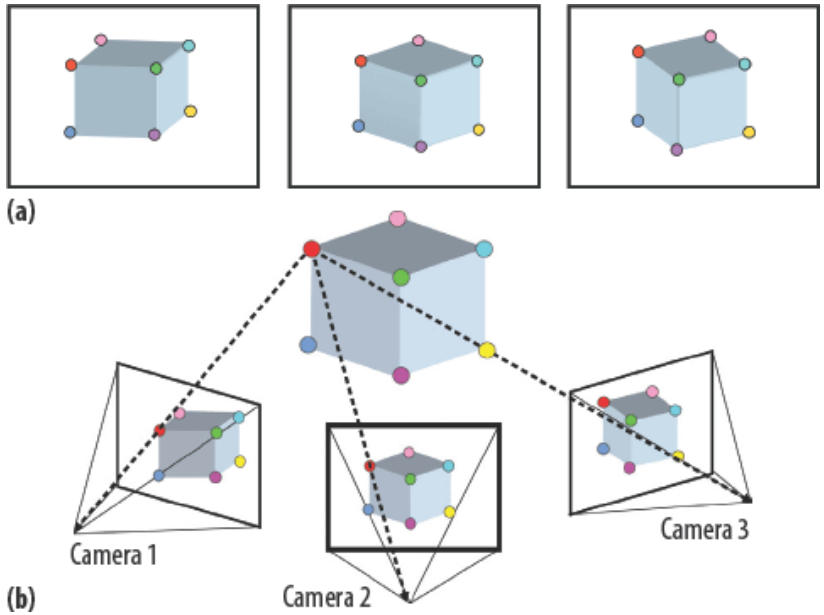
Summary – Pose estimation and SLAM

- Problem formulation
- Challenges for SLAM using conventional cameras
- Two main streams for event-based SLAM
  - Taking events as frames (frame-based)
  - Taking events as point clouds (pure event-based)
- Visual odometry (VO) - the core in a SLAM system
- Examples and related work

Image Reconstruction

- Close related to pose estimation, SLAM, and optical flow estimation.





- Here we define the **image reconstruction problem** as:
  - Estimate the **intensity** of each pixel on images
  - **Without explicitly** estimating camera's pose or environment

Usually focusing on image quality instead of the camera motion / environment modelling.

Seeing in 3D | hypraptive

Image Reconstruction

- Why brings event cameras into image reconstruction?
  - High speed imaging

  - High dynamic range imaging

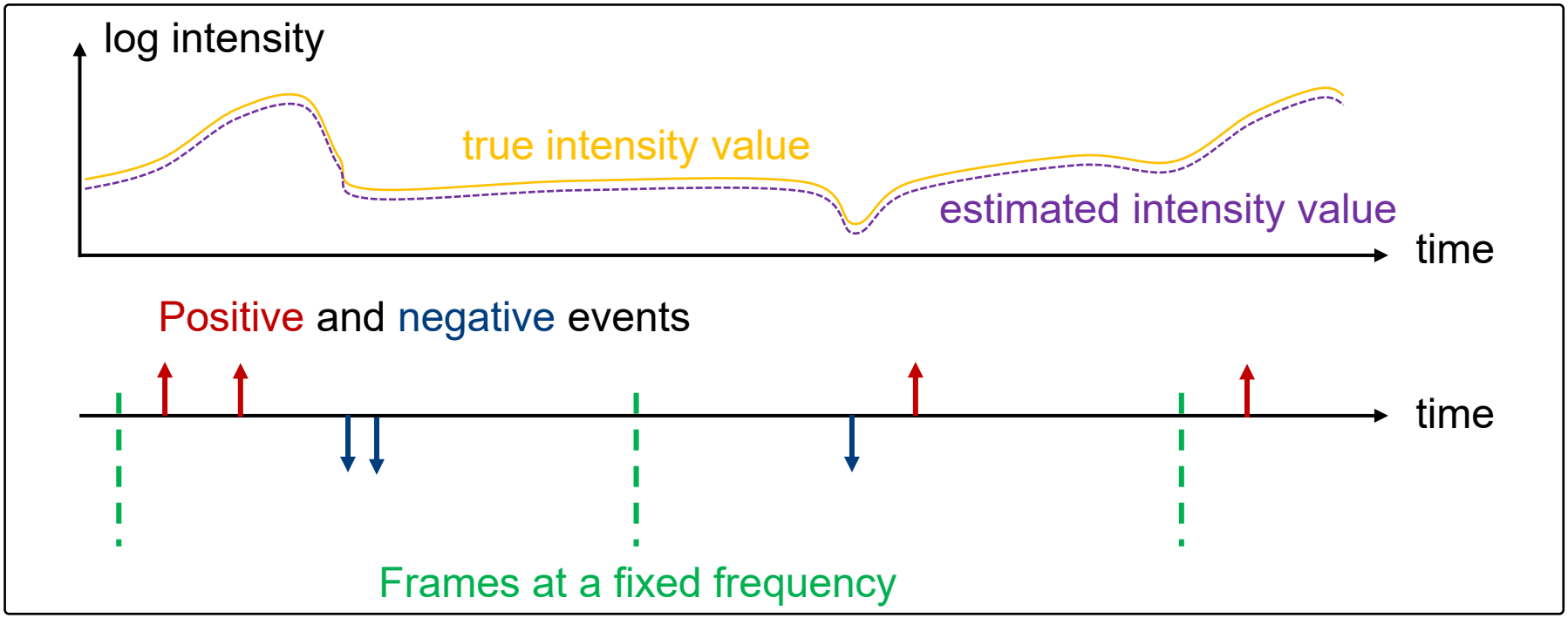
  - High resolution imaging
    by reconstructing sub-pixel intensity.

paper066.pdf (bmva.org)

Image Reconstruction

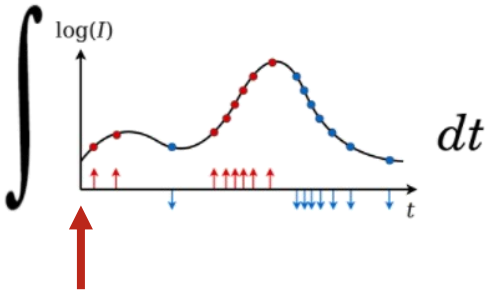
- Ideas
  - Pure event-based reconstruction
  - **Frame + event fusion**

Single-pixel frame + event fusion as a **state estimation problem**



Paper: Continuous-time Intensity Estimation Using Event Cameras

- A naïve and simple idea would be pure integration



dt

frame provides absolute intensity here

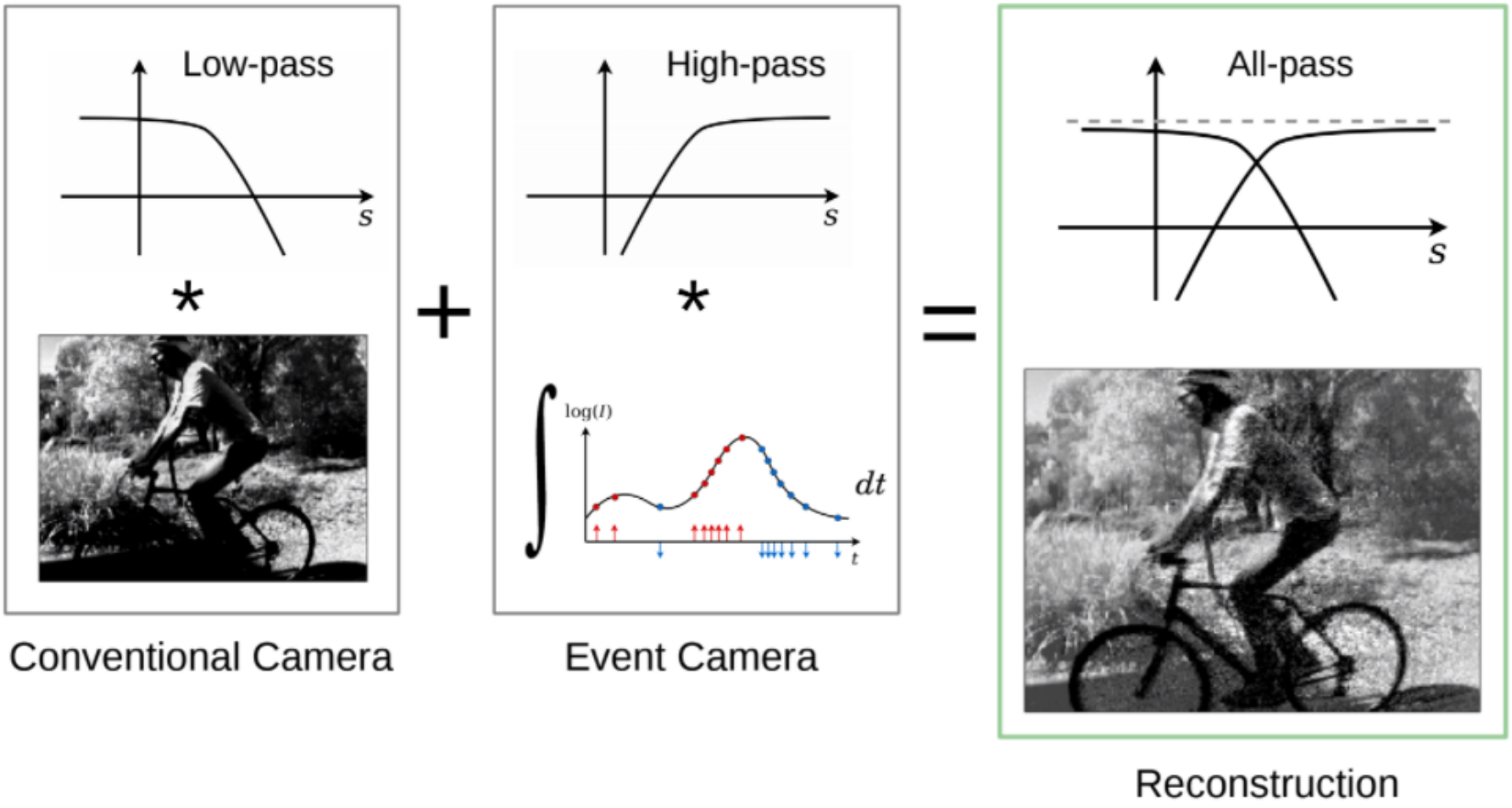
- Note that:
  - An event camera essentially outputs **high-frequency information**.
  - A conventional camera outputs **low-frequency information**.

Can we fuse low-frequency information from frames
with high-frequency information from events? And how?

https://arxiv.org/pdf/1811.00386.pdf
Continuous-time Intensity Estimation Using Event Cameras - Cedric Scheerlinck

Paper: Continuous-time Intensity Estimation Using Event Cameras

- The complementary filter

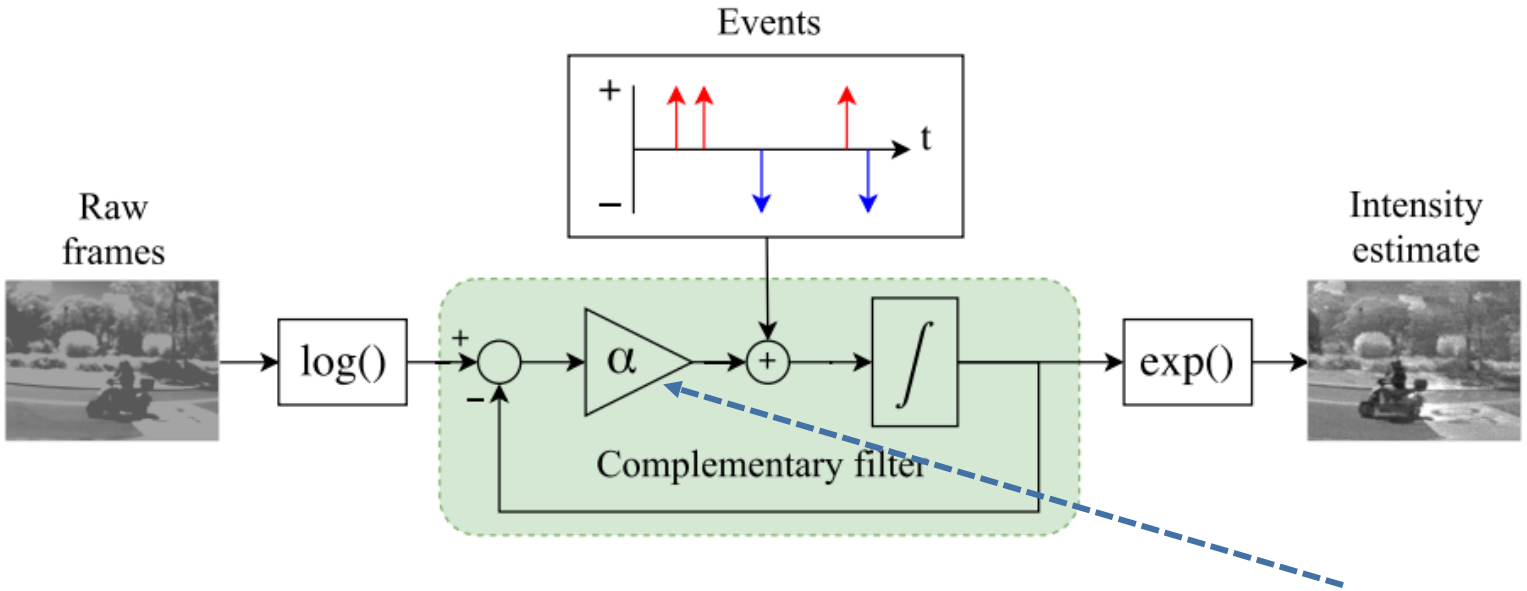


lecture_15.pdf (mit.edu)
https://arxiv.org/pdf/1811.00386.pdf
Continuous-time Intensity Estimation Using Event Cameras - Cedric Scheerlinck

Paper: Continuous-time Intensity Estimation Using Event Cameras

https://arxiv.org/pdf/1811.00386.pdf
Continuous-time Intensity Estimation Using Event Cameras - Cedric Scheerlinck

Paper: An Asynchronous Kalman Filter for Hybrid Event Cameras



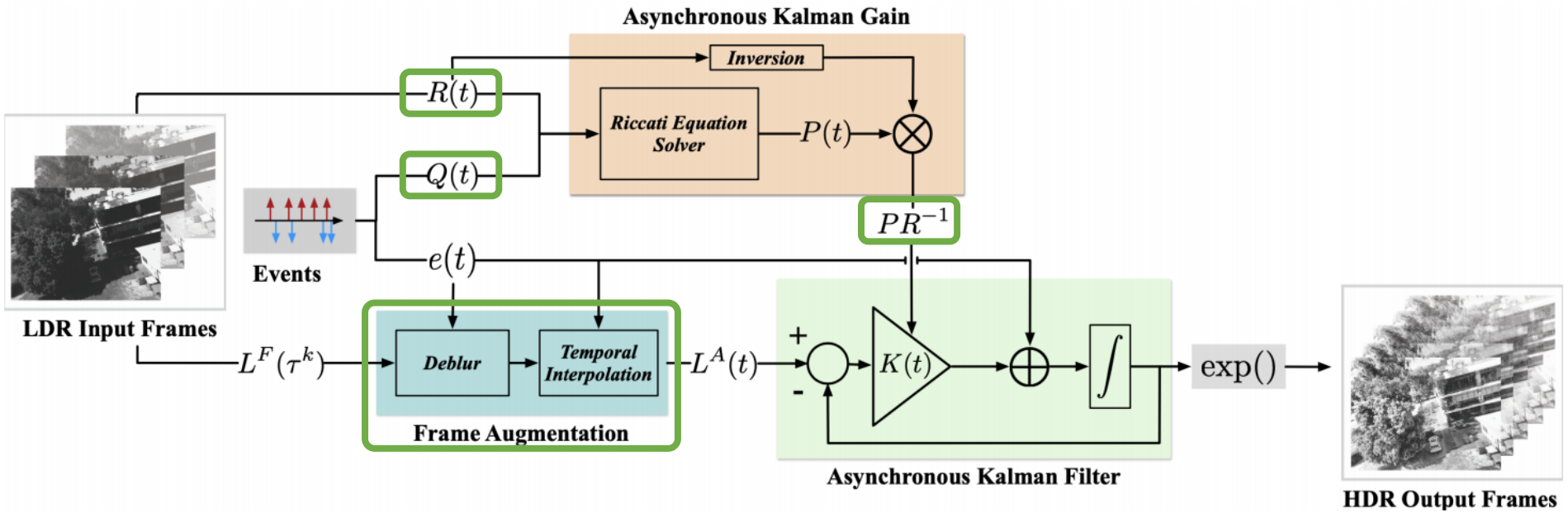
- A Complementary Filter has been proposed. There is a filter parameter that requires manual selection. Practical issues:
  - Hard to decide the parameter for different scenarios.
  - Not adaptive to scene changes.

Is it possible to set the filter parameter automatically?

An Asynchronous Kalman Filter for Hybrid Event Cameras - Cedric Scheerlinck

Paper: An Asynchronous Kalman Filter for Hybrid Event Cameras

- Framework



- Contributions
  - Varying and adaptive filter gain
  - EVS + CIS uncertainty models
  - Deblur and temporal interpolation for frame augmentation

An Asynchronous Kalman Filter for Hybrid Event Cameras - Cedric Scheerlinck

Paper: An Asynchronous Kalman Filter for Hybrid Event Cameras

- Event Camera (Noise) Model

$$e_{\boldsymbol{p}}(t) = \sum_{i=1}^{\infty} (c\sigma_{\boldsymbol{p}}^{i} + \eta_{\boldsymbol{p}}^{i})\delta(t - t_{\boldsymbol{p}}^{i})$$

$$\eta_{\boldsymbol{p}}^{i} \sim \mathcal{N}\left(0, Q_{\boldsymbol{p}}(t)\right)$$ Noise term

$$Q_{\boldsymbol{p}}(t) := \sum_{i=1}^{\infty} \left(Q_{\boldsymbol{p}}^{\text{proc.}}(t) + Q_{\boldsymbol{p}}^{\text{iso.}}(t) + Q_{\boldsymbol{p}}^{\text{ref.}}(t)\right) \delta(t - t_{\boldsymbol{p}}^{i})$$

Process noise

Isolated pixel noise

Refractory period noise

An Asynchronous Kalman Filter for Hybrid Event Cameras - Cedric Scheerlinck

Paper: An Asynchronous Kalman Filter for Hybrid Event Cameras

- Event Camera (Noise) Model
  - Process noise

$$Q_{\boldsymbol{p}}^{\text{proc.}}(t_{\boldsymbol{p}}^i) = \sigma_{\text{proc.}}^2 (t_{\boldsymbol{p}}^i - t_{\boldsymbol{p}}^{i-1})$$

  - Isolated pixel noise

$$Q_{\boldsymbol{p}}^{\text{iso.}}(t_{\boldsymbol{p}}^i) = \sigma_{\text{iso.}}^2 \min\{t_{\boldsymbol{p}}^i - t_{N(\boldsymbol{p})}^*\}$$

timestamp of neighborhood events

  - Refractory period noise

$$Q_{\boldsymbol{p}}^{\text{ref.}}(t_{\boldsymbol{p}}^i) = \begin{cases} 0 & \text{if } t_{\boldsymbol{p}}^i - t_{\boldsymbol{p}}^{i-1} > \rho \\ \sigma_{\text{ref.}}^2 & \text{if } t_{\boldsymbol{p}}^i - t_{\boldsymbol{p}}^{i-1} \leq \bar{\rho} \end{cases}$$

refractory time

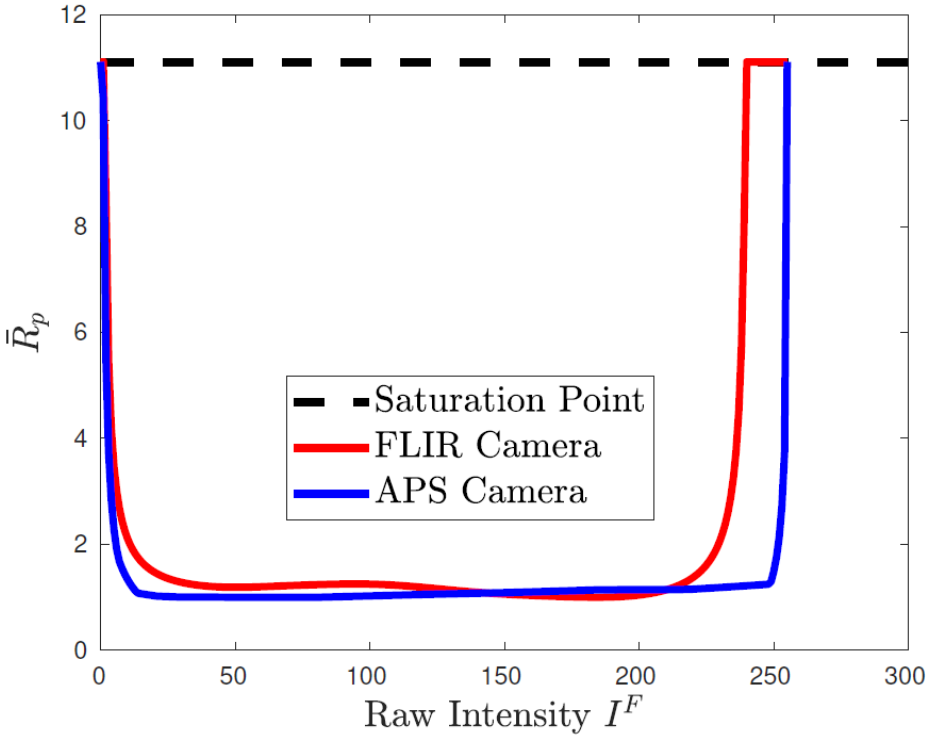
An Asynchronous Kalman Filter for Hybrid Event Cameras - Cedric Scheerlinck

Paper: An Asynchronous Kalman Filter for Hybrid Event Cameras

- Conventional Camera (Noise) Model
  - Gaussian assumption, covariance related to output intensity:

    Over-exposed/under-exposed areas → Larger covariance

    Normally-exposed areas → Smaller covariance



An Asynchronous Kalman Filter for Hybrid Event Cameras - Cedric Scheerlinck

Paper: An Asynchronous Kalman Filter for Hybrid Event Cameras

An Asynchronous Kalman Filter for Hybrid Event Cameras

Ziwei Wang, Yonhon Ng, Cedric Scheerlinck, Robert Mahony

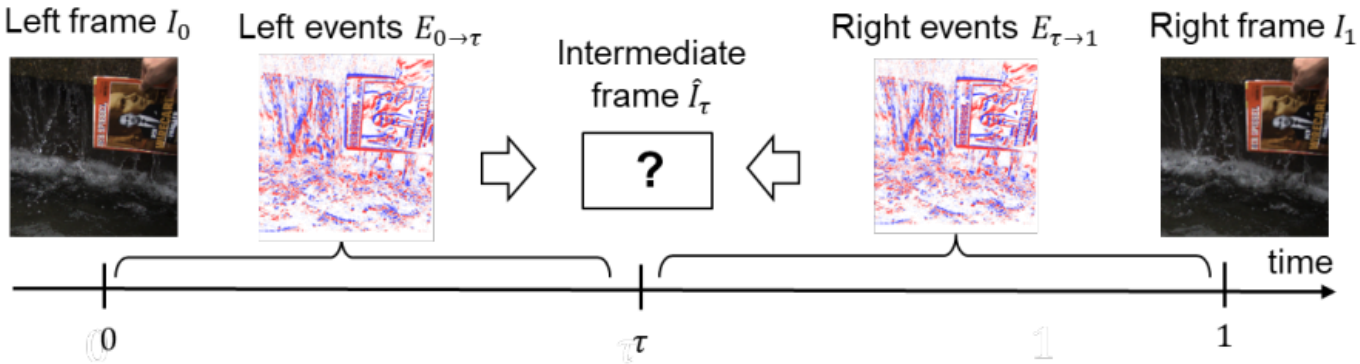
Australian National University

Systems Theory & Robotics Group

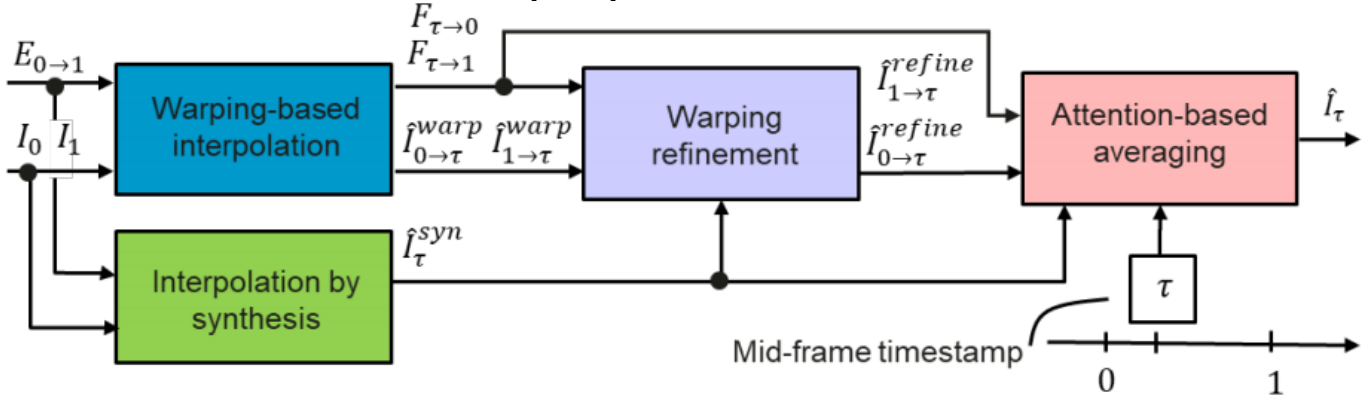
An Asynchronous Kalman Filter for Hybrid Event Cameras - Cedric Scheerlinck

Paper: TimeLens: Event-based Video Frame Interpolation

- Deep learning-based approach.
- Proposed event-based interpolation approach utilizes **events and frames from two directions**.



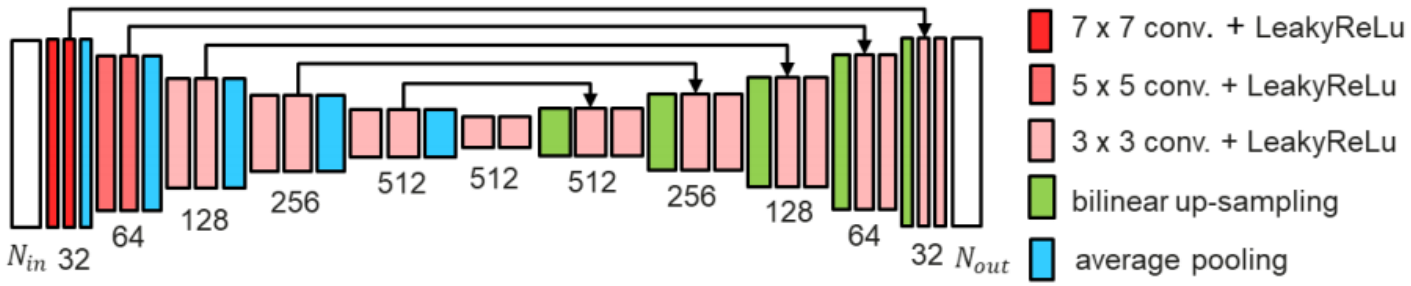
- Overview framework of the proposed method:



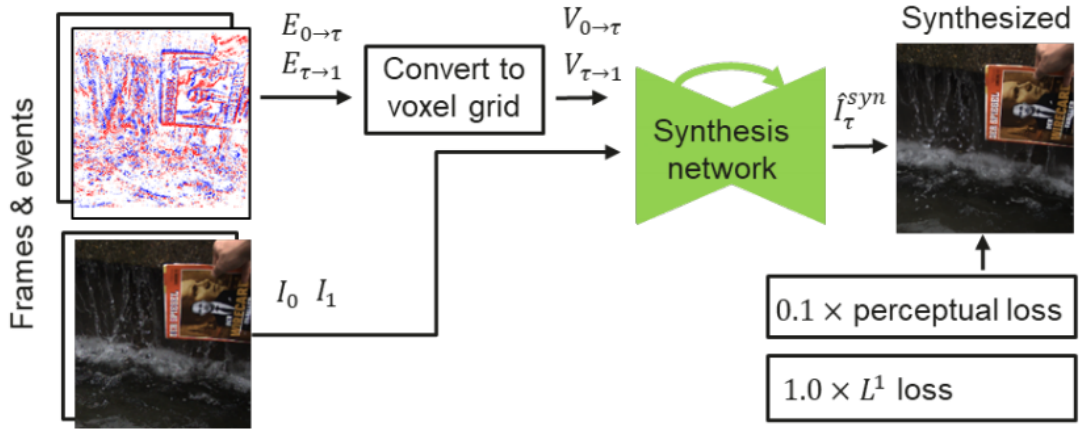
CVPR21_Gehrig.pdf (uzh.ch)
Time Lens: Event-based Video Frame Interpolation (uzh.ch)

Paper: TimeLens: Event-based Video Frame Interpolation

- Backbone network architecture: **hourglass network**



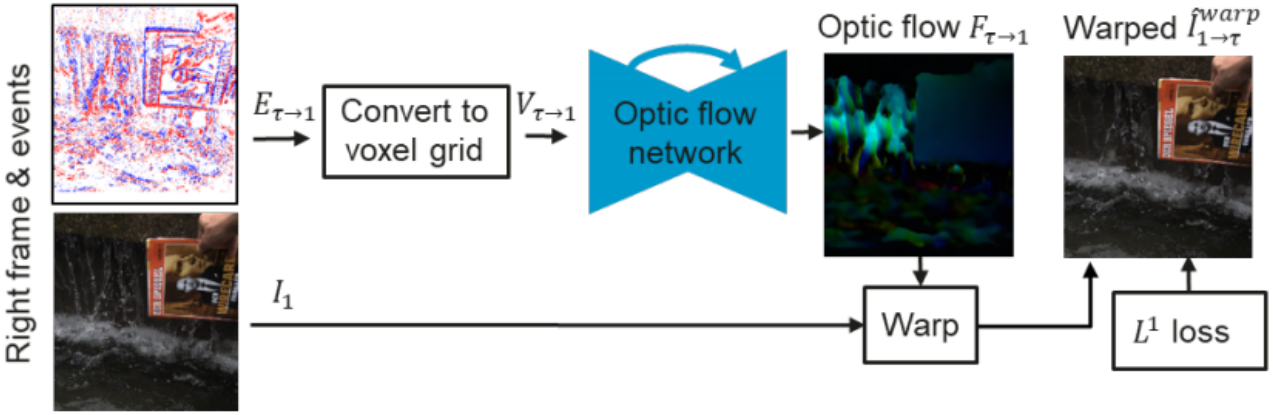
- Interpolation by synthesis module



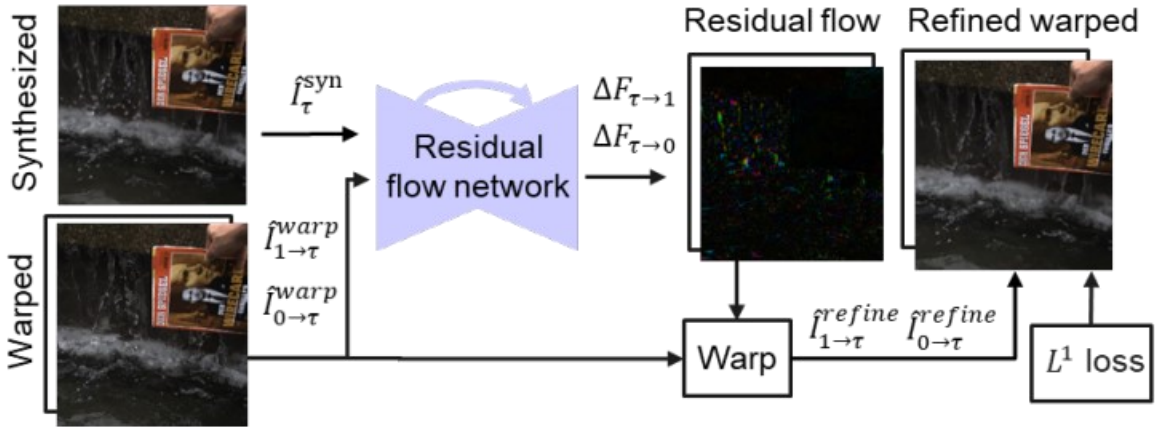
CVPR21_Gehrig.pdf (uzh.ch)
Time Lens: Event-based Video Frame Interpolation (uzh.ch)

Paper: TimeLens: Event-based Video Frame Interpolation

- Warping-based interpolation module



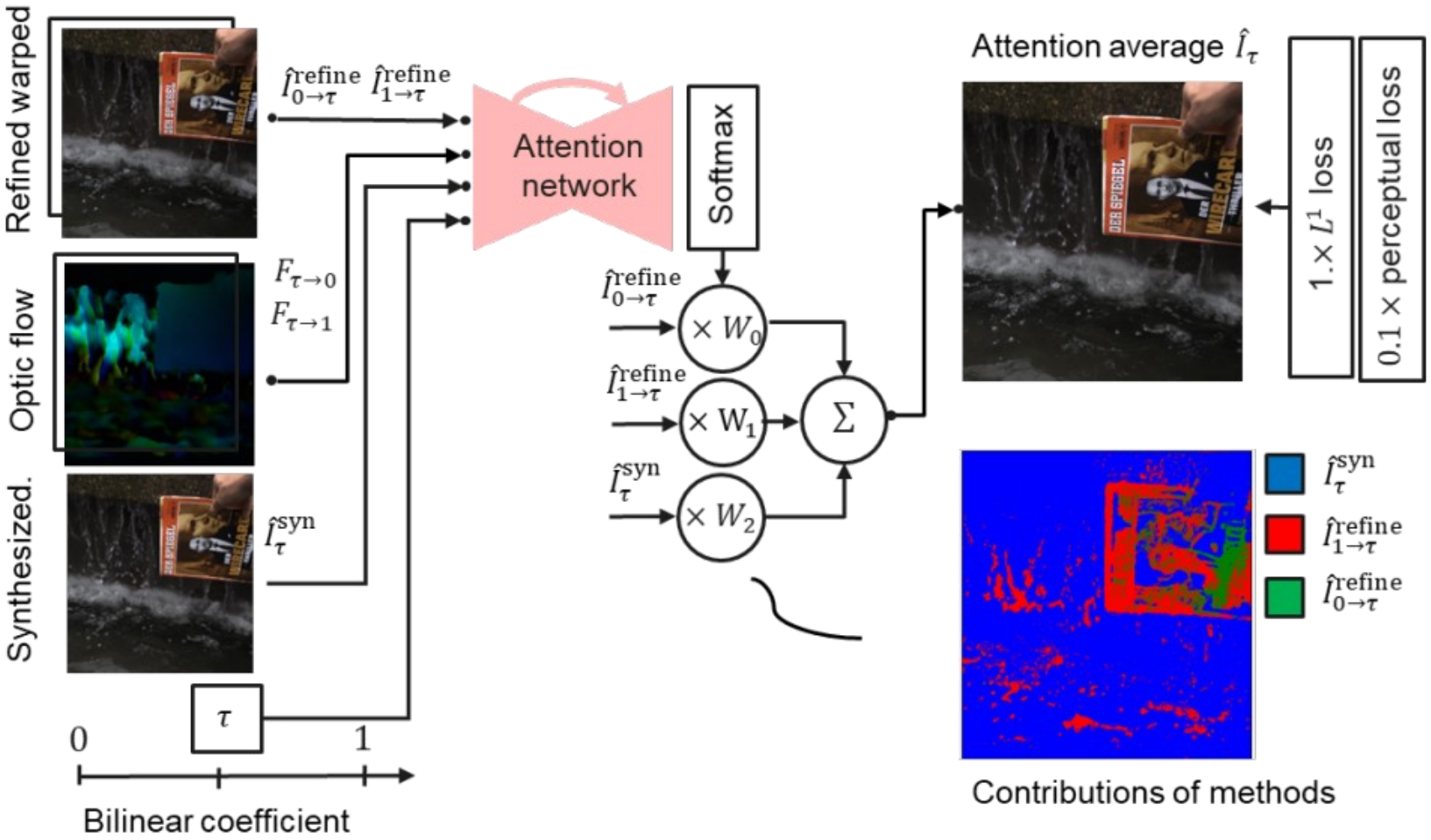
- Warping refinement module



CVPR21_Gehrig.pdf (uzh.ch)
Time Lens: Event-based Video Frame Interpolation (uzh.ch)

Paper: TimeLens: Event-based Video Frame Interpolation

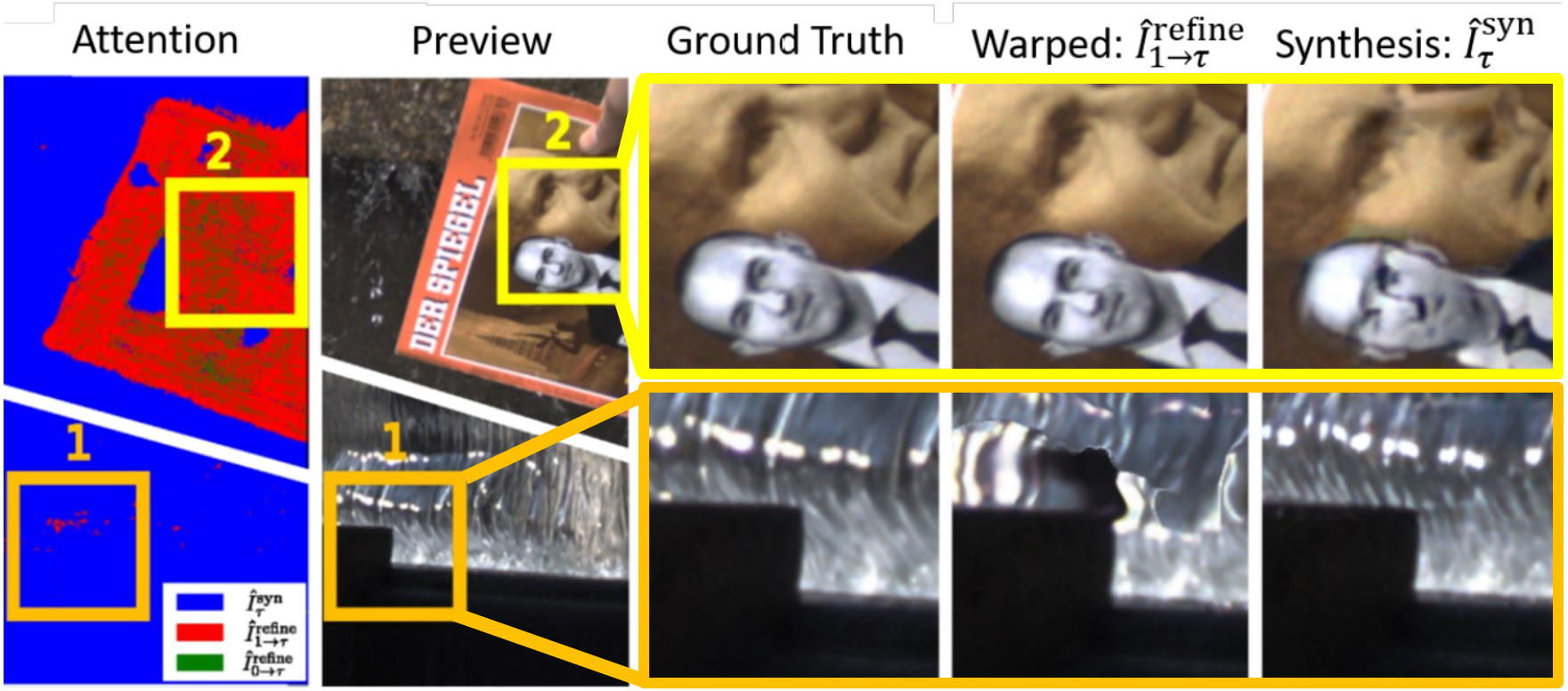
- Attention-based averaging module



CVPR21_Gehrig.pdf (uzh.ch)
Time Lens: Event-based Video Frame Interpolation (uzh.ch)

Paper: TimeLens: Event-based Video Frame Interpolation

- Complementarity of warping- and synthesis-based interpolation.
  - The attention network learns where to trust synthesized interpolation results, and where to trust warped interpolation results.



CVPR21_Gehrig.pdf (uzh.ch)
Time Lens: Event-based Video Frame Interpolation (uzh.ch)

Paper: TimeLens: Event-based Video Frame Interpolation

CVPR21_Gehrig.pdf (uzh.ch)
Time Lens: Event-based Video Frame Interpolation (uzh.ch)

Paper: Back to Event Basics: Self-Supervised Learning of Image Reconstruction for Event Cameras via Photometric Constancy

- One issue for learning-based methods: **lack of data.** Possible reasons:
  - The **cost** of event sensor is high;
  - The **data format** of event sensor is not unified.
  - …?
- How to the reduce the dependence of data in learning-based methods?
  - Self-supervised learning/unsupervised learning.
    e.g., EV-FlowNet for optical flow estimation:
    No ground-truth optical flow is used in training; Ground-truth CIS images are required.

Can we train a network using events only?

[2009.08283] Back to Event Basics: Self-Supervised Learning of Image Reconstruction for Event Cameras via Photometric Constancy (arxiv.org)

Paper: Back to Event Basics: Self-Supervised Learning of Image Reconstruction for Event Cameras via Photometric Constancy

Back to Event Basics: Self-Supervised Learning of Image Reconstruction for Event Cameras via Photometric Constancy

Federico Paredes-Valles and Guido C. H. E. de Croon

(Poster Session Three, ID: 8305)

MAVLab™

Code and models: mavlab.tudelft.nl/ssl_e2v/

TU Delft Delft University of Technology

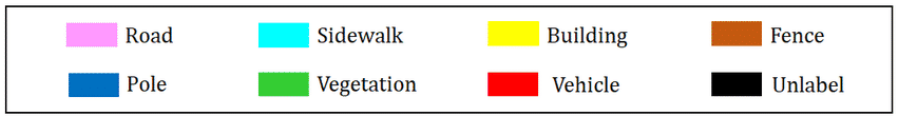
[2009.08283] Back to Event Basics: Self-Supervised Learning of Image Reconstruction for Event Cameras via Photometric Constancy (arxiv.org)

Summary – Image reconstruction

- Image reconstruction: closely related to pose estimation, SLAM, and optical flow estimation.
  - Modelling-based approaches
  - Learning-based approaches

Motion Segmentation

- **Image segmentation** is
  - the process of partitioning a digital image into multiple segments.
  - the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.



(4) (PDF) Towards a Meaningful 3D Map Using a 3D Lidar and a Camera (researchgate.net)
Deploying robotics applications — Using image segmentation to drive the Turtlebot3 Burger - Qualcomm Developer Network

Motion Segmentation

- Image segmentation with an event camera
  - Idea 1: **fuse events and frames:** combine motion patterns and texture/geometry patterns
  - Idea 2: use **events only:** segmenting objects based on motion patterns → **"motion segmentation"**

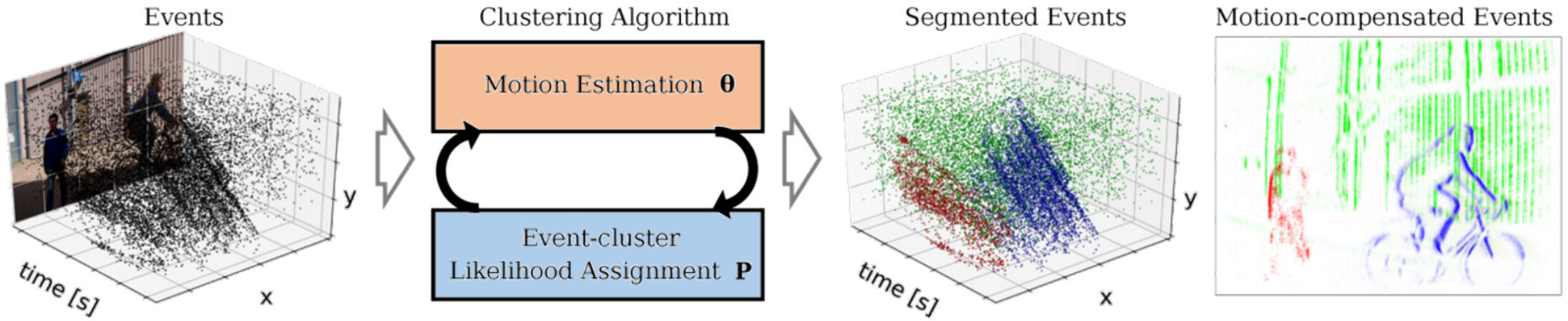
(4) (PDF) Towards a Meaningful 3D Map Using a 3D Lidar and a Camera (researchgate.net)

Paper:
Event-Based Motion Segmentation by Motion Compensation

- Framework of motion segmentation

Idea: maximizing event alignment, i.e., maximizing the sharpness of motion compensated images (one per cluster) of warped events.

Cluster	C1	C2
motion (u, v)	(-400, 2)	(500, -5)



Event e	P($e \in$ C1)	P($e \in$ C2)
(128,40,+,1)	0.75	0.25
(20,600,-,2)	0.30	0.70
(128,41,+,3)	0.80	0.20
…	…	…

rpg.ifi.uzh.ch/docs/ICCV19_Stoffregen.pdf

Paper:
Event-Based Motion Segmentation by Motion Compensation

- Mathematical formulation
  - **Soft event-cluster associations:**

$$p_{kj} = P(e_k \in \ell_j)$$

is the probability of the *k*-th event belonging to the *j*-th cluster.

Let $\mathbf{P} \equiv (p_{kj})$ be an $N_e \times N_\ell$ matrix with all event-cluster associations. Define the weighted IWE (Image of Warped Events) of the *j*-th cluster as

$$I_j(\mathbf{x}) \doteq \sum_{k=1}^{N_e} p_{kj}\, \delta(\mathbf{x} - \mathbf{x}'_{kj}),$$

with $\mathbf{x}'_{kj} = \mathbf{W}(\mathbf{x}_k, t_k; \boldsymbol{\theta}_j)$ the warped event location, and δ the Dirac delta. Above equation states that events are warped:

$$e_k \doteq (\mathbf{x}_k, t_k, s_k) \mapsto e'_k \doteq (\mathbf{x}'_k, t_{\text{ref}}, s_k)$$

and probabilities are accumulated at the warped locations.

rpg.ifi.uzh.ch/docs/ICCV19_Stoffregen.pdf

Paper:
Event-Based Motion Segmentation by Motion Compensation

We want to maximize the sharpness, but how to define the sharpness?

- Define the sharpness as the variance of the image:

$$\mathrm{Var}(I_j) \doteq \frac{1}{|\Omega|}\int_{\Omega}(I_j(\mathbf{x}) - \mu_{I_j})^2 d\mathbf{x}$$

where μ_{I_j} is the mean of IWE over the image plane Ω.

Problem: to find the associations **P** and cluster parameters $\boldsymbol{\theta}$ that maximize the sum of contrasts of all clusters:

$$(\boldsymbol{\theta}^*, \mathbf{P}^*) = \arg\max_{(\boldsymbol{\theta},\mathbf{P})} \sum_{j=1}^{N_\ell} \mathrm{Var}(I_j)$$

No closed-form solution → Only iterative, alternating optimization!

rpg.ifi.uzh.ch/docs/ICCV19_Stoffregen.pdf

Paper:
Event-Based Motion Segmentation by Motion Compensation

- Alternating Optimization
  - Each iteration contains 2 steps

The proposed alternating method **converges locally** (i.e., there is no guarantee of convergence to a global solution).

Algorithm 1 Event-based Motion Segmentation

1: **Input**: events $\mathcal{E} = \{e_k\}_{k=1}^{N_e}$ in a space-time volume V of the image plane, and number of clusters N_ℓ.
2: **Output**: cluster parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_j\}_{j=1}^{N_\ell}$ and event-cluster assignments $\mathbf{P} \equiv p_{kj} \doteq P(e_k \in \ell_j)$.
3: **Procedure:**
4: Initialize the unknowns $(\boldsymbol{\theta}, \mathbf{P})$ (see Section 3.5).
5: **Iterate** until convergence:
6: • Compute the event-cluster assignments p_{kj} using (7).
7: • Update the motion parameters of all clusters (6).

If associations **P** are fixed, update the motion parameters using

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \mu \nabla_{\boldsymbol{\theta}} \left(\sum_{j=1}^{N_\ell} \mathrm{Var}(I_j) \right) \qquad (6)$$

"gradient ascent"

Fix motion parameters $\boldsymbol{\theta}$, refine the associations based on probability partitioning law:

$$p_{kj} = \frac{c_j(\mathbf{x}'_k(\boldsymbol{\theta}_j))}{\sum_{i=1}^{N_\ell} c_i(\mathbf{x}'_k(\boldsymbol{\theta}_i))} \qquad (7)$$

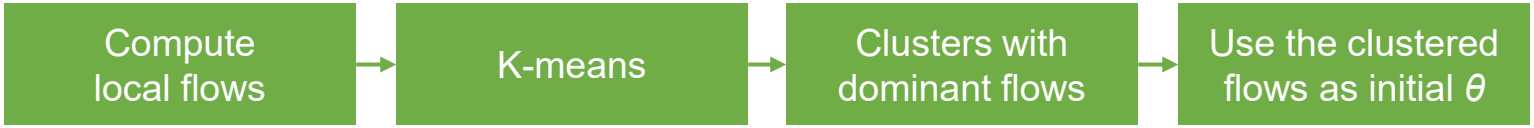
where $c_j(\mathbf{x}) \doteq I_j(\mathbf{x})$.

rpg.ifi.uzh.ch/docs/ICCV19_Stoffregen.pdf

Paper:
Event-Based Motion Segmentation by Motion Compensation

- Initialization
  - Requires initialization of $\boldsymbol{\theta}$, $\mathbf{P}$ to start the iteration.
  - Several initialization schemes are possible, depending on the motion models.

  1. K-means initialization for **optical flow-type** motion;



  2. Greedy initialization for **all kinds of motion**.
     No details/codes released from the authors.

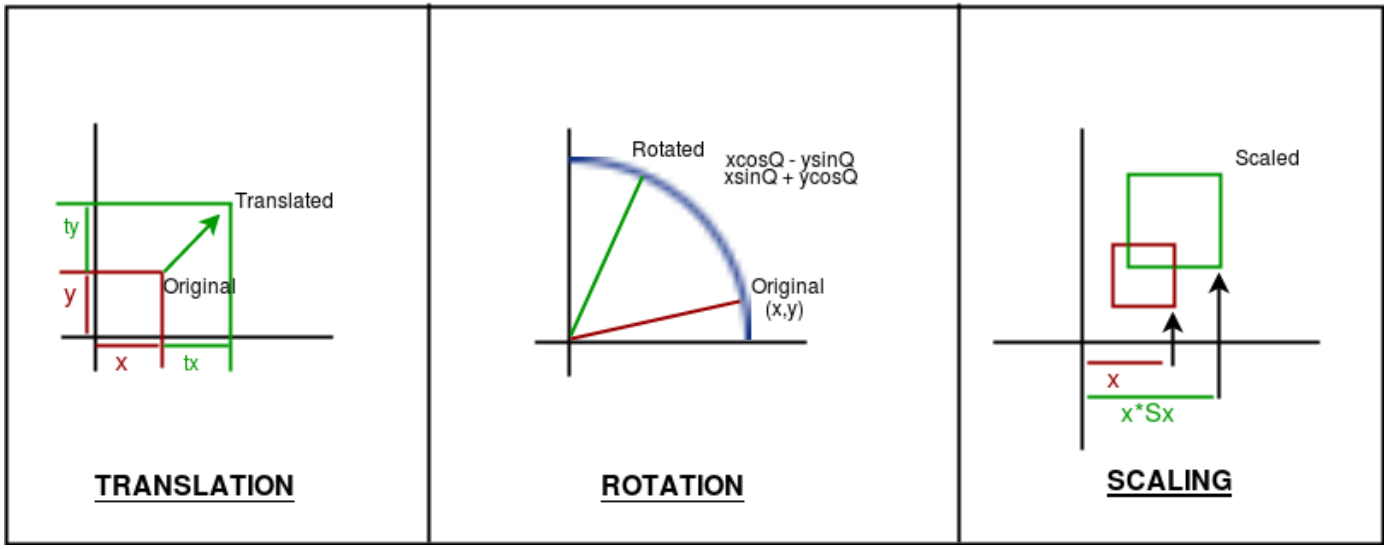
rpg.ifi.uzh.ch/docs/ICCV19_Stoffregen.pdf

Paper:
Event-Based Motion Segmentation by Motion Compensation

- Discussion on **motion model** $\mathbf{x}'_{kj} = \mathbf{W}(\mathbf{x}_k, t_k; \boldsymbol{\theta}_j)$

$$e_k \doteq (\mathbf{x}_k, t_k, s_k) \mapsto e'_k \doteq (\mathbf{x}'_k, t_{\text{ref}}, s_k)$$

  - Optical flow type motion (translation only)
  - Other types of motion (both translation and rotation)

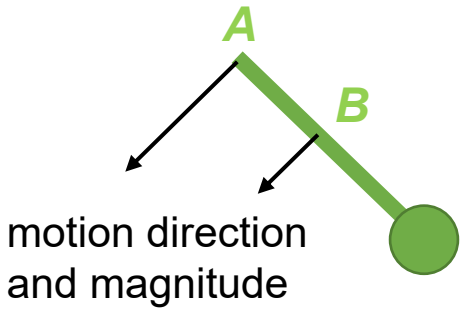


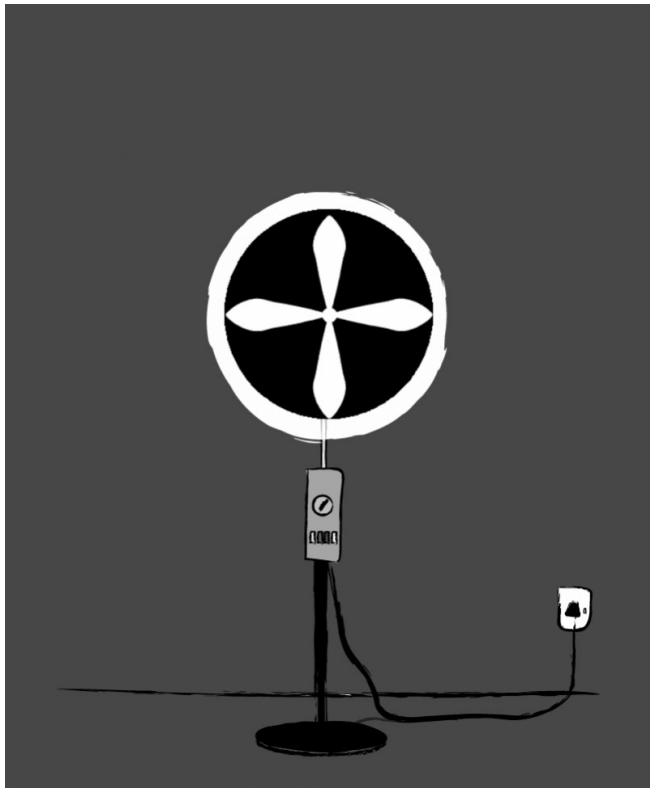
An appropriate motion model helps to **reduce the cluster number**!

Basic Transformations in OPENGL - GeeksforGeeks

Paper:
Event-Based Motion Segmentation by Motion Compensation

- Discussion on **motion model**
  Consider the rotating fan scenario.
  What motion model should be used?
- Do points A and B share the same motion parameter, if using an optical flow type model?
- What if using a rotation model instead?

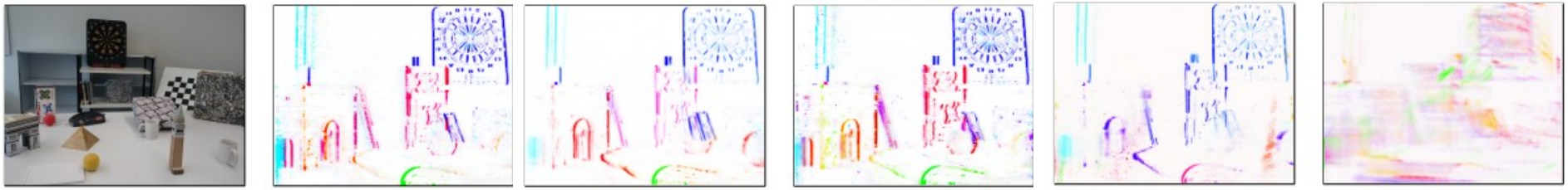




All points on the edge of blade have **different motion parameters**, if using an flow type model.
However, they do share the **same motion parameters**, if using a rotation model!

Paper:
Event-Based Motion Segmentation by Motion Compensation

- Discussion on **limit of object number**
  - Algo requires to pre-define number of clusters N_l;
  - It is noted that N_l is not a particularly important parameter;
  - If N_l too large, unnecessary clusters end up not having any events allocated to them and thus "die";
  - N_l can simply be chosen to be large.



(a) Scene (b) $N_\ell = 5\times$Optical Flow (c) $N_\ell = 10\times$Optical Flow (d) $N_\ell = 20\times$Optical Flow (e) $N_\ell = 5\times$OF$+5\times$Rotation (f) $N_\ell = 10\times$Rotation

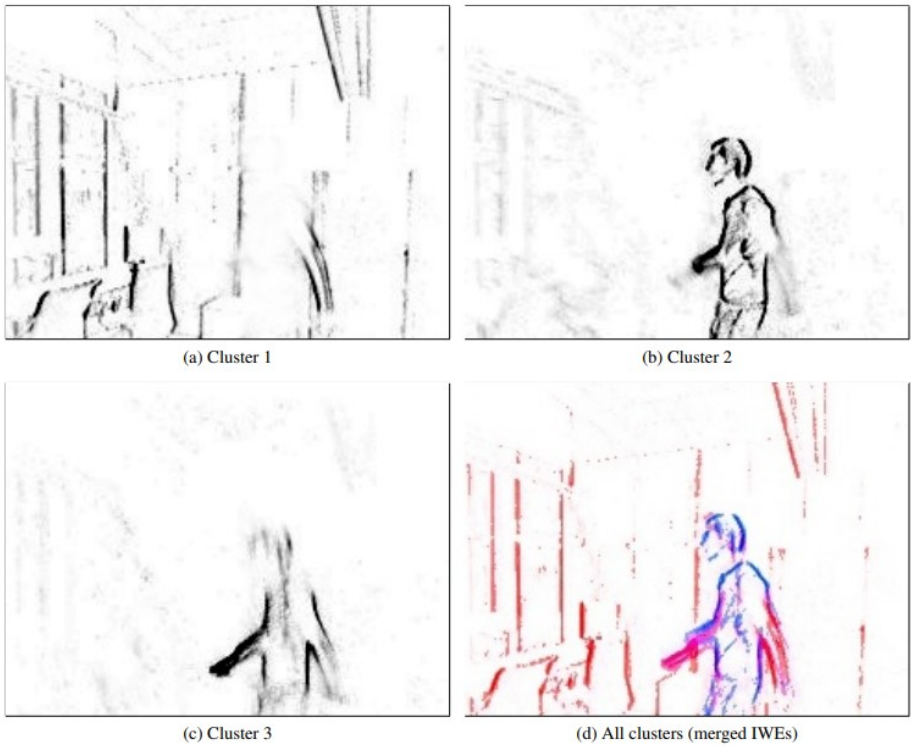
Experiment on *slider depth* sequence. Motion-compensated images in (b) to (f) show events colored by cluster.

The real motion for all clusters should be in "Optical Flow" type.

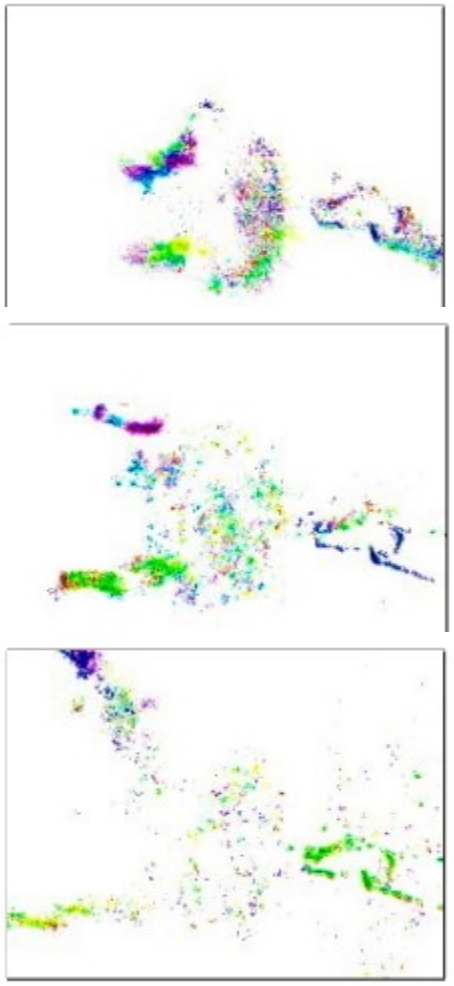
rpg.ifi.uzh.ch/docs/ICCV19_Stoffregen.pdf

Paper:
Event-Based Motion Segmentation by Motion Compensation

- Discussion on **non-rigid objects' motion**



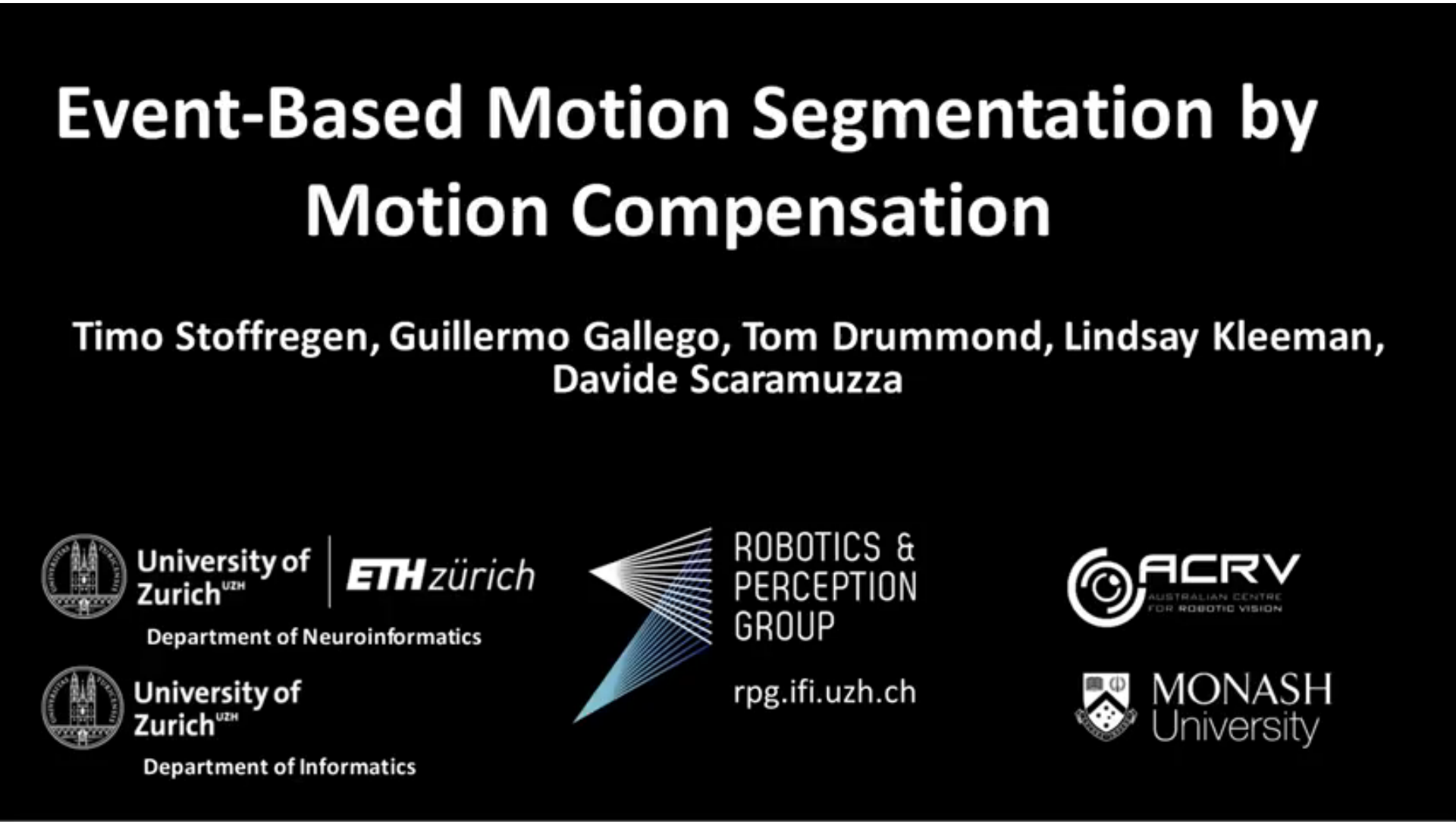
A person walks across a room, arms swinging.



Snapshots of segmentation of balloon popping.
Num. of clusters = 4,
Events colored by cluster membership.

rpg.ifi.uzh.ch/docs/ICCV19_Stoffregen.pdf

Paper:
Event-Based Motion Segmentation by Motion Compensation



rpg.ifi.uzh.ch/docs/ICCV19_Stoffregen.pdf

Summary – Motion segmentation

- Image segmentation and motion segmentation
- Problem formulation
- Ideas
  - Events + frame fusion-based
  - Pure event-based
- Selected papers
  - Motion segmentation towards **contrast maximization**

Supplementary Materials for Self-Learning

Paper: Simultaneous Estimation for Flow and Intensity

- Method: Sliding window **variational optimisation**
  - Idea on how to design the energy function:

$\vec{u}$ & L : functions $\vec{u}(\vec{x},t)$, $\vec{L}(\vec{x},t)$

joint estimation on flow & intensity

smoothness of the flow & intensities

$$\min_{\mathbf{u},L} \int_\Omega \int_T \Big(\lambda_1 \|\mathbf{u_x}\|_1 + \lambda_2 \|\mathbf{u}_t\|_1 + \lambda_3 \|L_\mathbf{x}\|_1 + \lambda_4 \|\langle L_\mathbf{x}, \delta_t \mathbf{u}\rangle + L_t\|_1 + \lambda_5 h_\theta(L - L(t_p)) \Big) \mathrm{d}t\, \mathrm{d}\mathbf{x}$$

$$\mathrm{grad}^t(I)\begin{pmatrix} v_x \\ v_y \end{pmatrix} + \frac{\partial I}{\partial t} = 0$$

brightness consistancy

time of the last event

$$+ \int_\Omega \sum_{i=2}^{|P(\mathbf{x})|} \|L(t_i) - L(t_{i-1}) - \theta\rho_i\|_1 \mathrm{d}\mathbf{x},$$

polarity

all events fired at $\vec{x}$

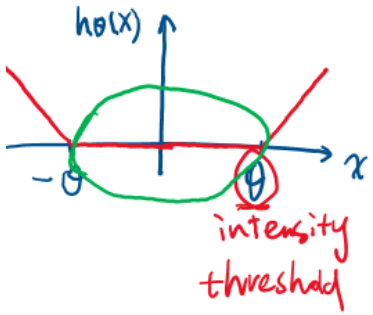
Bardow, Patrick, Andrew J. Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. CVPR 2016.

Paper: Simultaneous Estimation for Flow and Intensity

- The first 3 terms: regularise the smoothness of the flow (both spatially and temporally) and the smoothness of the intensities.
- The 4th term: first order Taylor approximation of the brightness consistency equation.
- The last 2 terms: the **no-event data term** and the **event data term**.

No-event data term:

$$\lambda_5 \, h_\theta(L - L(t_p))$$



$$h_\theta(x) = \begin{cases} |x| - \theta, & \text{if } |x| > \theta \\ 0, & \text{otherwise,} \end{cases}$$

Event data term:

$$\int_\Omega \sum_{i=2}^{|P(\mathbf{x})|} \|L(t_i) - L(t_{i-1}) - \theta\rho_i\|_1$$

We want event data to satisfy

$$|L(\mathbf{x}, t) - L(\mathbf{x}, t_p(\mathbf{x}, t))| \geq \theta$$

as much as possible.

Paper: Simultaneous Estimation for Flow and Intensity

Conference on Computer Vision and Pattern Recognition (CVPR 2016)

Simultaneous Optical Flow and Intensity Estimation from an Event Camera

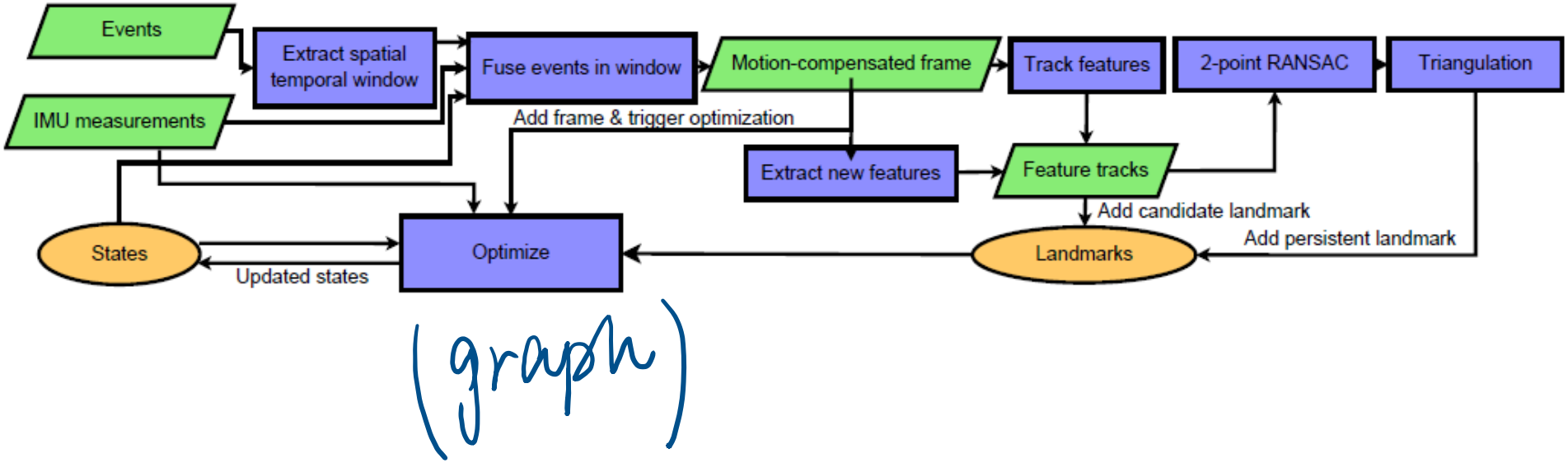
Patrick Bardow, Andrew Davison and Stefan Leutenegger

Dyson Robotics Laboratory,
Imperial College London

Bardow, Patrick, Andrew J. Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. CVPR 2016.

Paper: Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization

- Overview

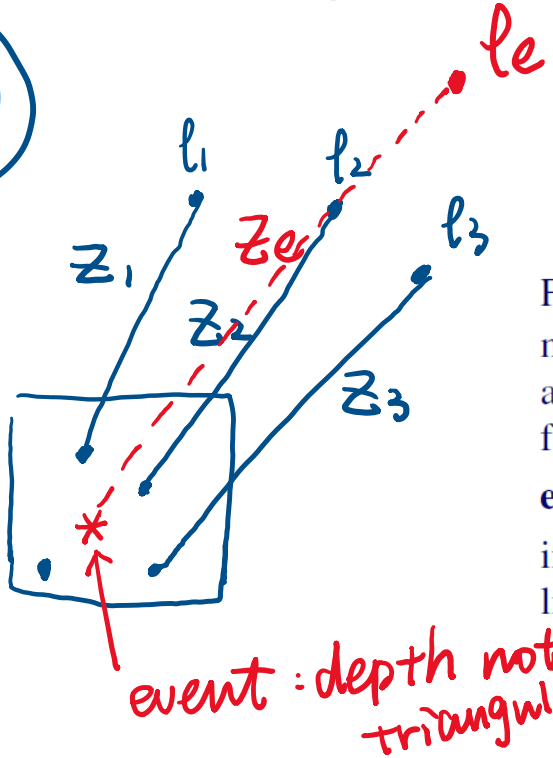


Paper: Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization

- Creating Motion-Corrected Event Frames

$$\mathbf{x}'_j = \pi\left(T_{t_k^f, t_j}\left(Z(\mathbf{x}_j)\pi^{-1}(\mathbf{x}_j)\right)\right)$$

To obtain $T_{t_k^f, t_j}$ and $Z(\mathbf{x}_j)$



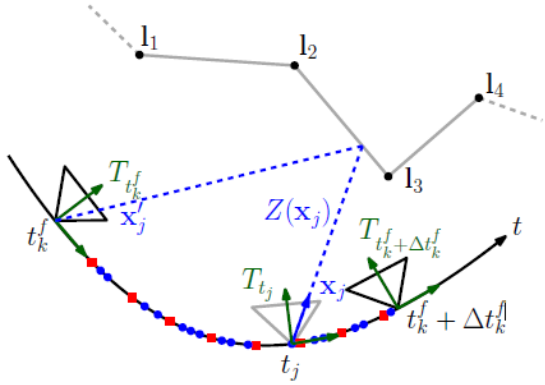
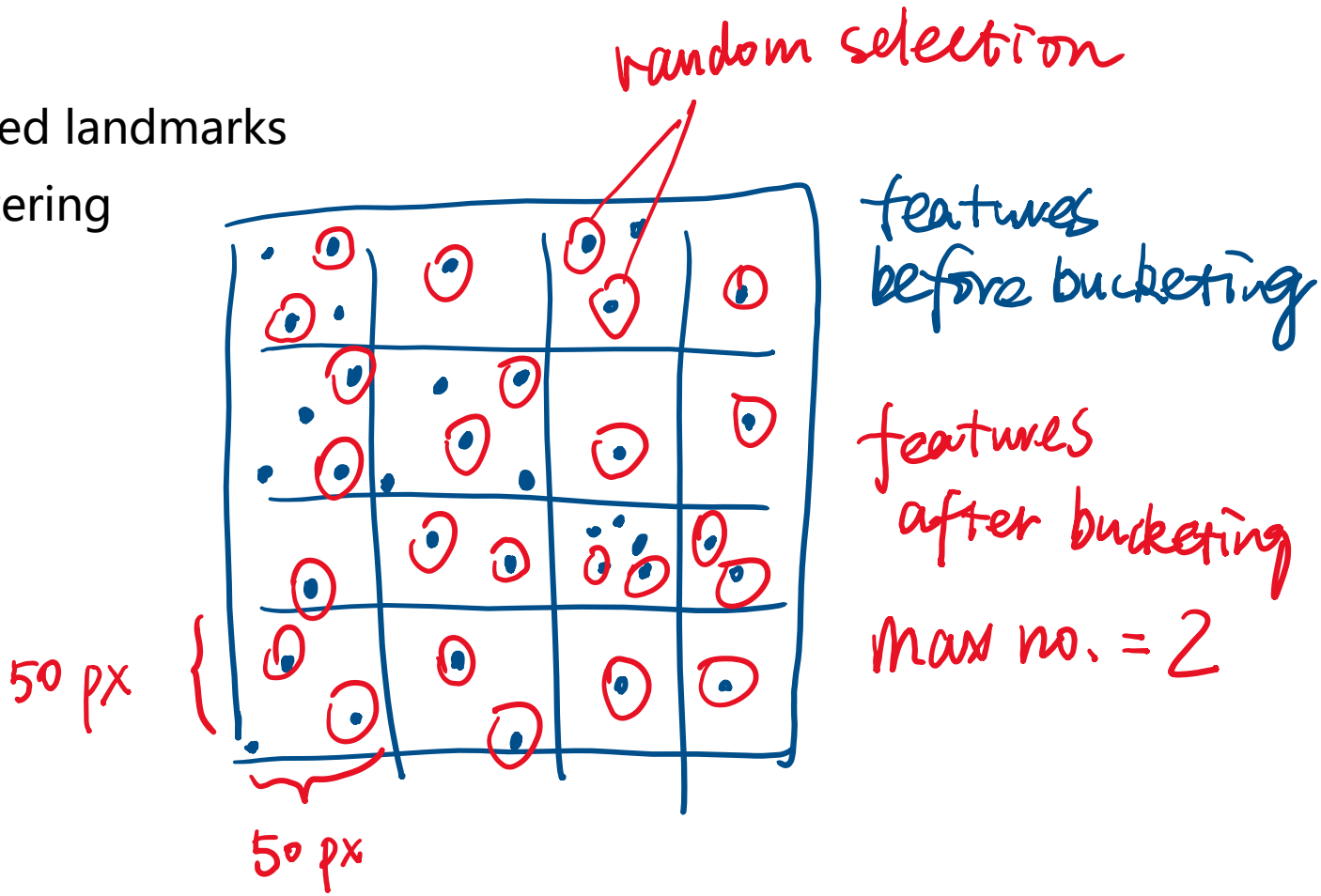


Figure 3: Motion correction: the inertial measurements in $[t_k^f, t_k^f + \Delta t_k^f]$ (red squares) are integrated to compute the relative transformation $T_{t_k^f, t_k^f + \Delta t_k^f}$. Each event (blue dot) $\mathbf{e}_j$ is reprojected to camera frame $C_{t_k^f}$ using the linearly interpolated pose T_{t_j} and the linearly interpolated depth $Z(\mathbf{x}_j)$.

BMVC17_Rebecq.pdf (uzh.ch)

Paper: Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization

- Feature Tracking and Landmark Triangulation
  - FAST corner detector
  - Bucketing
  - Dispose failed landmarks
  - RANSAC filtering



BMVC17_Rebecq.pdf (uzh.ch)

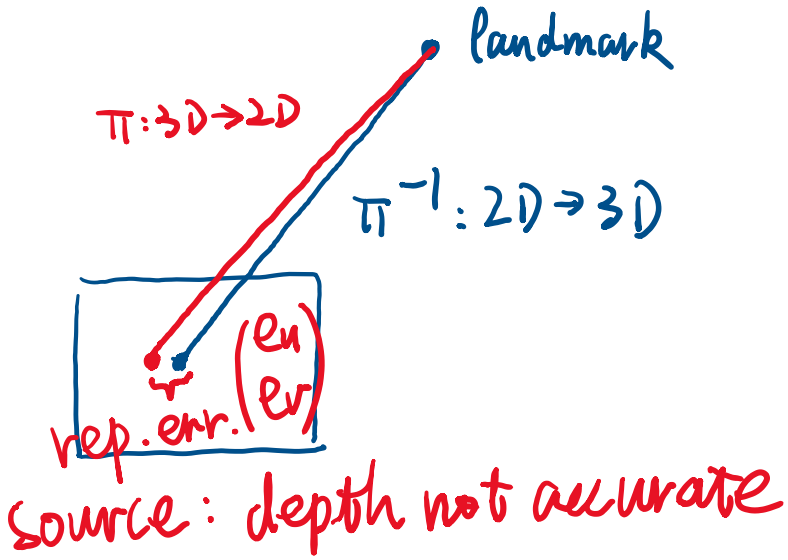
Paper: Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization

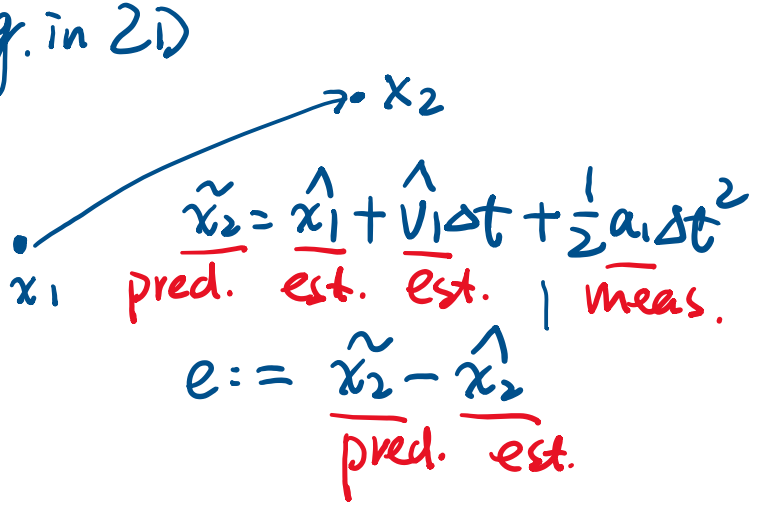
- Optimization on Reprojection Error and Inertial Error:

$$J := \sum_{k=1}^{K} \sum_{j \in \mathcal{J}(k)} \mathbf{e}^{j,k^T} \mathbf{W}_r^{j,k} \mathbf{e}^{j,k} + \sum_{k=1}^{K-1} \mathbf{e}_s^{k^T} \mathbf{W}_s^k \mathbf{e}_s^k$$

j: landmark index

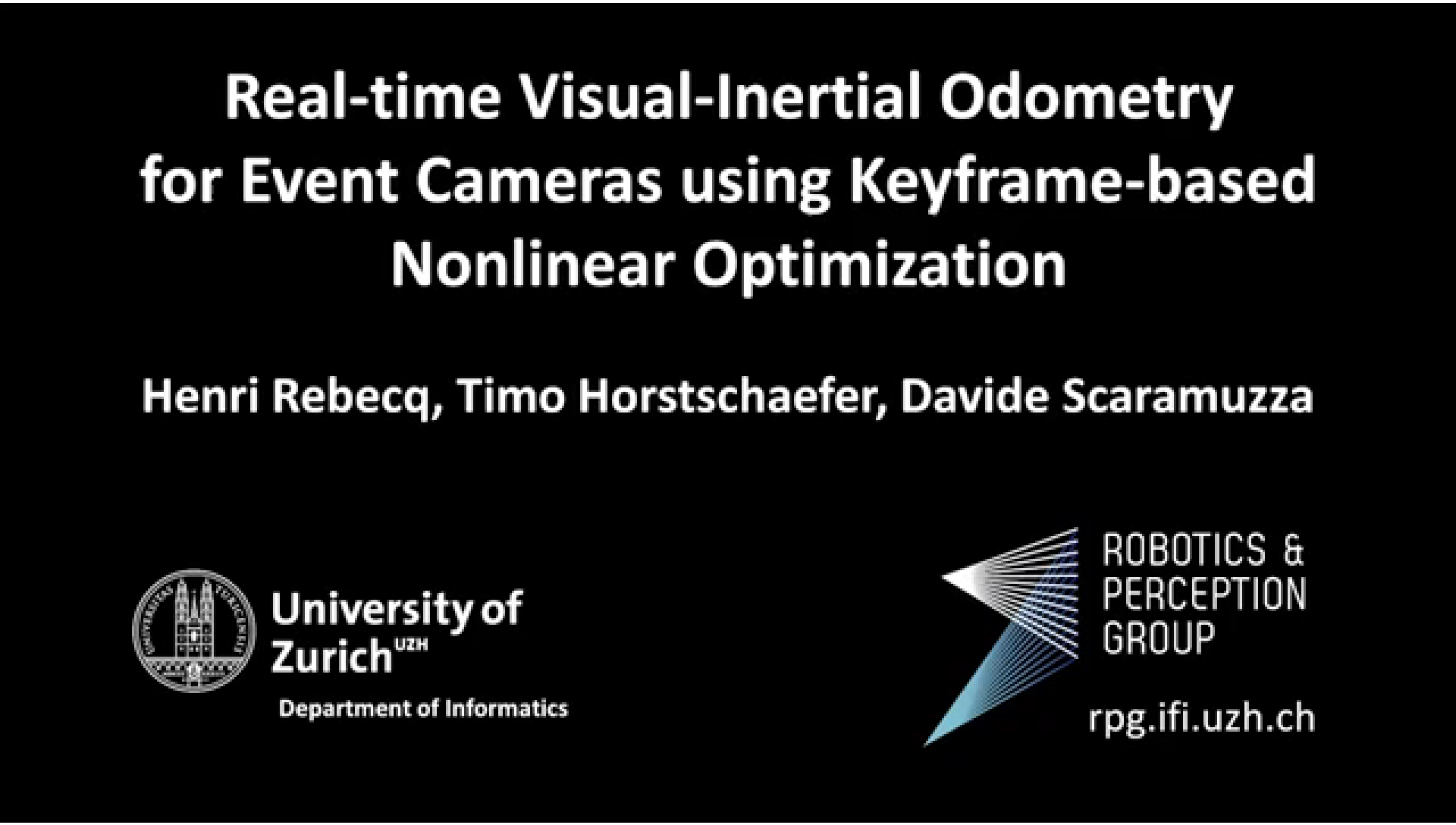
k: frame index





BMVC17_Rebecq.pdf (uzh.ch)

Paper: Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization

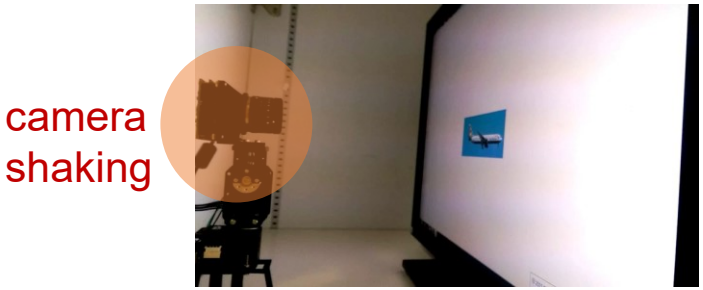


(3) Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization - YouTube

Recognition

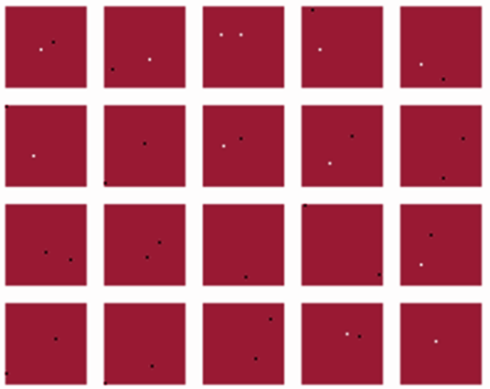
- Tasks (from easy to hard)
  - Detecting the presence of a simple shape (such as a circle)
  - Classification of more complex shapes, such as card pips, block letters and faces
  - More difficult objects: similar to conventional Caltech-101 and Caltech-256 datasets
  - Gestures and Motions

Recording Neuromorphic-Caltech101



Card symbols recognition

Event-based MNIST datasets

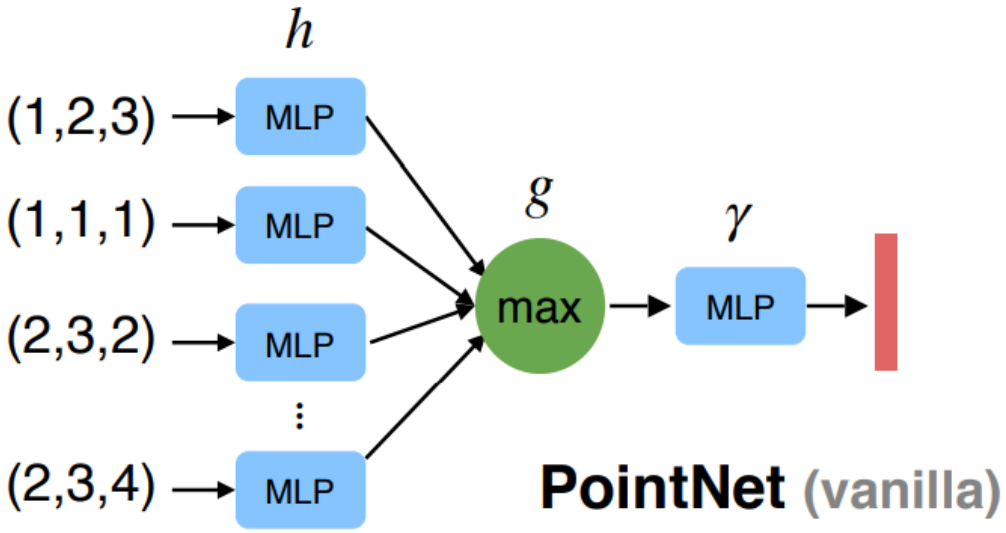


Datasets | Neuromorphic Engineering (greg-cohen.com)
www2.imse-cnm.csic.es/caviar/POKERDVS.html
Garrick Orchard - N-Caltech101

Recognition

- Ideas
  - **Event representation**: event frames, point clouds, …
  - **Recognition algorithm**: mostly using neural networks, including Convolutional Neural Network (CNN), Multi-layer Perceptron (MLP), and Spiking Neural Network (SNN)

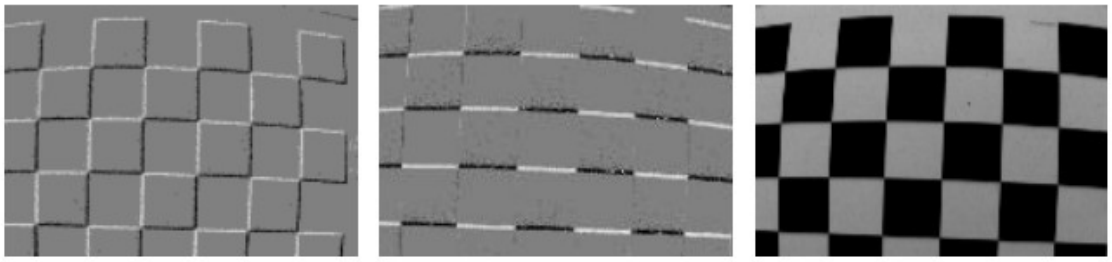
The Pointnet (which is based on MLP), can be used to process event data.



cvpr17_oral_pointnet (stanford.edu)

Recognition

- Advantages
  - Recognizing fast-moving objects / in HDR scene
  - Sparse data: may lead to less computational complexity / power consumption
  - Good at capturing motion within a time interval, instead of a timestamp
- Challenges
  - Still yet to find the most appropriate application scenarios
  - Inadequate dataset, compared to frame-base recognition
  - Requires relative motion to generate data, and data pattern depends on the motion

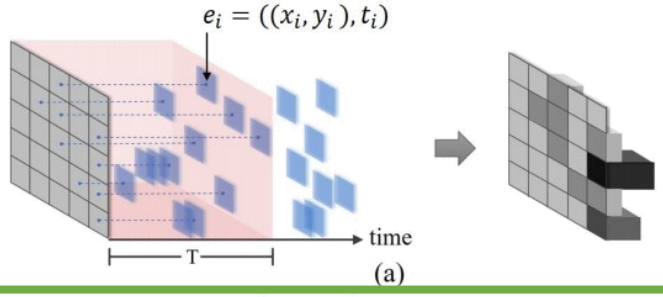


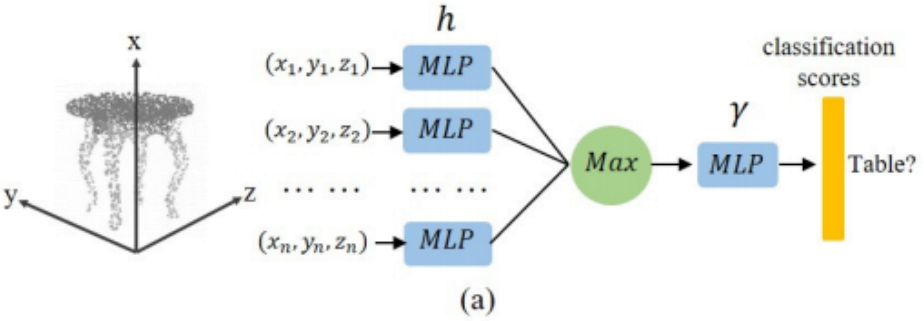
(a) (b) (c)

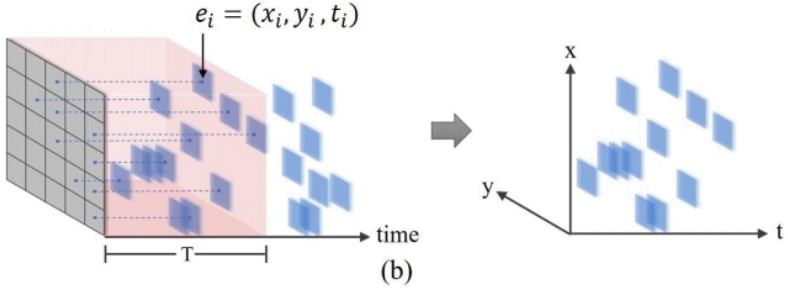
Different relative motions cause different event patterns, which brings difficulties in recognition!

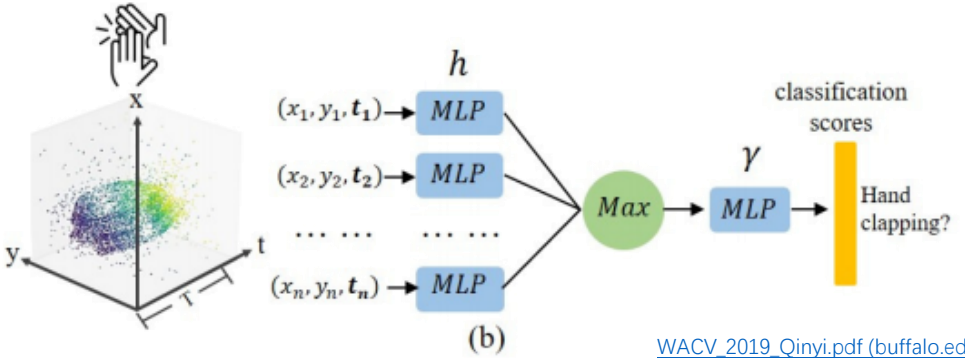
Paper: Space-time Event Clouds for Gesture Recognition

- Left: the authors treat events as space-time event clouds in this paper.
- Right: the PointNet framework can be adapted to consume space-time event cloud for gesture recognition.





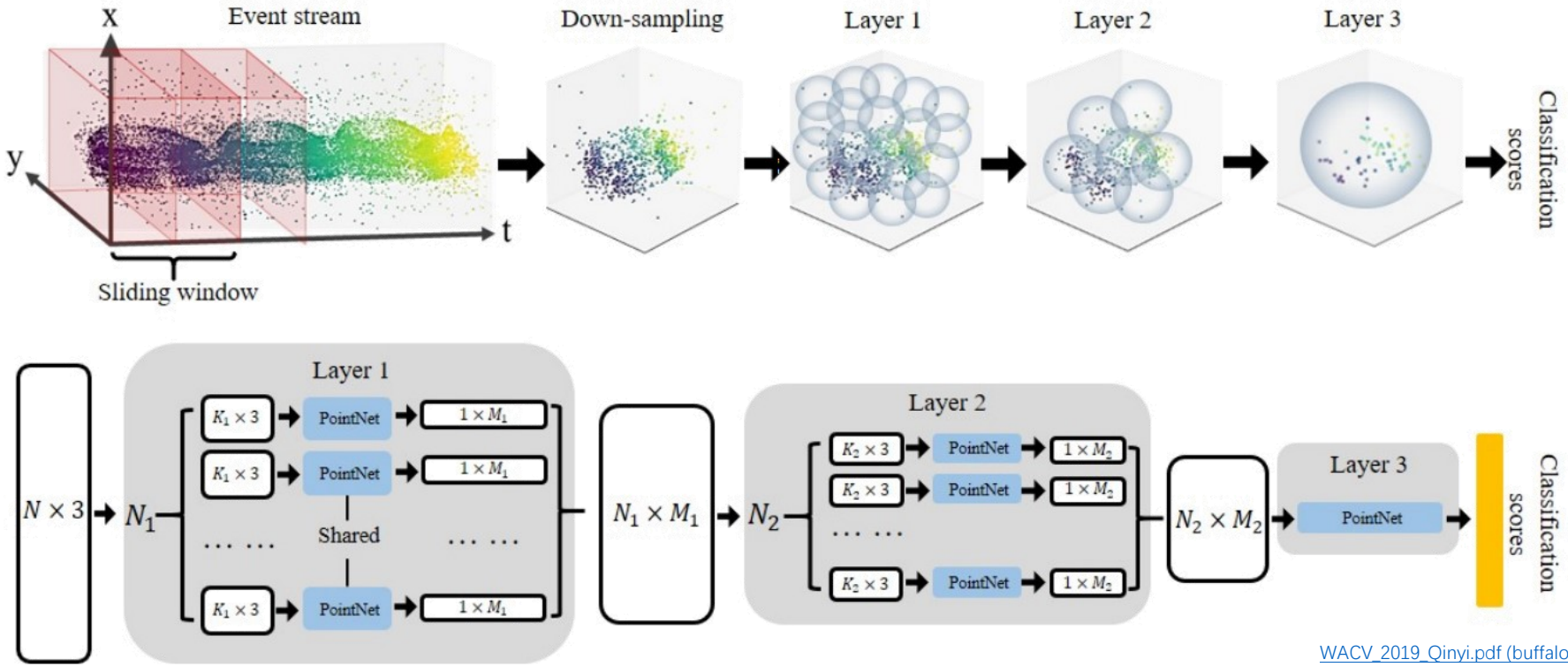




WACV_2019_Qinyi.pdf (buffalo.edu)

Paper: Space-time Event Clouds for Gesture Recognition

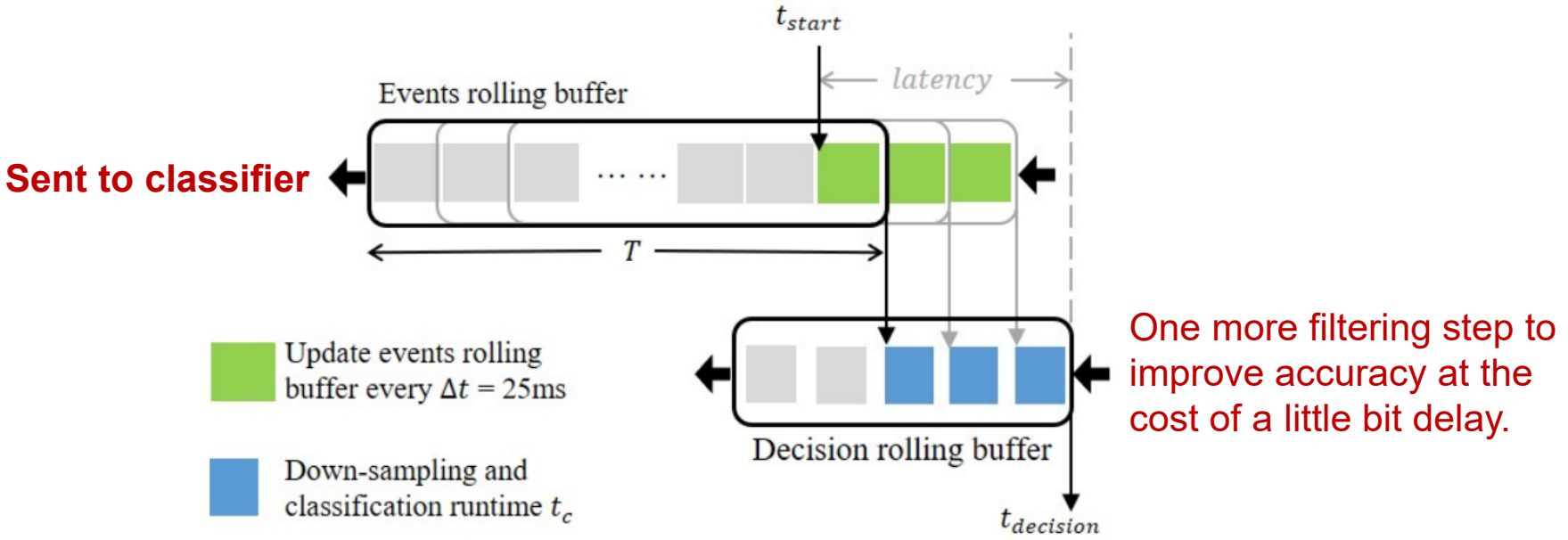
- The gesture classification framework
  - Trained based on PointNet++, which applies PointNet recursively on a nested partitioning of the input set.
  - Fine-to-coarse, to extract local and global features!



WACV_2019_Qinyi.pdf (buffalo.edu)

Paper: Space-time Event Clouds for Gesture Recognition

- The rolling buffer framework for real-time recognition
  - **Events rolling buffer** to store the event point clouds.
  - **Decision rolling buffer** to store the classification results.



WACV_2019_Qinyi.pdf (buffalo.edu)

Paper: Space-time Event Clouds for Gesture Recognition

- Results
  - Tested on IBM DVS128 Gesture dataset
  - Results for recognizing 10 classes >95%
  - Small data scale and efficient computation, compared to CNN-based approach!

Experiment (model,classes,events)	Sliding window		
	T=0.25s	T=0.50s	T=1.00s
PointNet,10,256	87.85	89.63	88.54
PointNet,10,512	88.67	**90.20**	89.61
PointNet,10,1024	88.77	89.68	89.92
PointNet++,10,256	95.28	95.59	95.54
PointNet++,10,512	95.39	**96.34**	95.61
PointNet++,10,1024	94.93	95.89	95.97
PointNet++,11,256	91.92	93.38	93.61
PointNet++,11,512	92.23	**94.10**	93.83
PointNet++,11,1024	91.87	91.91	92.63

Why T=0.50s results are the best?

Why 1024-points results worse then 512-point results?

WACV_2019_Qinyi.pdf (buffalo.edu)