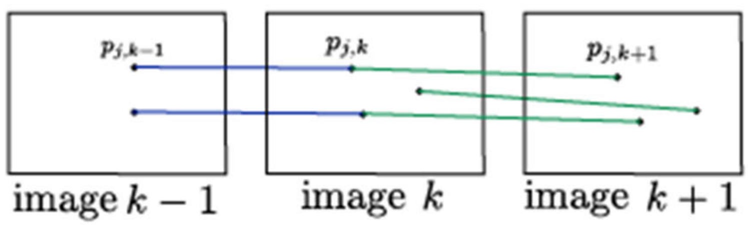
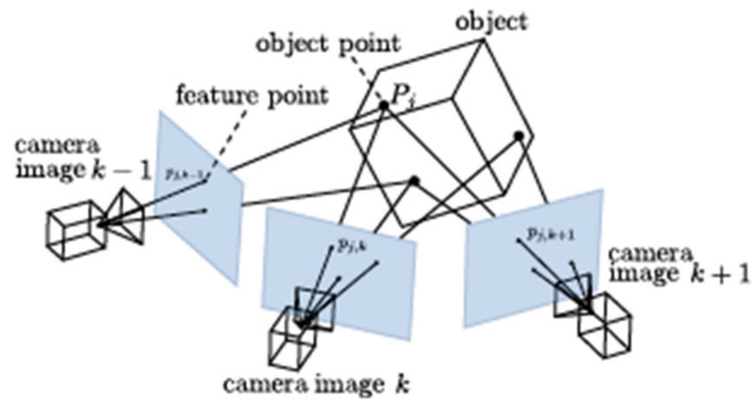
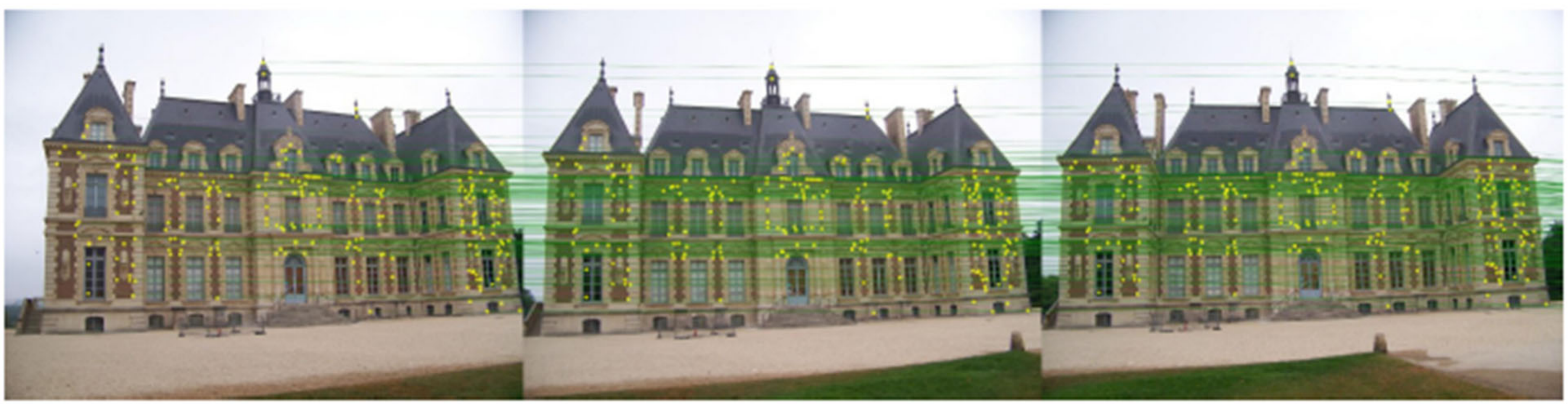
3D Motion Analysis with Motion Signals

Cornelia Fermüller
Computer Vision Laboratory, UMIACS
University of Maryland

Motion Estimation: Classical Approach

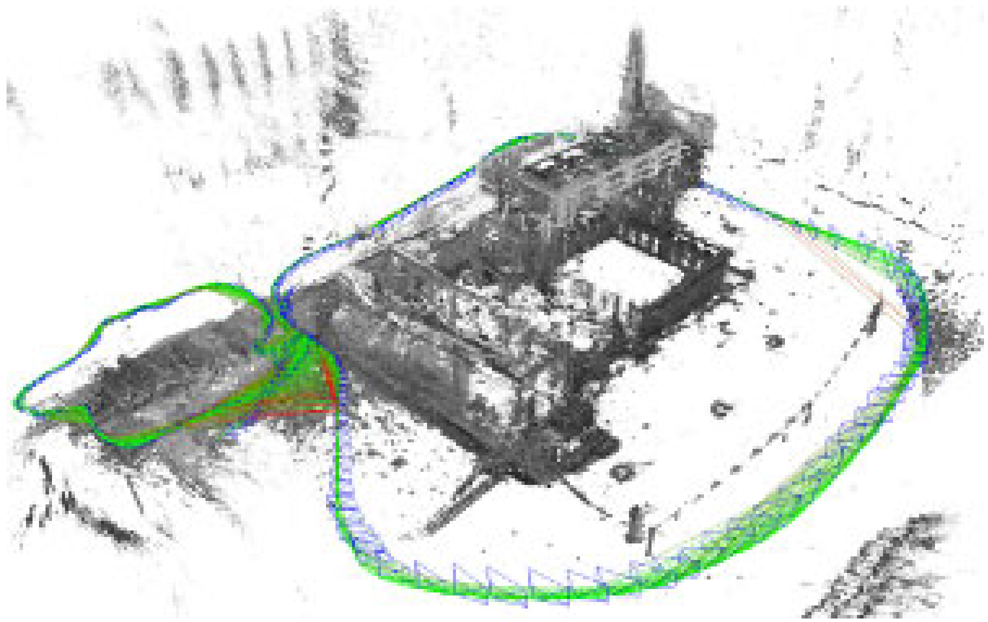






original image: [OpenMVG]

State of the Art: Visual SLAM



LSD-SLAM (Vision Group TUM)

Issues with the Reconstruction Approach

- LIDAR/Vision: dependent on sensor and its range
- Difficulties with moving objects
- Challenges with fast changes in system motion
- Computationally expensive

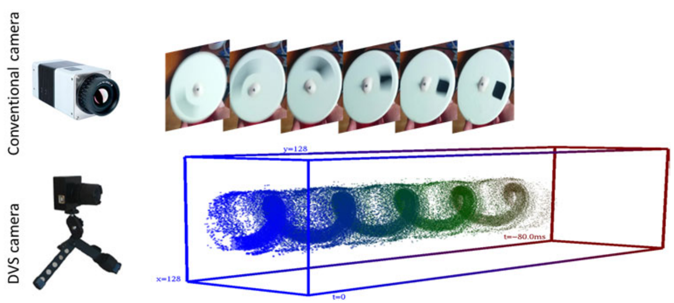
Robust Visual Navigation in Nature

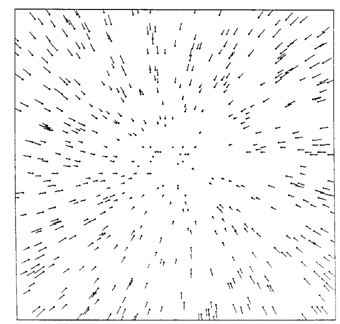
Visual Motion Capabilities

- Kinetic stabilization, Ego-motion
- Independent motion detection
- Obstacle avoidance
- Target pursuit
- Homing
- Landing

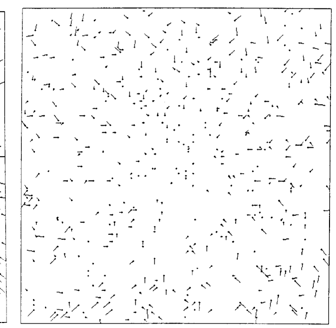
Image Motion Measurements

1. The motion signal from
   spatio-temporal filters: normal flow
2. Events with the DVS sensor

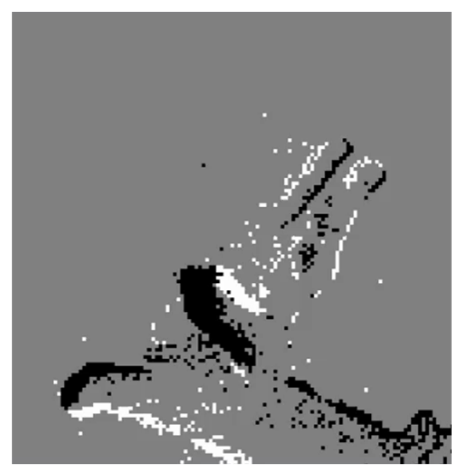




Optic flow

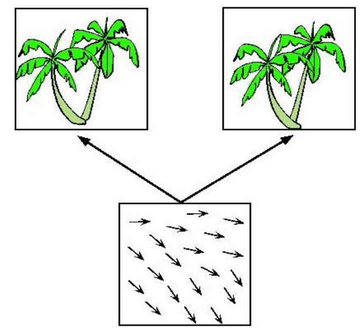


Normal flow

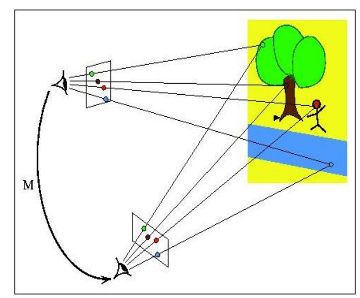


Motion Interpretation

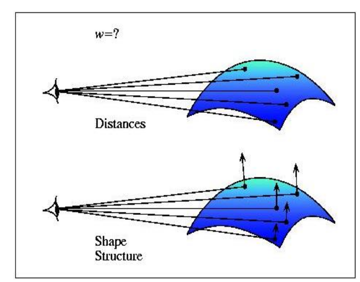
1. Image motion
   or correspondence



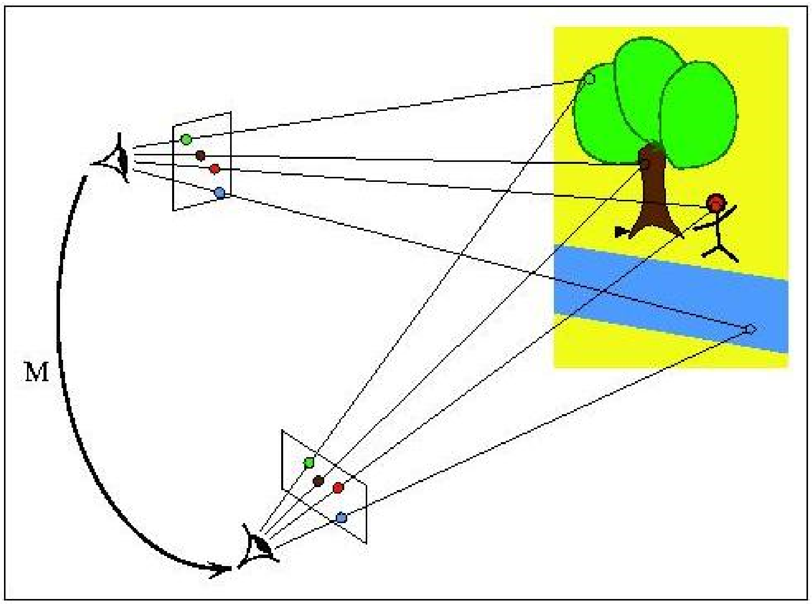
2. Transformation between
   views (3D motion)

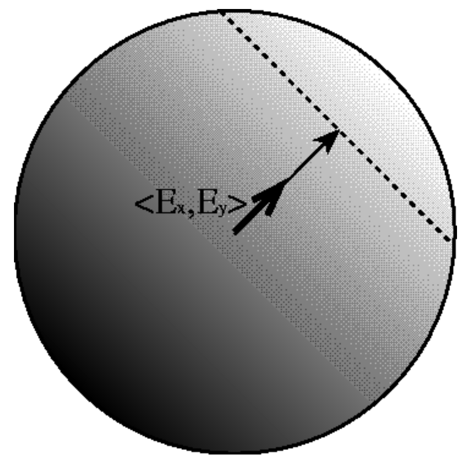


3. Scene geometry

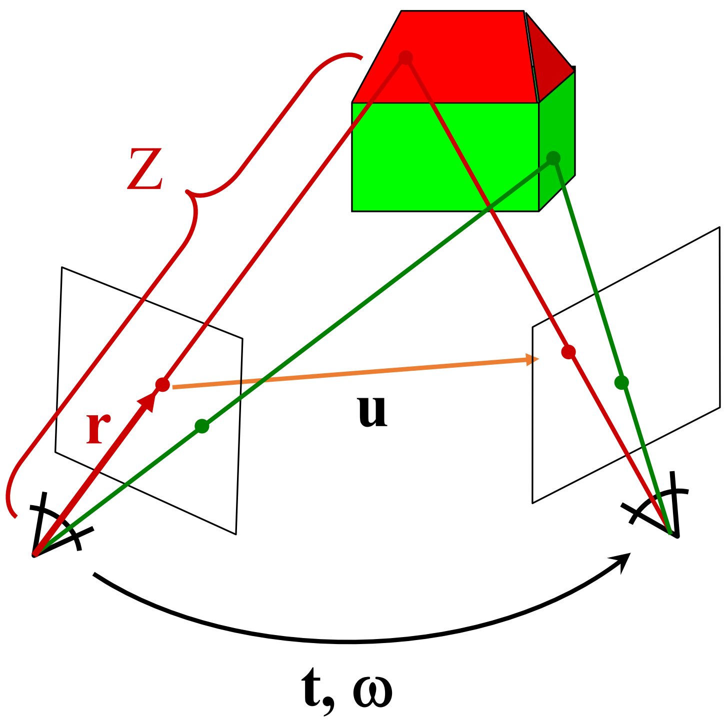


3D Motion from Normal Flow





Structure from Motion



$$\mathbf{u} = \mathbf{u}_{\mathrm{tr}} + \mathbf{u}_{\mathrm{rot}}$$

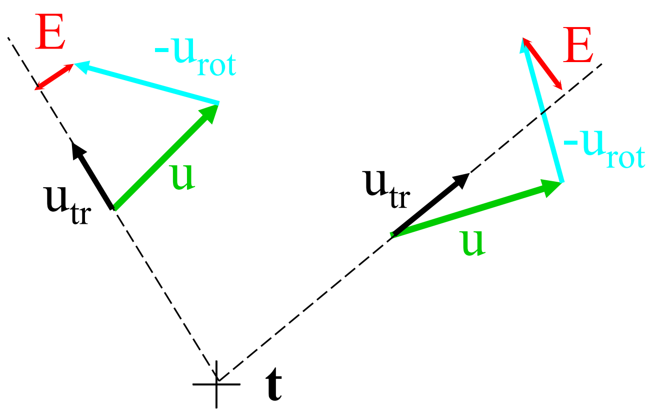
$$\mathbf{u}_{\mathrm{tr}} = \frac{1}{Z}\left(\hat{\mathbf{z}} \times (\mathbf{t} \times \mathbf{r})\right)$$

$$\mathbf{u}_{\mathrm{rot}} = \frac{1}{F}\left(\hat{\mathbf{z}} \times \left(\mathbf{r} \times ([\ \omega]_{\times}\mathbf{r})\right)\right)$$

Classical Structure from Motion

- Established approach is the epipolar minimization: The "derotated flow" should be parallel to the translational flow.

$$\left(\mathbf{u} - \mathbf{u}_{\mathrm{rot}}(\widehat{\omega})\right) \cdot (\hat{\mathbf{z}} \times \mathbf{u}_{\mathrm{tr}}(\hat{\mathbf{t}})) = 0$$



$$argmin\ \hat{\mathbf{t}}, \widehat{\omega}\ \textstyle\sum_i \left|\left|\left(\mathbf{u}_i - \mathbf{u}_{\mathrm{rot}_i}(\widehat{\omega})\right) \cdot \left(\hat{\mathbf{z}} \times \mathbf{u}_{\mathrm{tr}_i}(\hat{\mathbf{t}})\right)\right|\right|_2$$

With Normal Flow only

$$u_{tr}(\mathbf{t}) = A(x)\mathbf{t}$$
$$u_{rot}(\omega) = \mathbf{B}(\mathbf{x})\mathbf{w}$$

$$\|\mathbf{u_n}(\mathbf{x})\| = \frac{A(\mathbf{x})\mathbf{t}}{Z(\mathbf{x})} \cdot \mathbf{n}(\mathbf{x}) + B(\mathbf{x})\mathbf{w} \cdot \mathbf{n}(\mathbf{x})$$

$$(\|\mathbf{u_n}(\mathbf{x})\| - B(\mathbf{x})\mathbf{w} \cdot \mathbf{n}(\mathbf{x})) \cdot (A(\mathbf{x})\mathbf{t} \cdot \mathbf{n}(\mathbf{x})) > 0$$

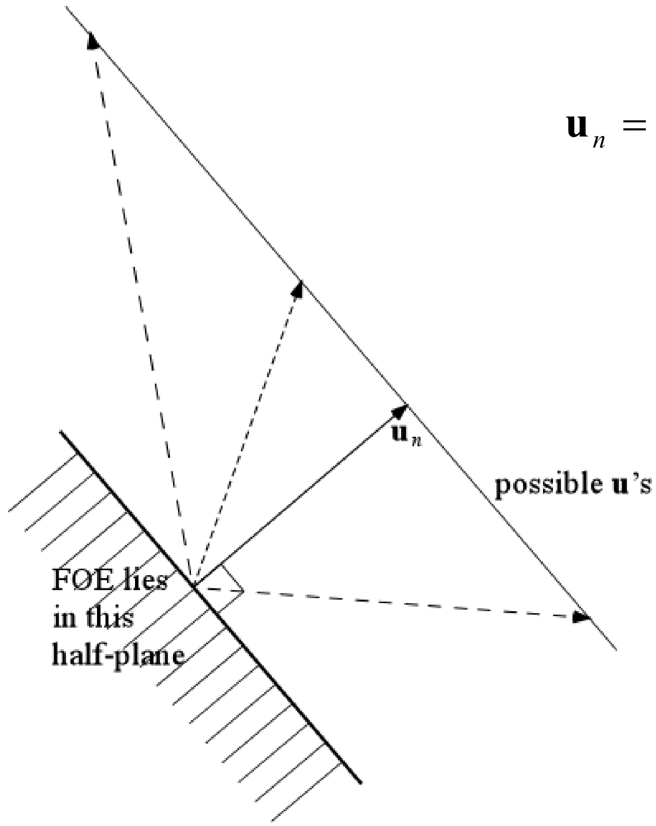
How was it implemented ?

$$arg\,min\ _{\tilde{\mathbf{t}},\mathbf{w}} \sum_{i=1}^{N} \mathcal{V}(\mathbf{x}_i, \tilde{\mathbf{t}}, \mathbf{w}) \quad \text{with} \tag{1}$$

$$\mathcal{V}(\mathbf{x}, \mathbf{t}, \mathbf{w}) =$$

$$\begin{cases} 0 & \text{if} \, (\|\mathbf{u_n}(\mathbf{x})\| - \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x})\mathbf{w}) \cdot (\mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x})\mathbf{t}) > 0 \\ 1 & \text{if} \, (\|\mathbf{u_n}(\mathbf{x})\| - \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x})\mathbf{w}) \cdot (\mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x})\mathbf{t}) < 0 \end{cases} \tag{2}$$

Translational Normal Flow



$$\mathbf{u}_n = \frac{\mathbf{u}_{\mathrm{tr}}}{Z} \cdot \mathbf{n}$$

- In the case of translation each normal flow vector constrains the location of the FOE to a half-plane.
- Intersection of half-planes provides FOE.

Pattern Constraints

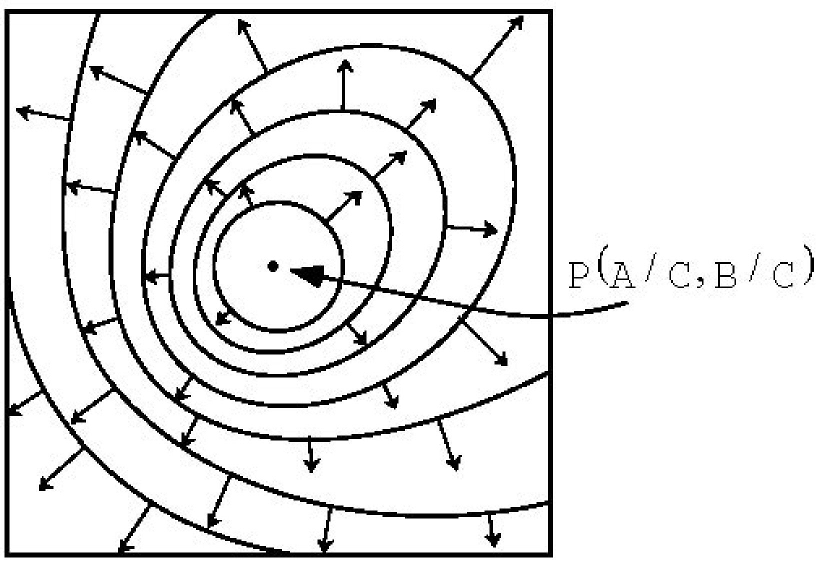
With only sign of normal flow

$$\mathcal{V}_r(\mathbf{x}, \tilde{\mathbf{t}}, \mathbf{w}) =$$

$$\begin{cases} 1 & \text{if } \|\mathbf{u_n}(\mathbf{x})\| > 0, \ \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x})\mathbf{w} < 0, \ \mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x})\tilde{\mathbf{t}} < 0 \\ 1 & \text{if } \|\mathbf{u_n}(\mathbf{x})\| < 0, \ \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x})\mathbf{w} > 0, \ \mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x})\tilde{\mathbf{t}} > 0 \\ 0 & \text{otherwise} \end{cases}$$

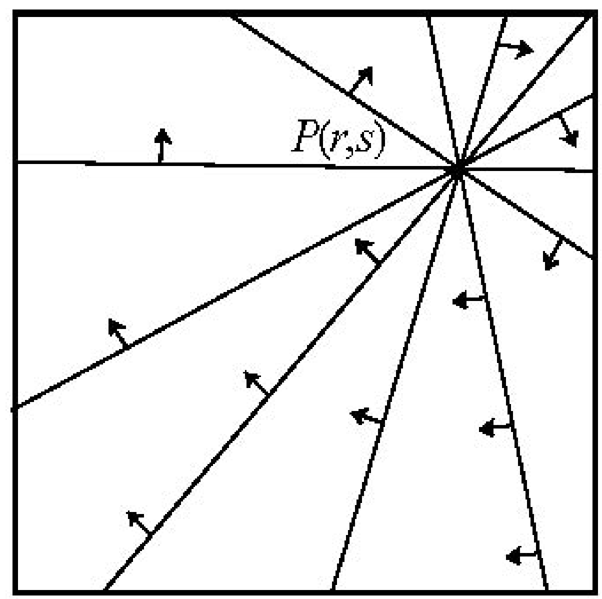
Coaxis vectors

with respect to axis ω = (A,B,C)



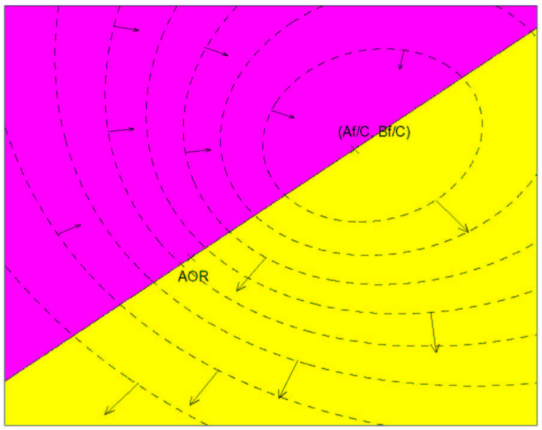
Copoint vectors

with respect to point **t**/||**t**|| = (r,s)

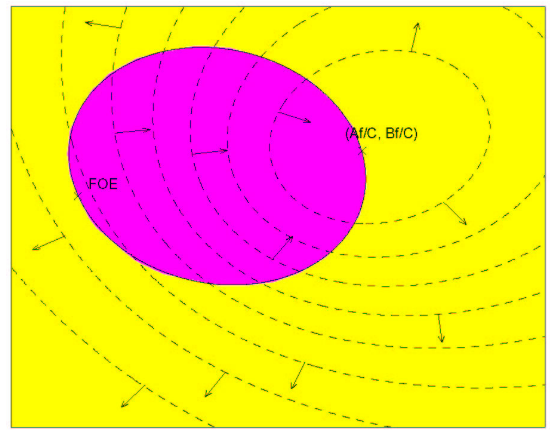


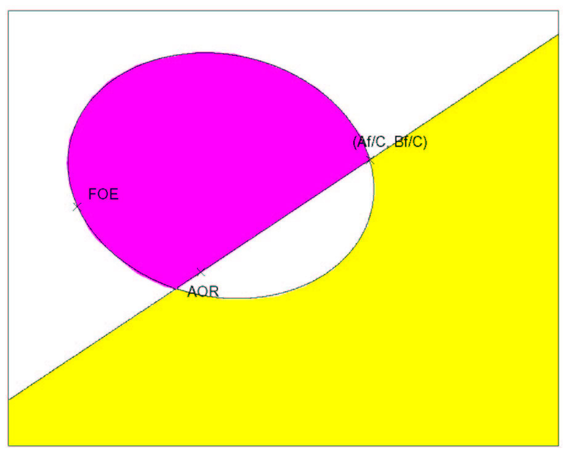
Coaxis pattern

translational

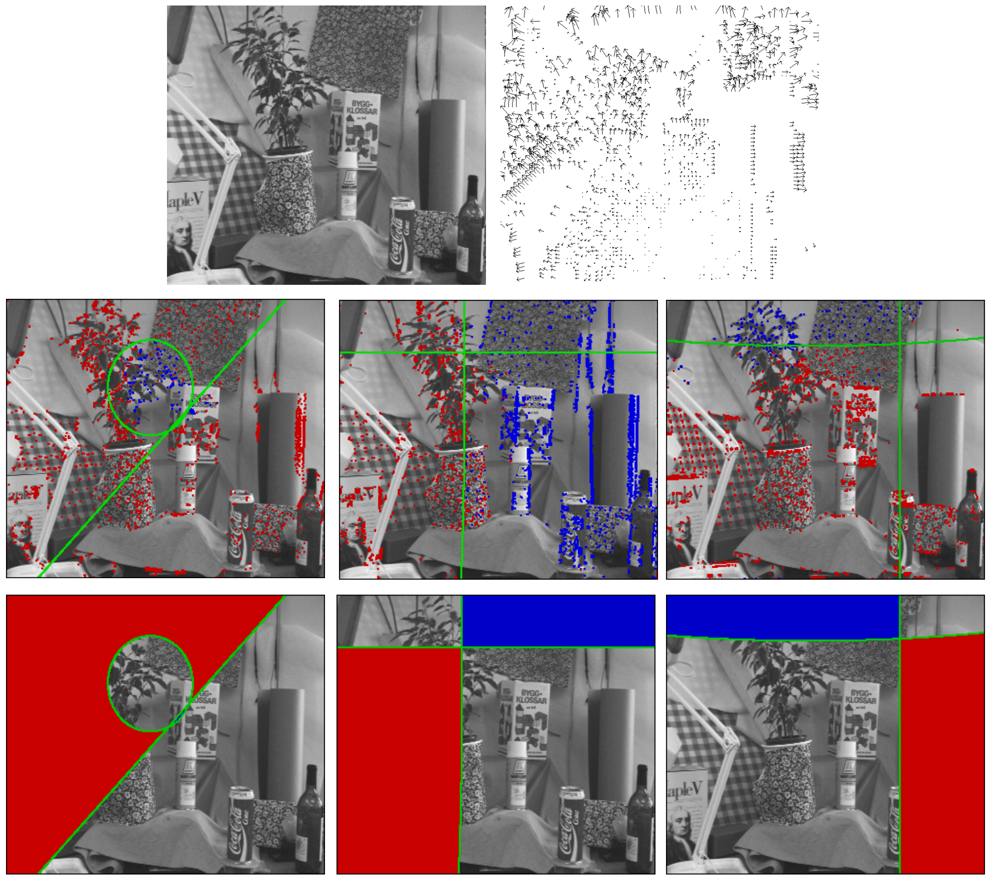


rotational





combined



C. Fermüller, Y. Aloimonos. Direct Perception of Three-Dimensional Motion from Patterns of Visual Motion. Science 22, 27 1995

A new implementation of the positivity constraint

$$f(\mathbf{t}, \mathbf{w}, \mathbf{x}) = (u_{\mathbf{n}}(\mathbf{x}) - \mathbf{n}(\mathbf{x}) \cdot B(\mathbf{x})\mathbf{w}) \cdot (\mathbf{n}(\mathbf{x}) \cdot A(\mathbf{x})\mathbf{t})$$

$$arg\,min_{\tilde{\mathbf{t}},\mathbf{w}} \sum_{i=1}^{N} \mathcal{H}(f(\tilde{\mathbf{t}}, \mathbf{w}, \mathbf{x}_i) \quad (1)$$

where

$$\mathcal{H}(x) = \begin{cases} -x & \text{if } x \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

F Barranco, C Fermüller, Y Aloimonos, E Ros. Joint direct estimation of 3D geometry and 3D motion using spatio temporal gradients. Pattern Recognition, 2020

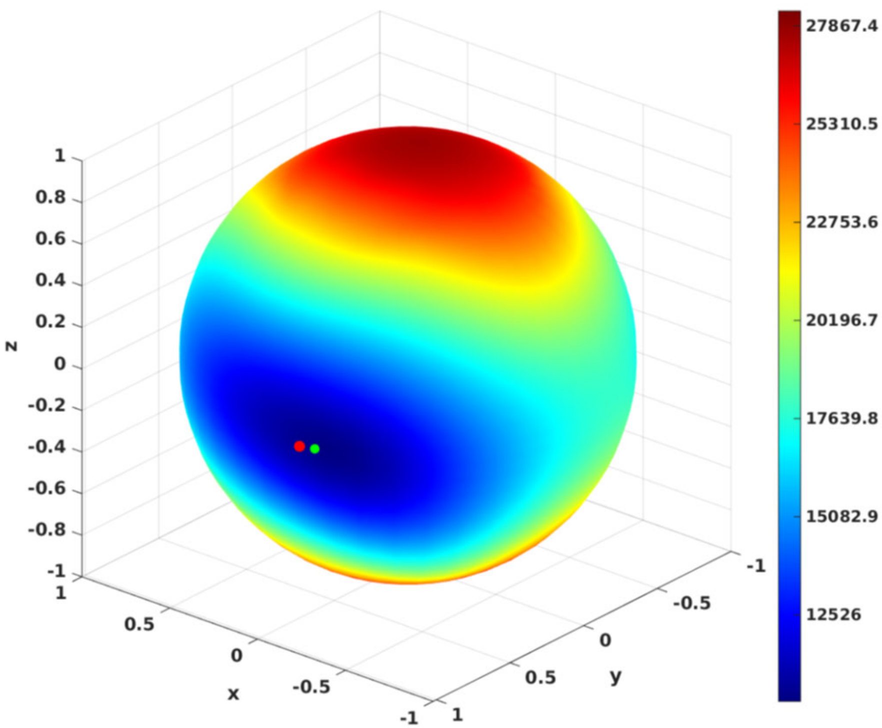
The complete method

- Step 1: Solve iteratively in **t** and **ω** the inequality using an interior method

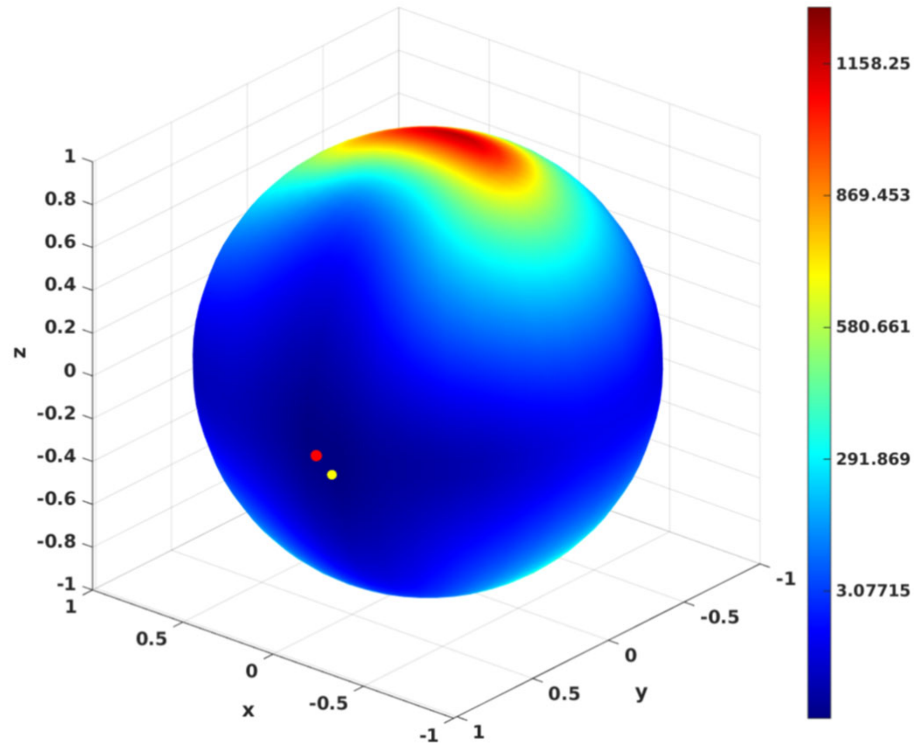
Iterate:

  - Step 2: Solve for depth, run regularization on depth via an inpainting method
  - Step 3: Solve Least squares for **t** and **ω** (given the depth)

Behavior of Error function



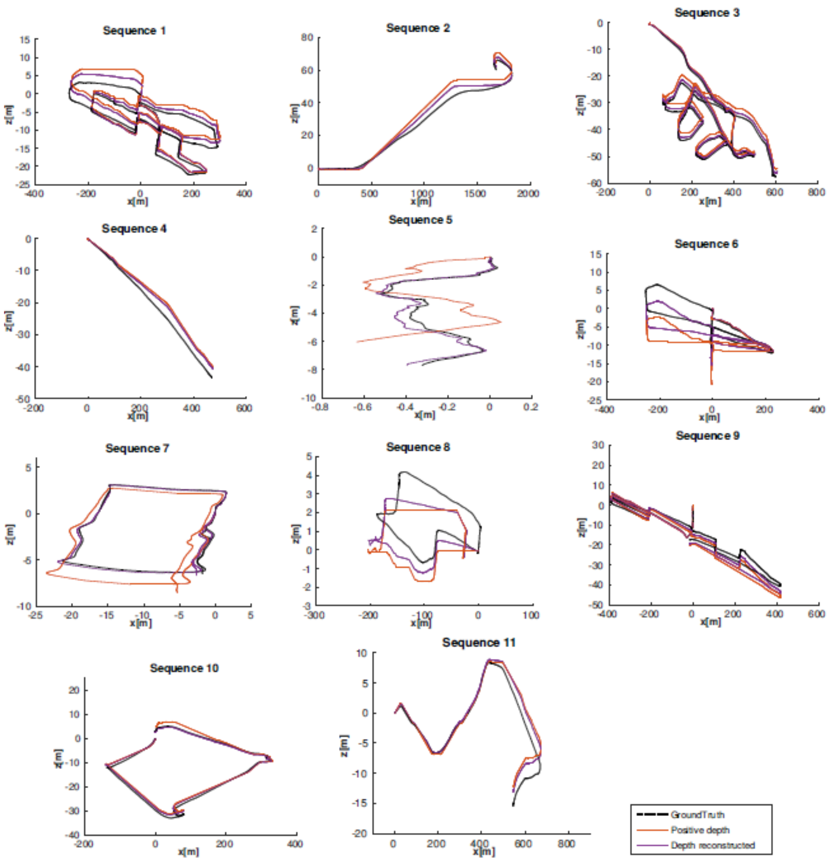
epipolar constraint

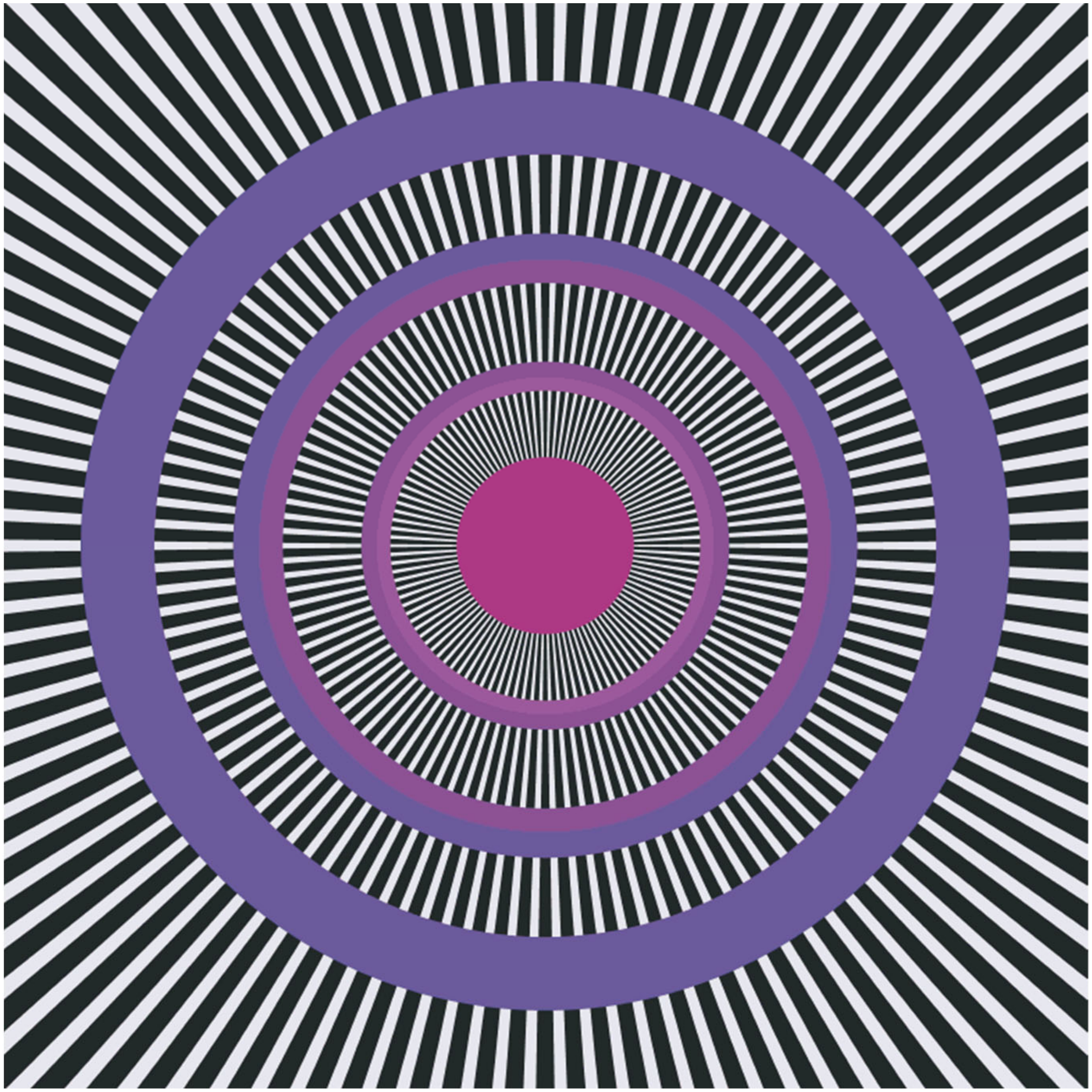


positive depth constraint with normal flow vectors

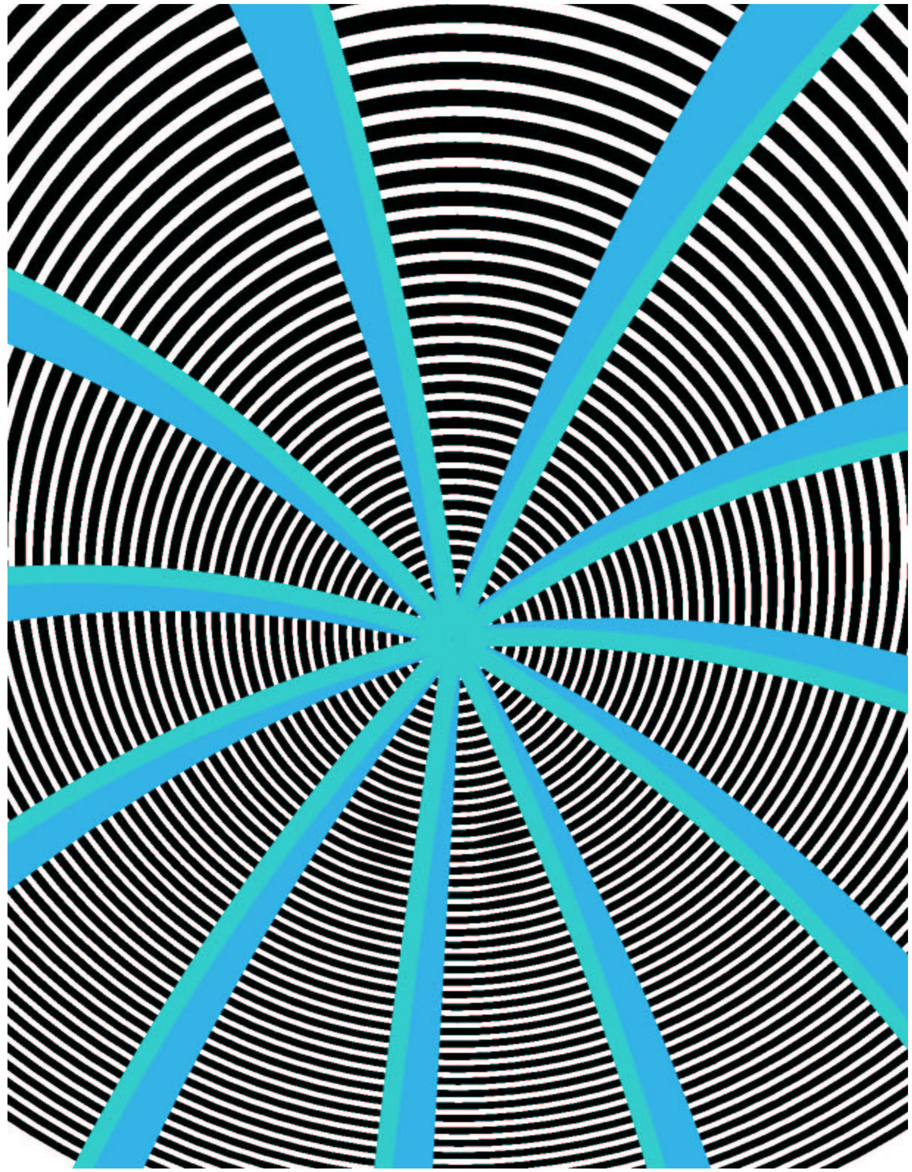
ground-truth : red dot , estimated solution: yellow/green dot.

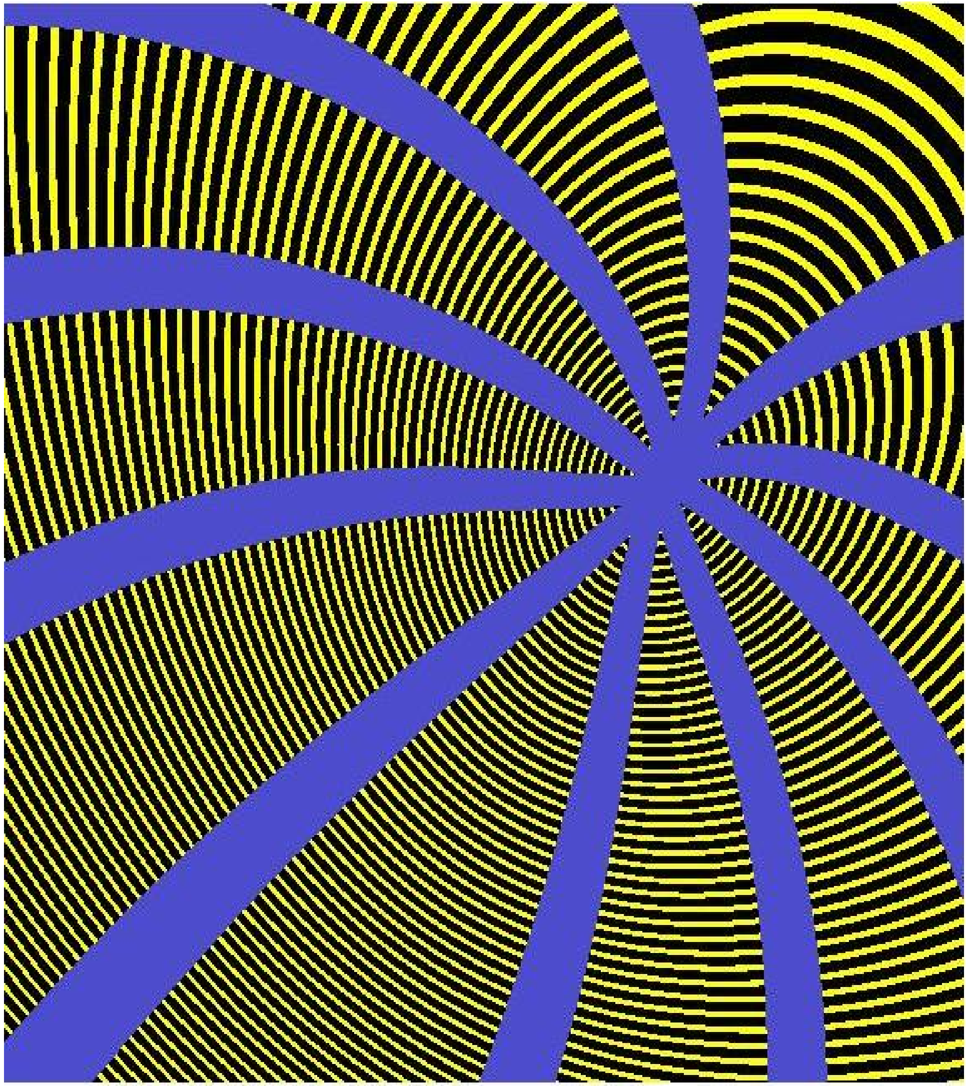
Estimated path for Kitti dataset





Optical illusion



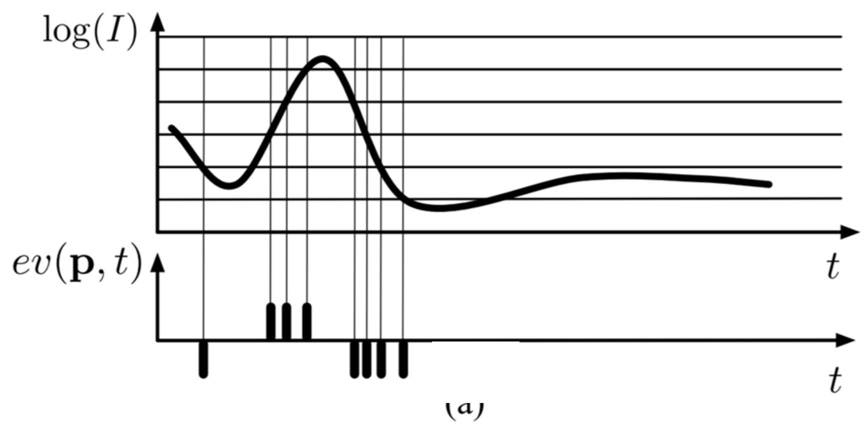


C. Fermüller, R. Pless, Y. Aloimonos. Families of stationary patterns producing illusory movement: insights into the visual system, Proc. Roy. Soc.. B., 1997.

The Dynamic Vision Sensor

DVS: An asynchronous differential camera

Events with +1 or -1 polarity are emitted when the change in log intensity exceeds a predefined threshold:

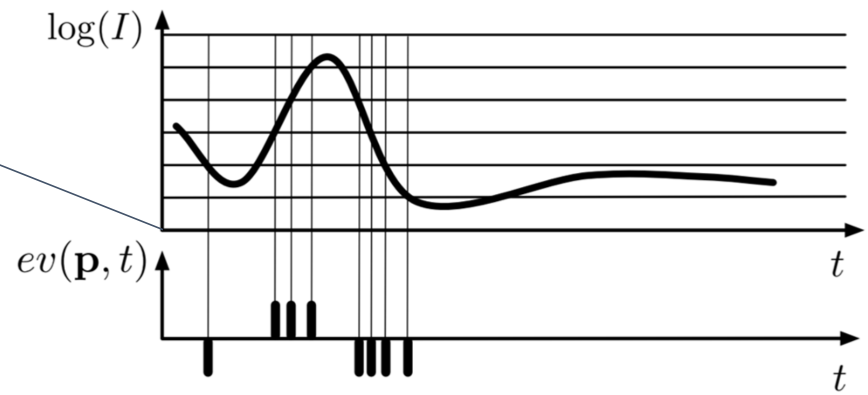


The Dynamic Vision Sensor

DVS: An asynchronous differential camera

No motion blur

Events with +1 or -1 polarity are emitted when the change in log intensity exceeds a predefined threshold:



Event = {x,y,timestamp,polarity}

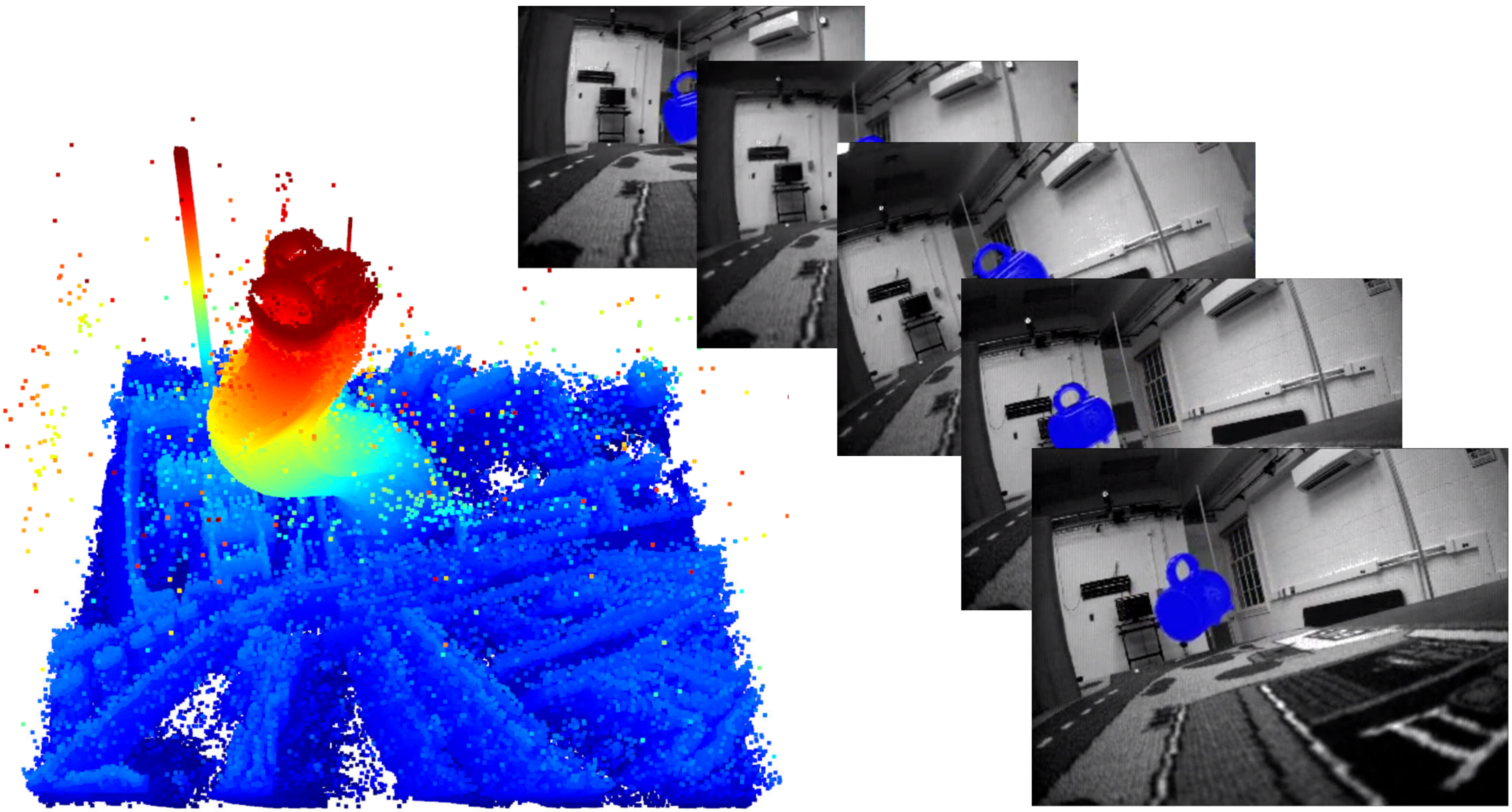
The Dynamic Vision Sensor

DVS: An asynchronous differential camera

High dynamic range

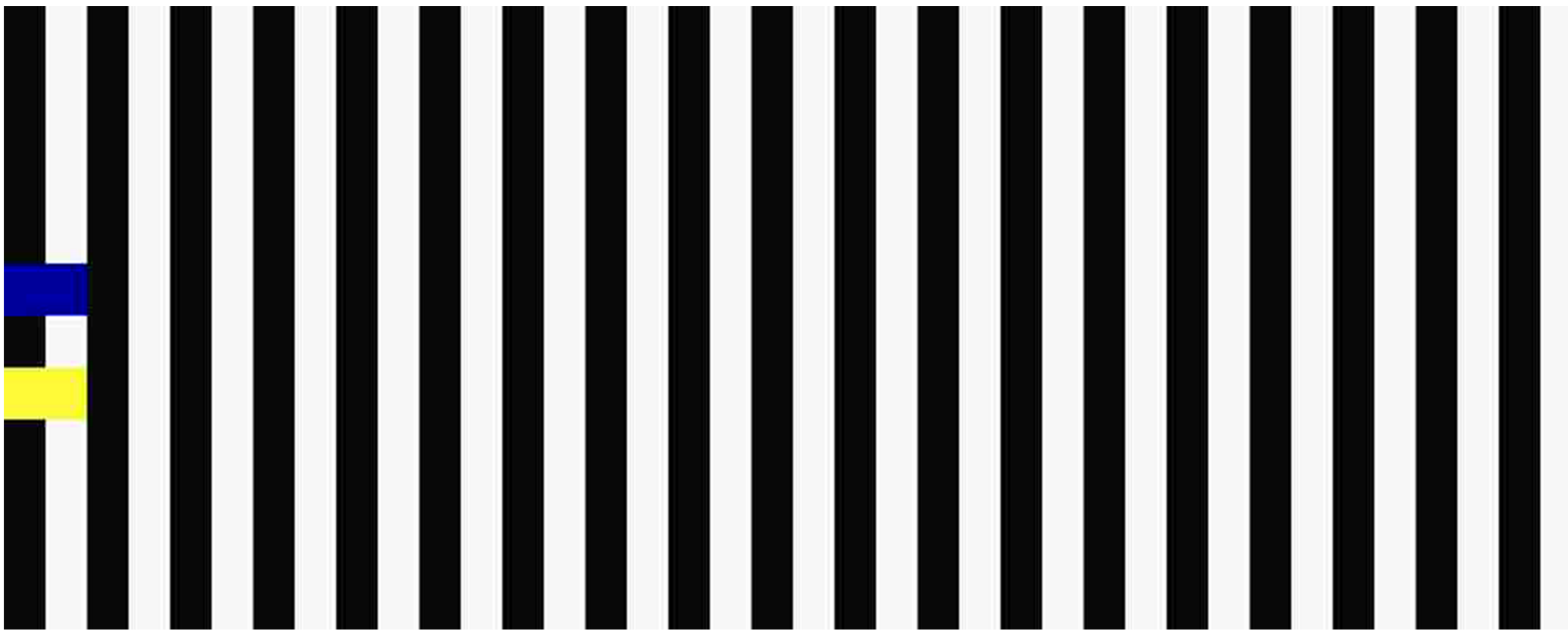
List of resources on Event-Based vision:

https://github.com/uzh-rpg/event-based_vision_resources

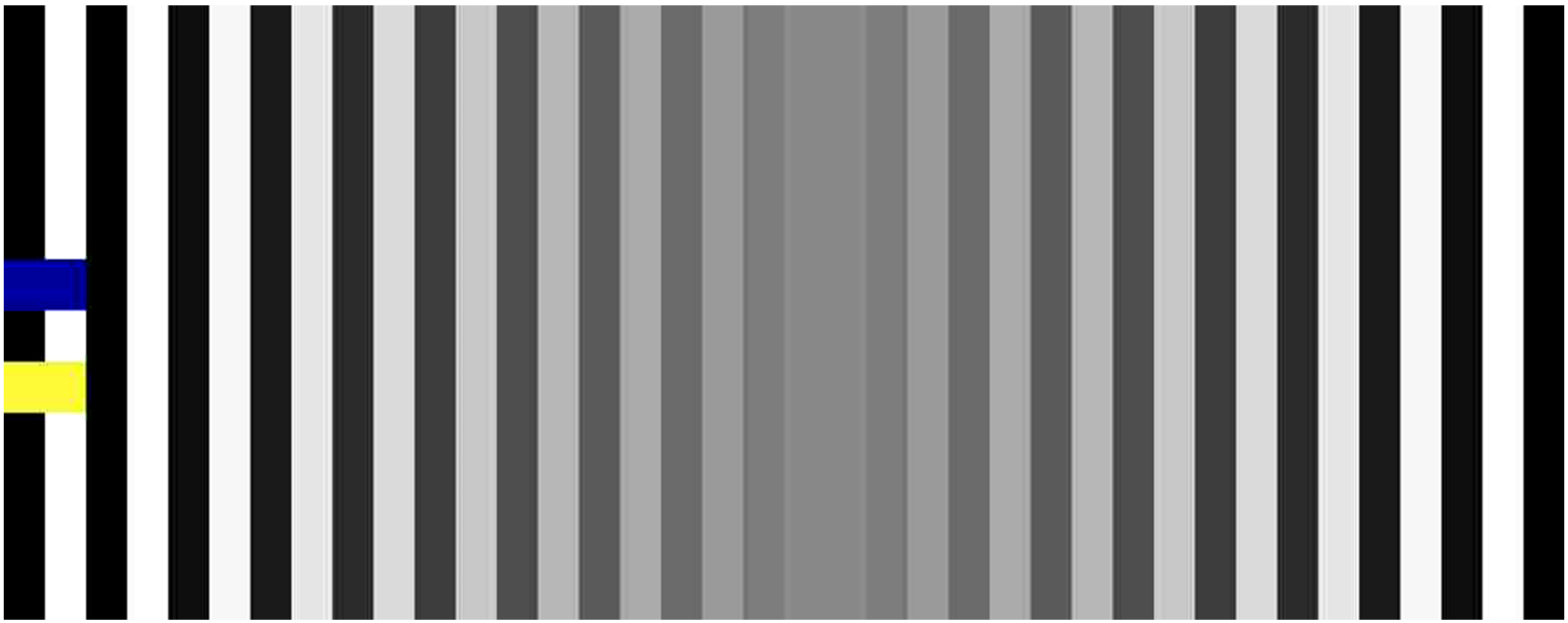


Fast events aid in segmentation

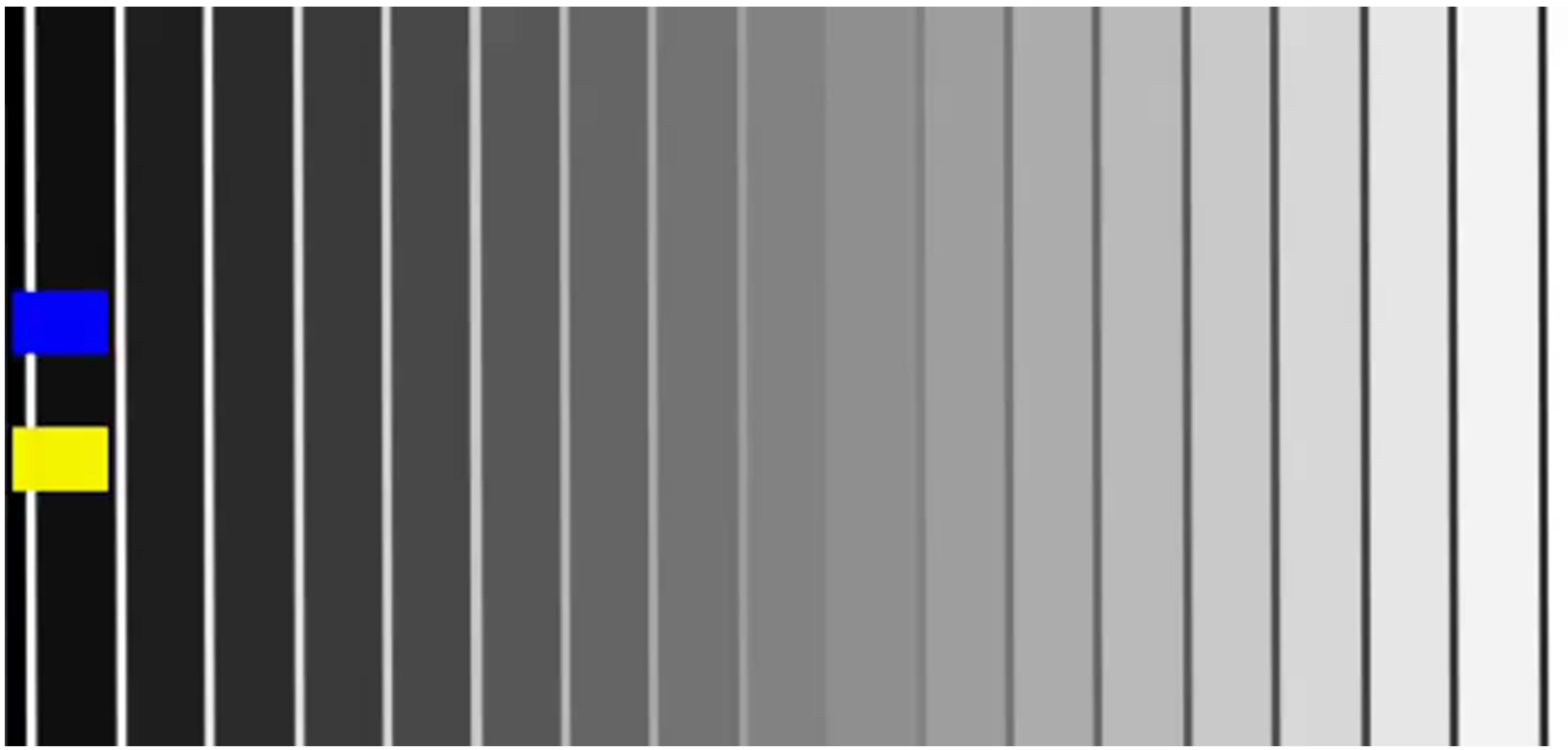
Stepping Feet Illusion



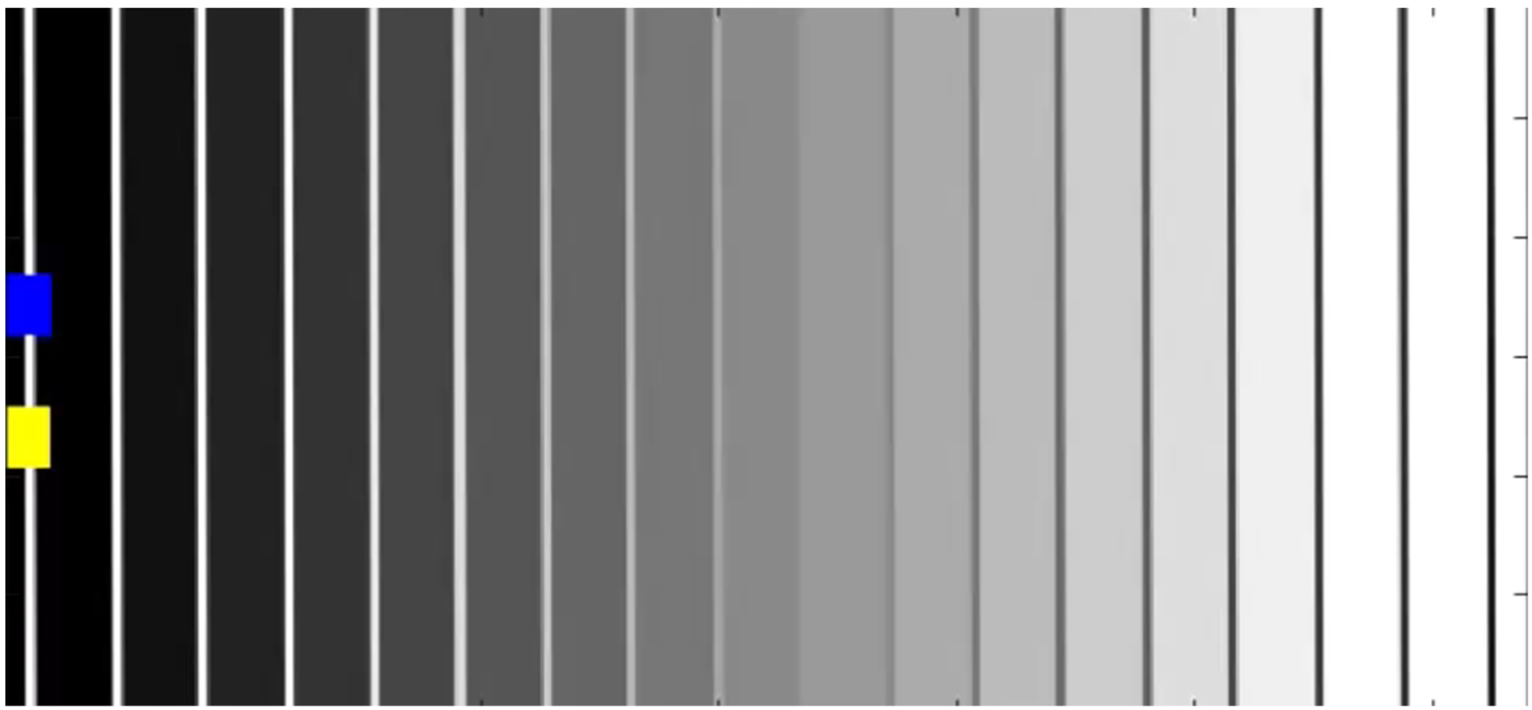
Variation of Stepping Feet Illusion



Variation of Stepping Feet Illusion



Simulation 2

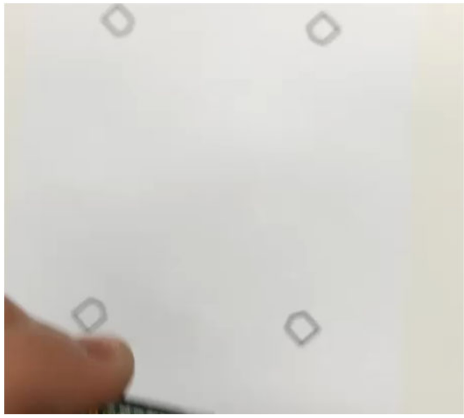


Overview

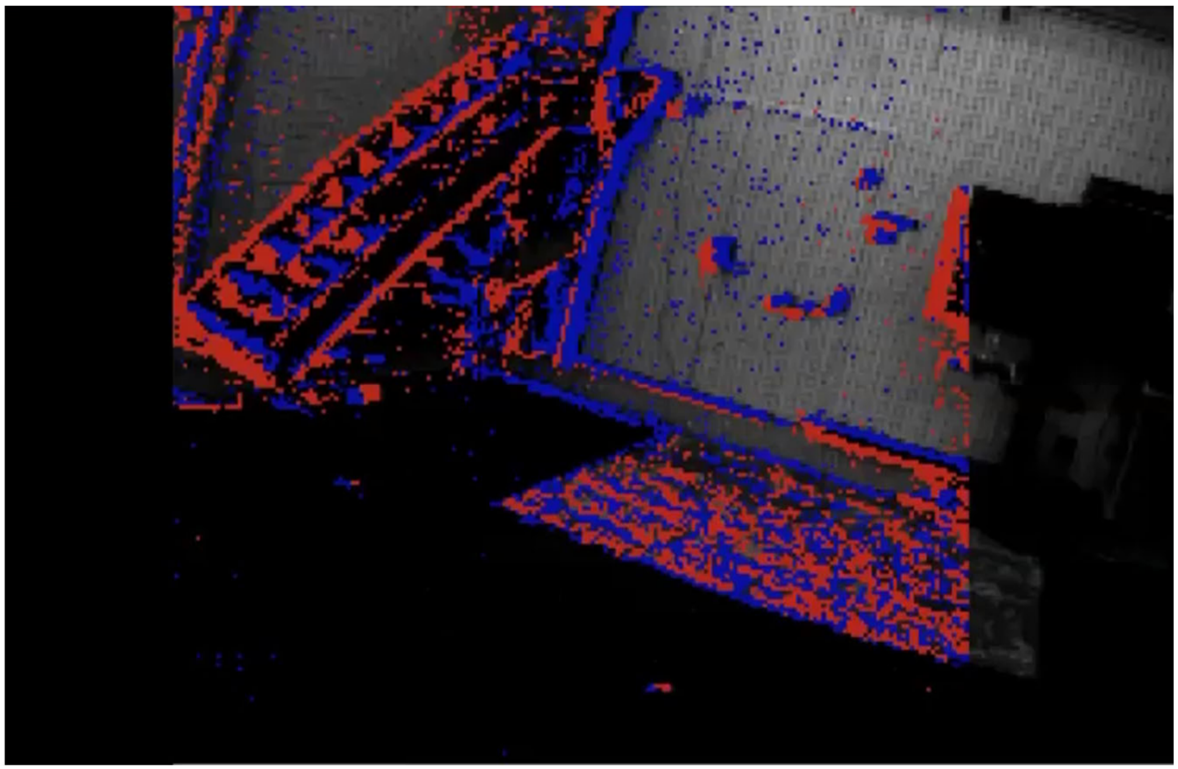
I. Optimization approach for event alignment
II. Self-supervised deep learning for motion estimation and segmentation
III. EV-IMO Dataset
IV. EVDodge: Motion detection as input to control dodging
VI. Motion segmentation in full 3D

Properties of this sensor

+ High temporal resolution
+ High dynamic range
+ Low Bandwidth signal
+ Low latency
- High noise

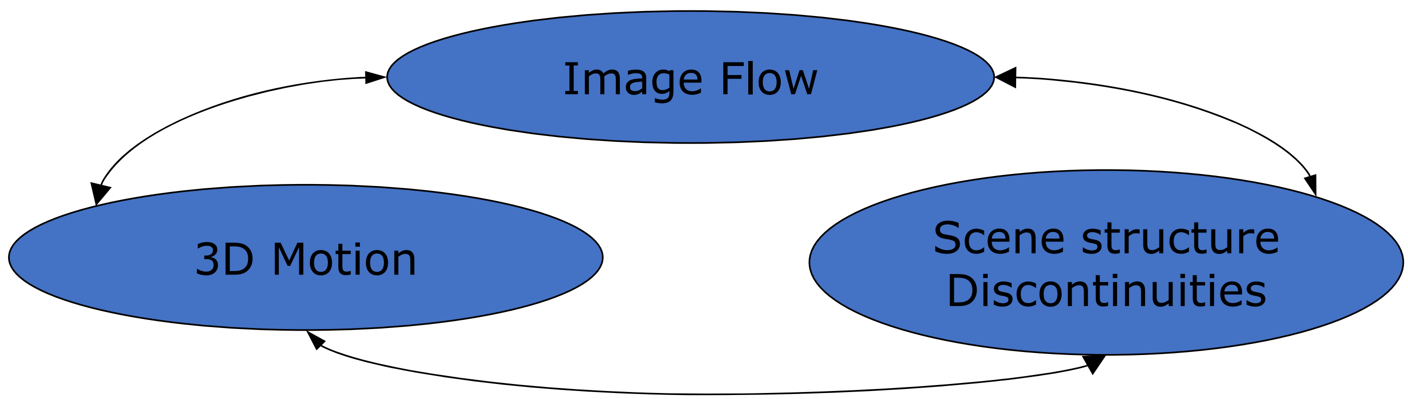


I. Egomotion+ Independent Motion

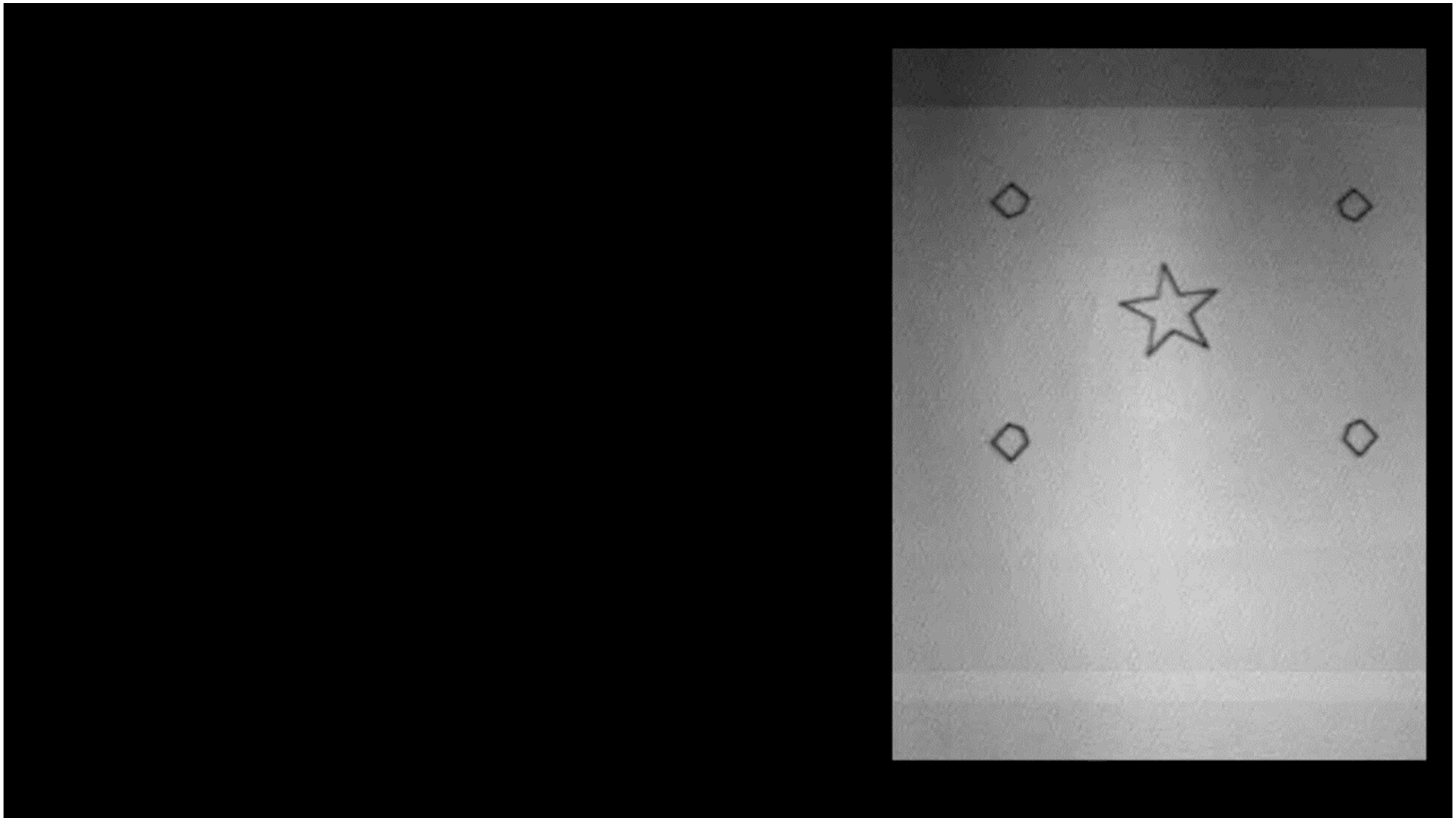


What is the problem?

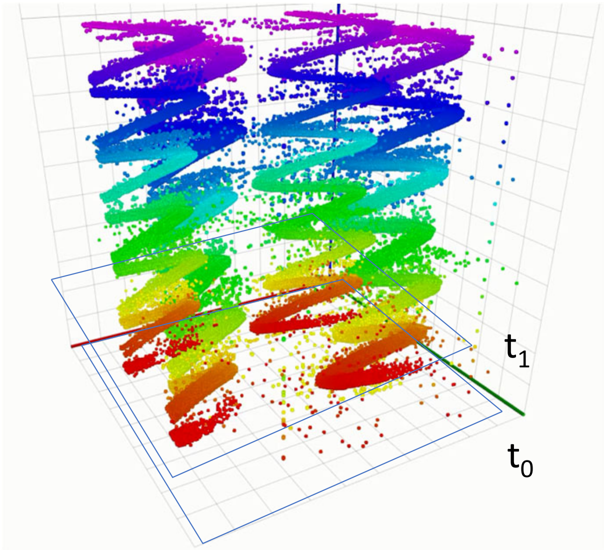
- All the components are related.



*



Treat events as point clouds



Warp field $\Phi(d_x, d_y, d_r, d_\phi) : (x, y, t) \rightarrow (x + u\Delta t, y + v\Delta t, t)$

d_x, d_y Shift in x and y

d_r, d_θ Radial expansion, and rotation around z-axis
Derived from divergence and curl

Approximation of 3D Motion Estimation

$$u(x,y)\Delta t = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} + \left\{ \frac{1}{2} curl_g \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} + \frac{1}{2} div_g \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right\} \begin{pmatrix} x \\ y \end{pmatrix} \Delta t$$

d_x, d_y $\quad d_\theta$ $\quad d_z$

Approximates rigid movement of fronto-parallel plane

How to compute it?

- **Density** (from Event Count image)

$$\xi_{ij} = \{\{x', y', t\} : \{x', y', 0\} \in C', \; i = x', \; j = y'\}$$

$$D = \frac{\sum i,j|\xi_{i,j}|}{\#I}$$

Sum of events over all pixels / Number of occupied pixels

- **Average time (**from Time Stamp image)

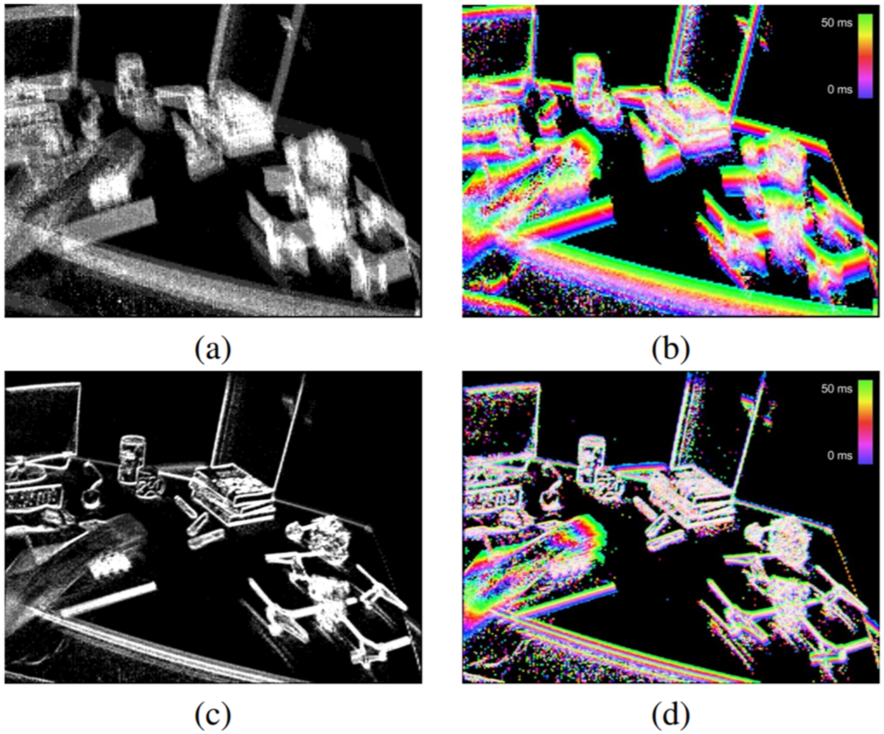
$$\mathcal{T}_{ij} = \frac{1}{\mathcal{I}_{ij}} \sum t : t \in \xi_{ij}$$

Event-based Moving Object Detection and Tracking

Idea:

1) Warp the 3D events according to a motion model: 4-parameter {x-y-z-roll}
2) Downproject all 3D events on a camera plane
3) Each pixel is the average of the event timestamps
4) Compute error gradients on the image
5) Go to (1)

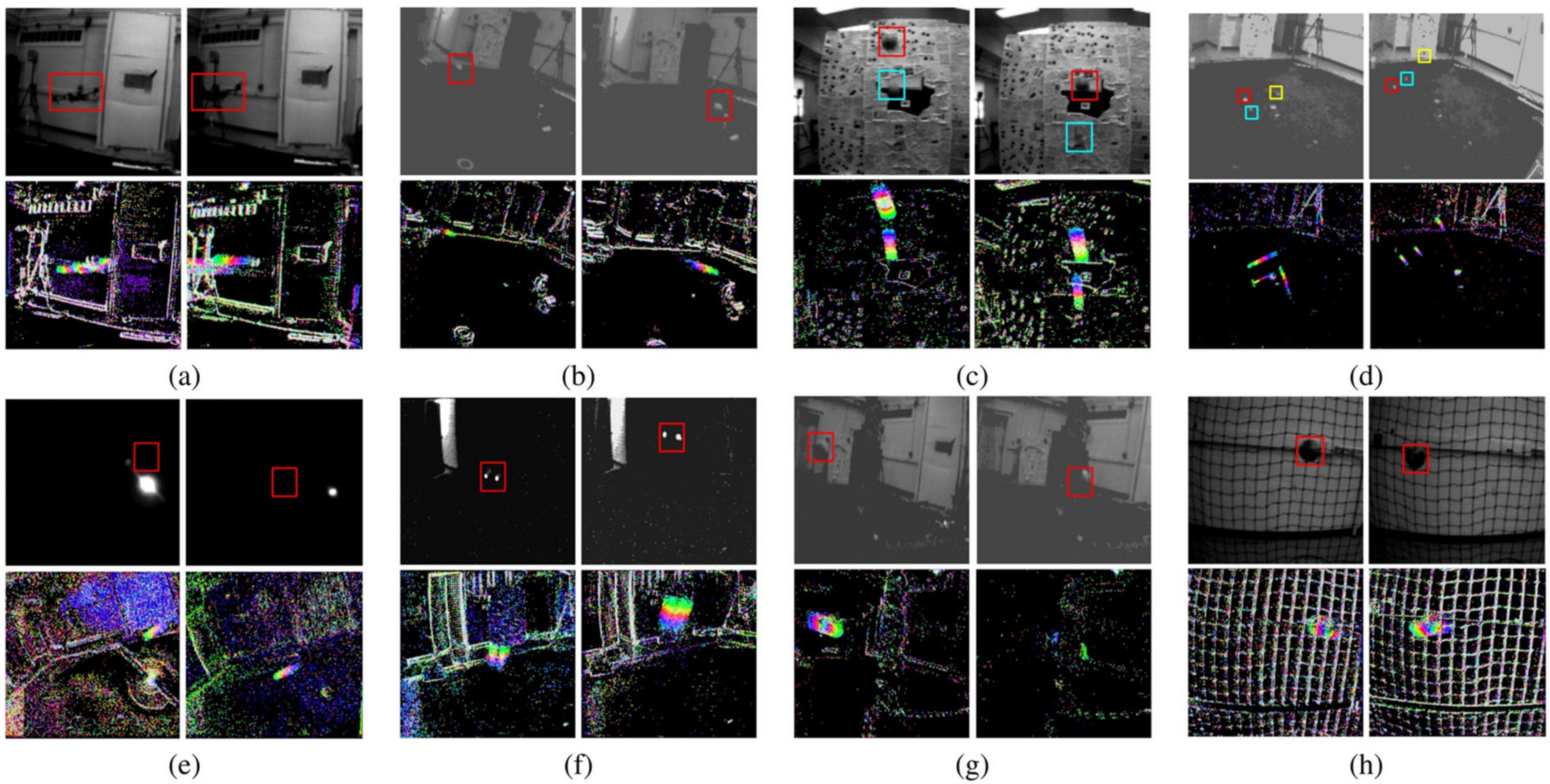
Then, detect points which do not conform to a 4-parameter model



http://prg.cs.umd.edu/BetterFlow.html

A Mitrokhin, C Fermüller, C Parameshwara, Y Aloimonos. Event-based moving object detection and tracking, IROS 2018.

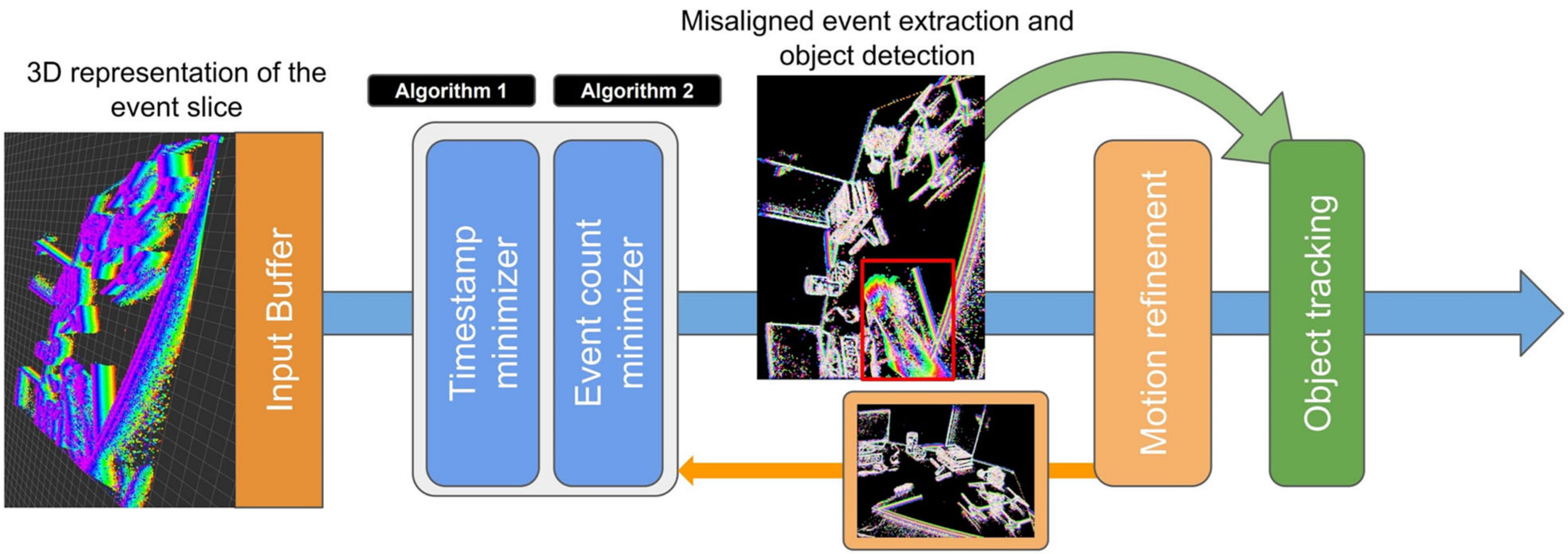
Dataset



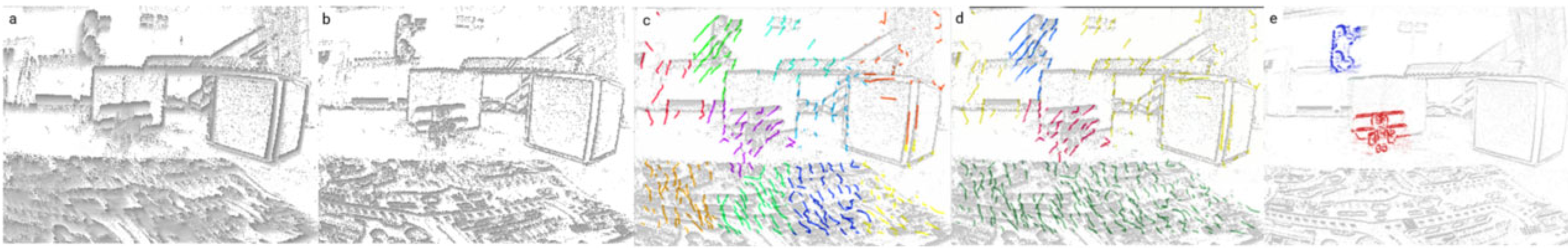
Fast motion, Multiple Objects, Lighting Variations, Occlusion

Results

Algorithm

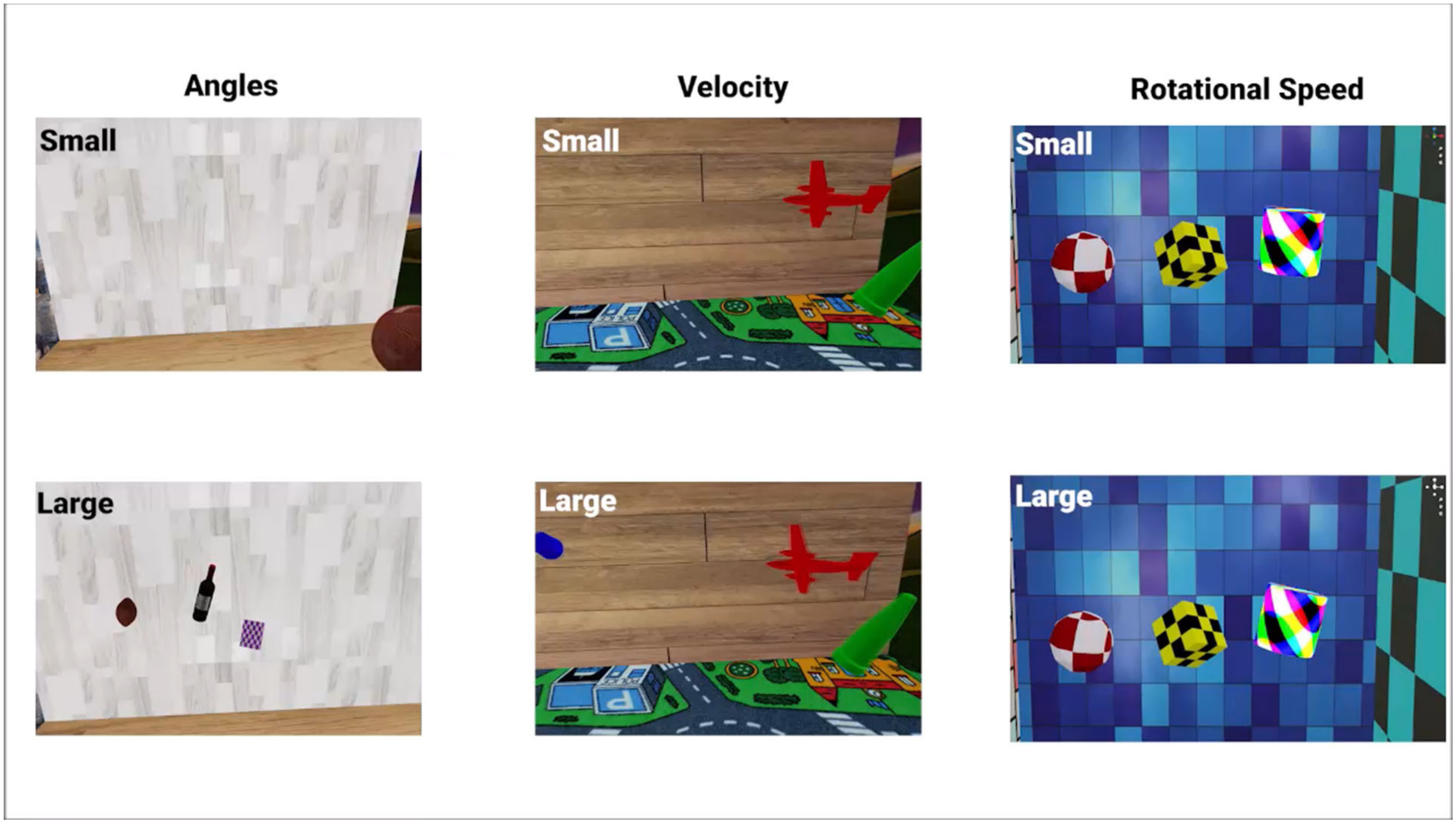

3D representation of the event slice
Algorithm 1
Algorithm 2
Misaligned event extraction and object detection
Input Buffer
Timestamp minimizer
Event count minimizer
Motion refinement
Object tracking

Improved Segmentation



(a) event cloud (b) after global motion compensation (c) Sparse tracklets on compensated event cloud, (d) Merged feature clusters (e) Output

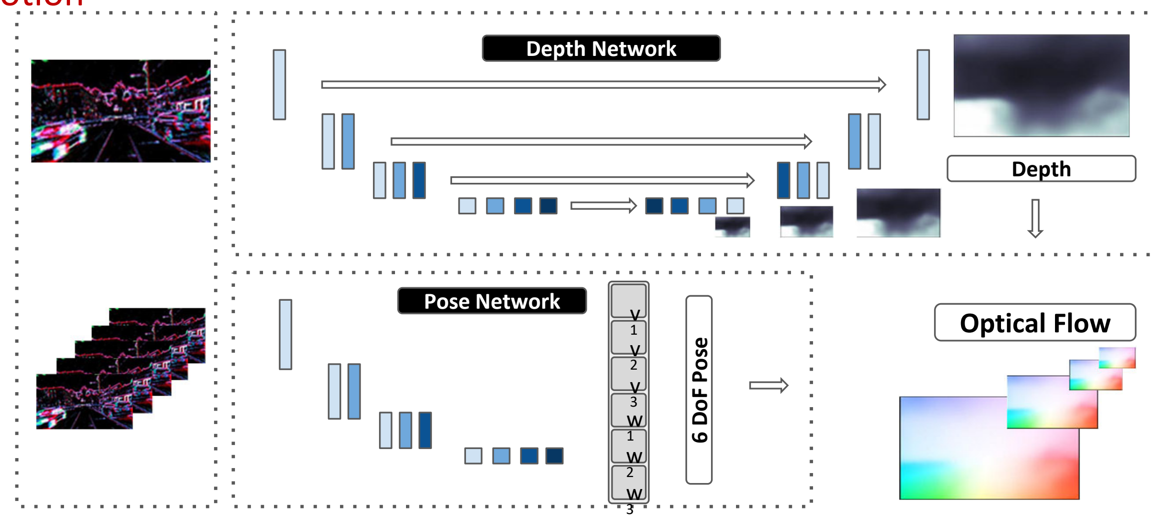
C. Parameshwara, N. Sanket, C.Singh, C. Fermüller, Y. Aloimonos: Zero-Shot Multi-Motion Segmentation With a Monocular Event Camera. arXiv


Angles
Velocity
Rotational Speed
Small
Small
Small
Large
Large
Large

Replace Optimization with Learning:
I: Flow Depth and 3D Motion Estimation

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{Z}\begin{pmatrix} -1 & 0 & x \\ 0 & -1 & y \end{pmatrix}\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} + \begin{pmatrix} xy & -1-x^2 & y \\ 1+y^2 & -xy & -x \end{pmatrix}\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$$

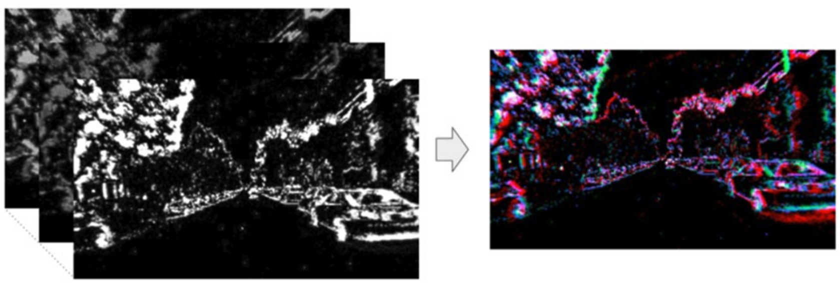
flow depth translation 3D motion rotation

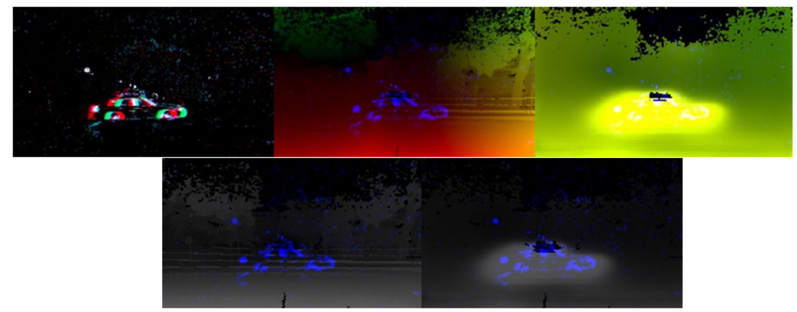


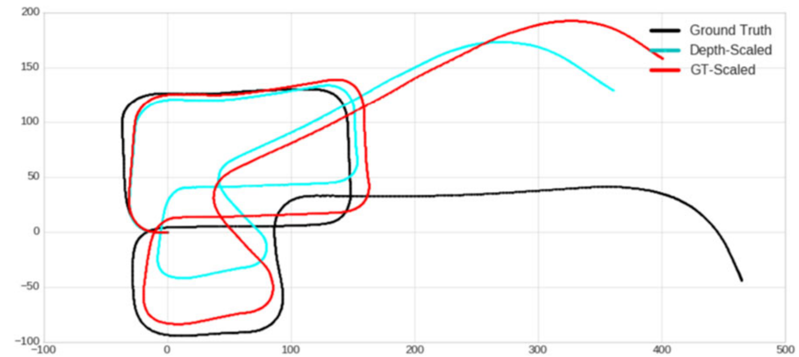
C. Ye, A. Mitrokhin, C. Fermüller, JA. Yorke, and Y. Aloimonos, "Unsupervised Learning of Dense Optical Flow, Depth and Egomotion with Event-Based Sensors," IROS, 2020.

Highlights

Unsupervised Learning of Dense Optical Flow, Depth and Egomotion from Sparse Event Data





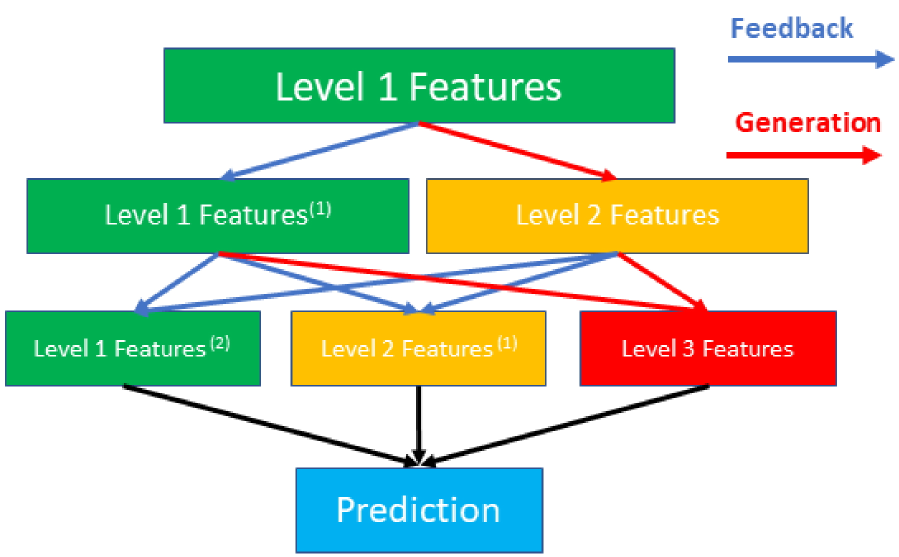


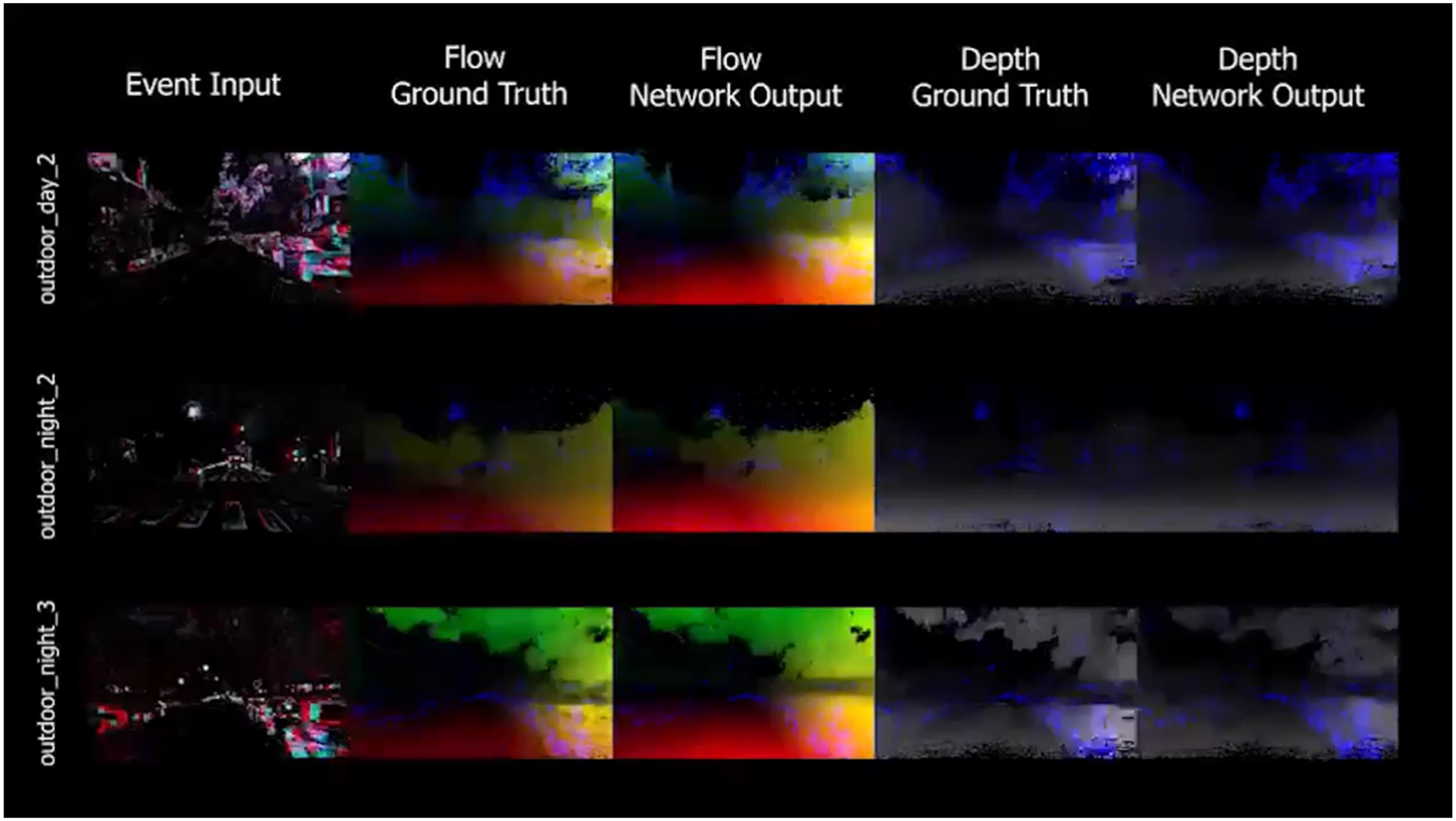
- Transfers from day to night!
- Fixes data sparsity
- Good results

http://prg.cs.umd.edu/ECN.html

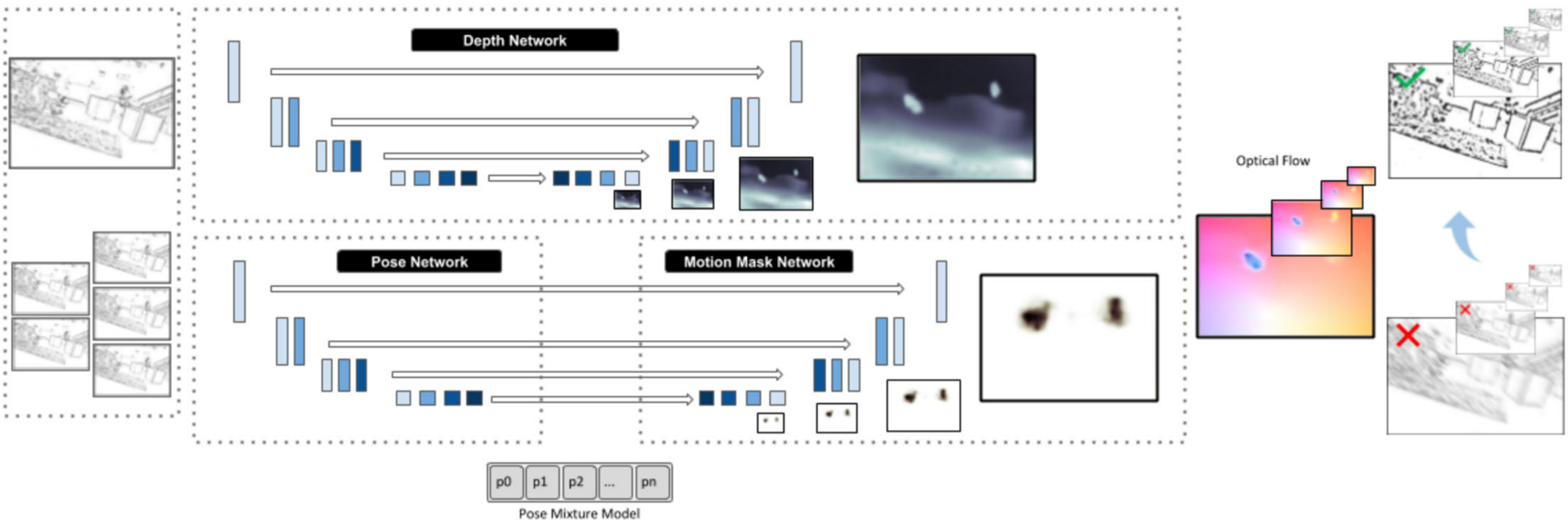
II. Highlights

- A new light-weight architecture ECN



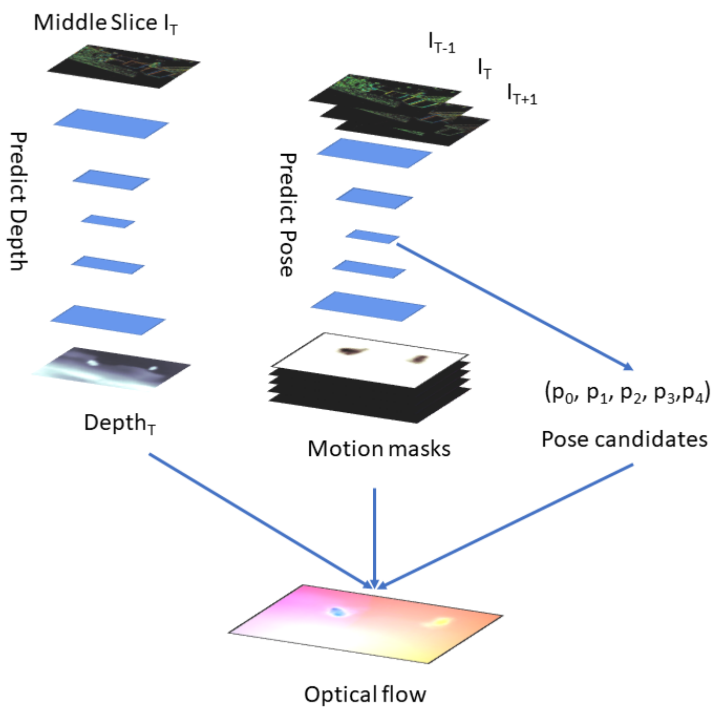

Event Input
Flow
Ground Truth
Flow
Network Output
Depth
Ground Truth
Depth
Network Output
outdoor_day_2
outdoor_night_2
outdoor_night_3

II:EV-IMO: Motion Segmentation Dataset and Learning Pipeline for Event Cameras

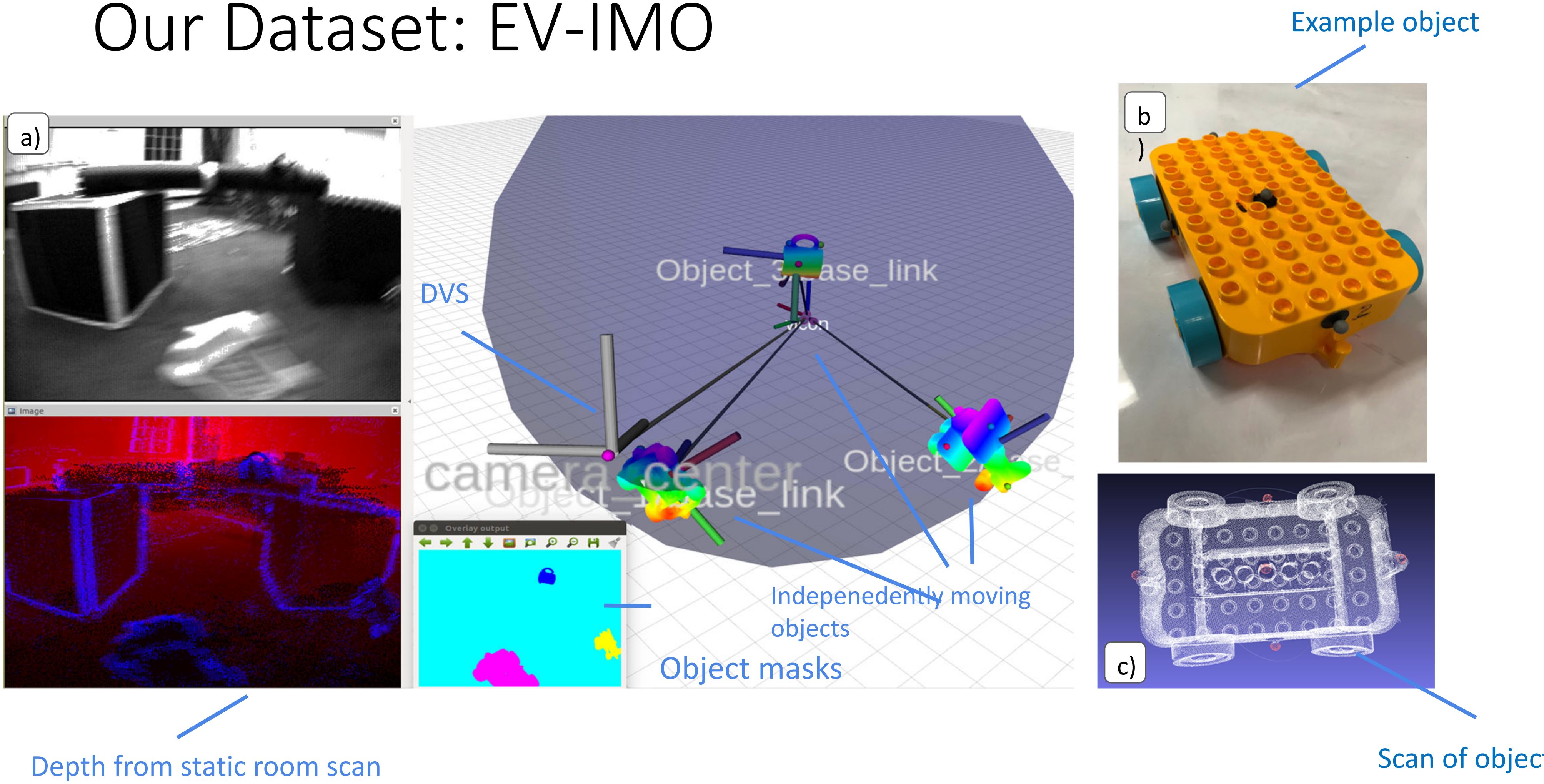


Ye, C., Mitrokhin, A., Fermüller, C., Aloimonos Y and Delbruck, T. "EV-IMO: Motion Segmentation Dataset and Learning Pipeline for Event Cameras." IROS, 2019.

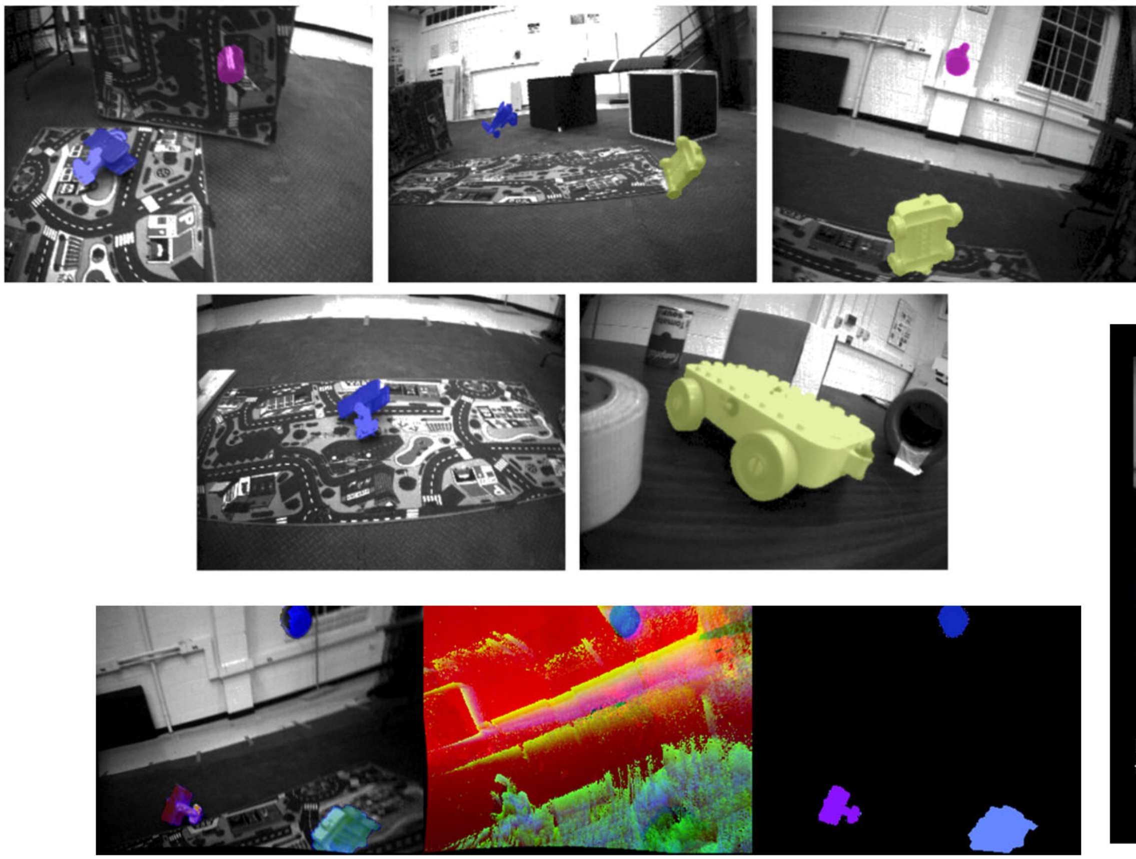
Using motion masks to learn a pose mixture model



Our Dataset: EV-IMO

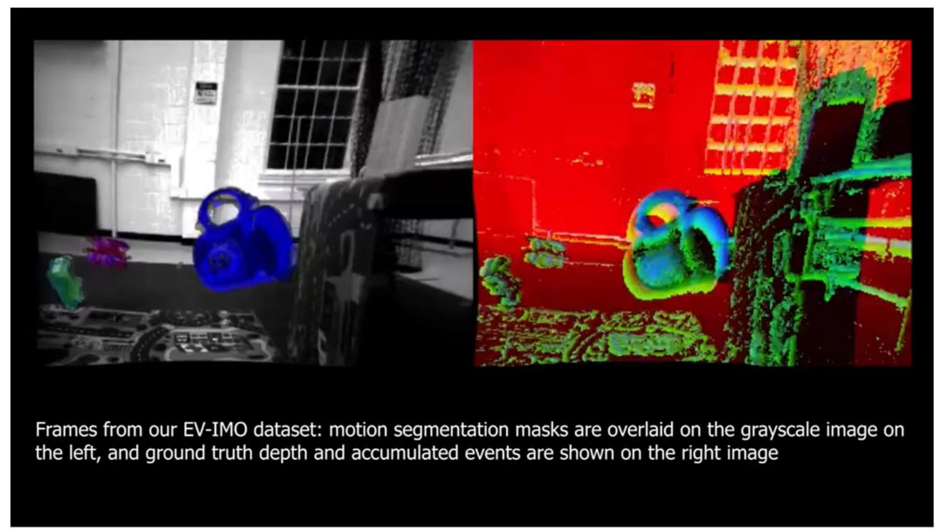


Our Dataset: EV-IMO



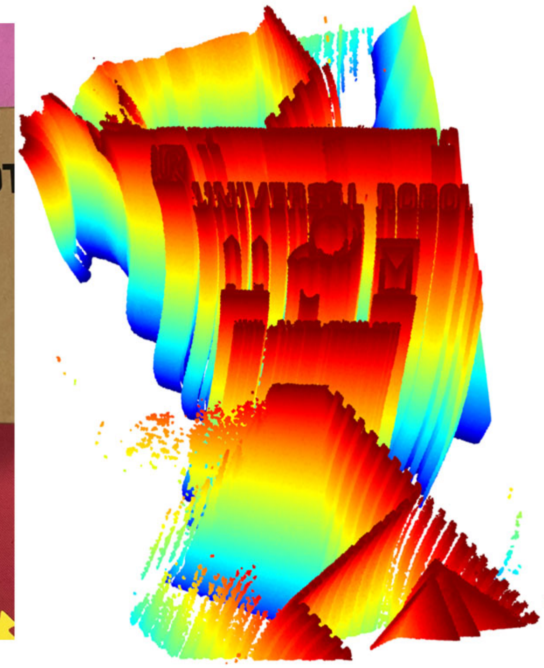
First dataset featuring

- Pixelwise object masks
- Depth ground truth
- Object and Camera trajectories



Frames from our EV-IMO dataset: motion segmentation masks are overlaid on the grayscale image on the left, and ground truth depth and accumulated events are shown on the right image

Newest set-up:



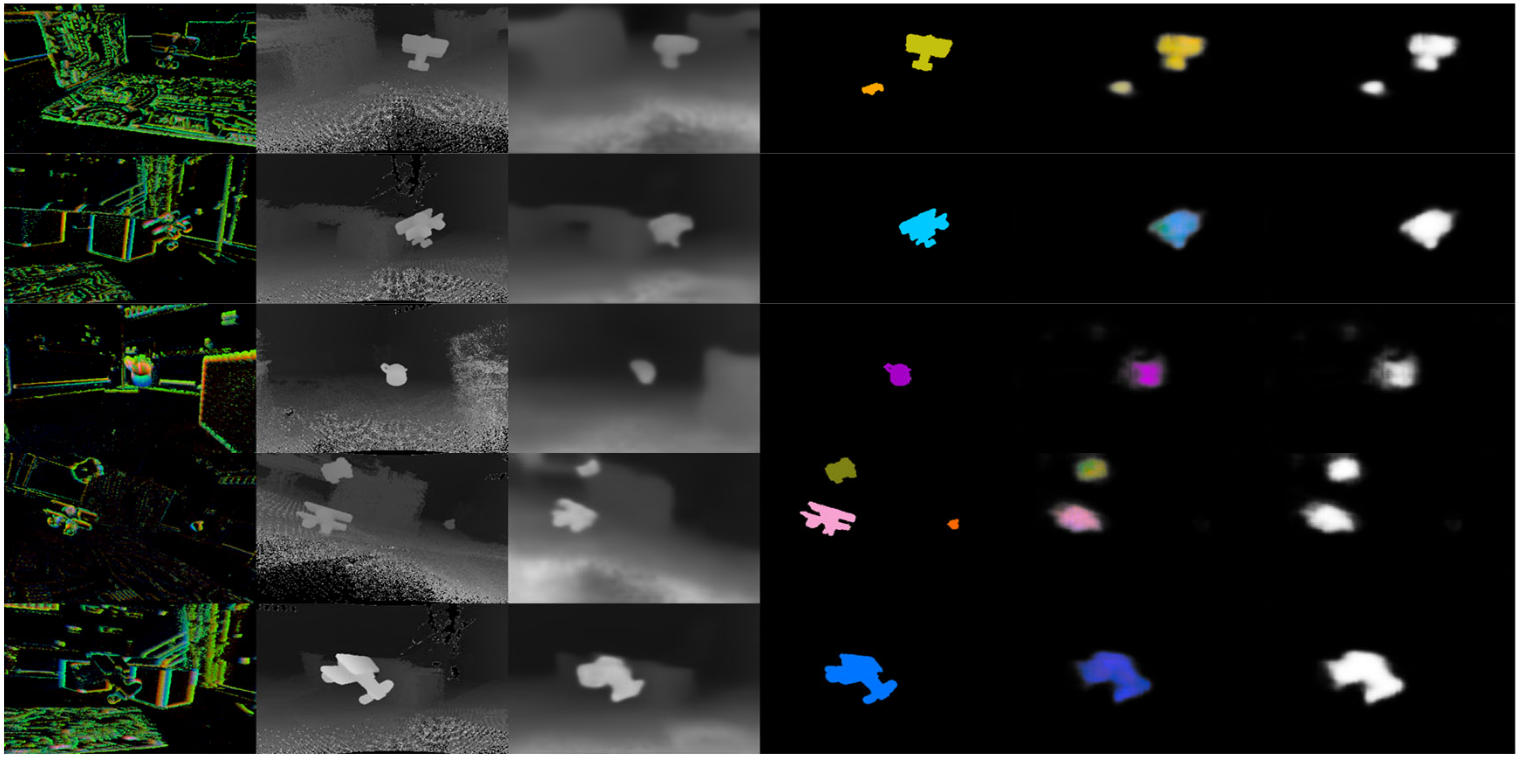
New setup:

- 2x Prophesee 640x480 sensors (stereo)
- Samsung 640x480 sensor
- Prophesee 480x320 sensor (with grayscale)

- Better image quality
- Better calibration
- Diverse objects

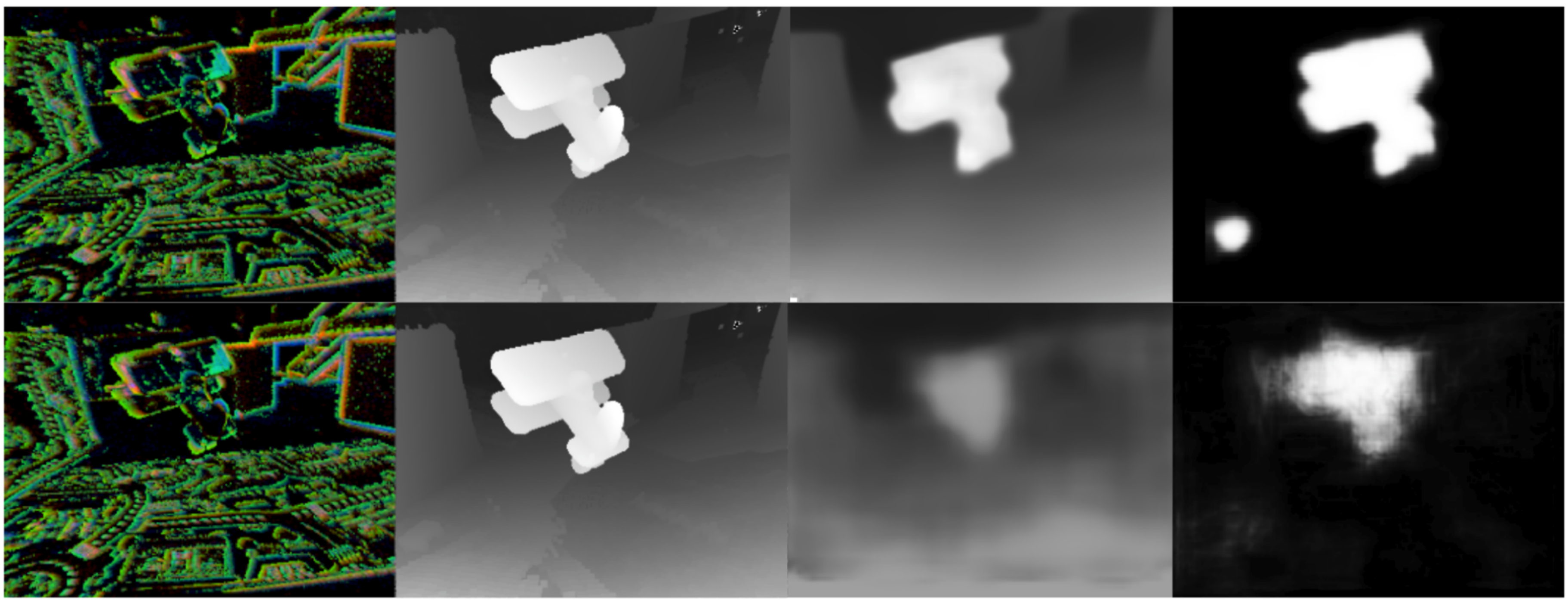
Scene Motion With Event-Based Vision: Learning (II)

- First Work ever to estimate 3D Object Motion and Evaluate it.
- Supervised (mask and depth)
- Warping done on tiny subslices (closer to 3D)



http://prg.cs.umd.edu/EV-IMO.html

Comparison of full and small network
(2000K Vs 40K parameters)



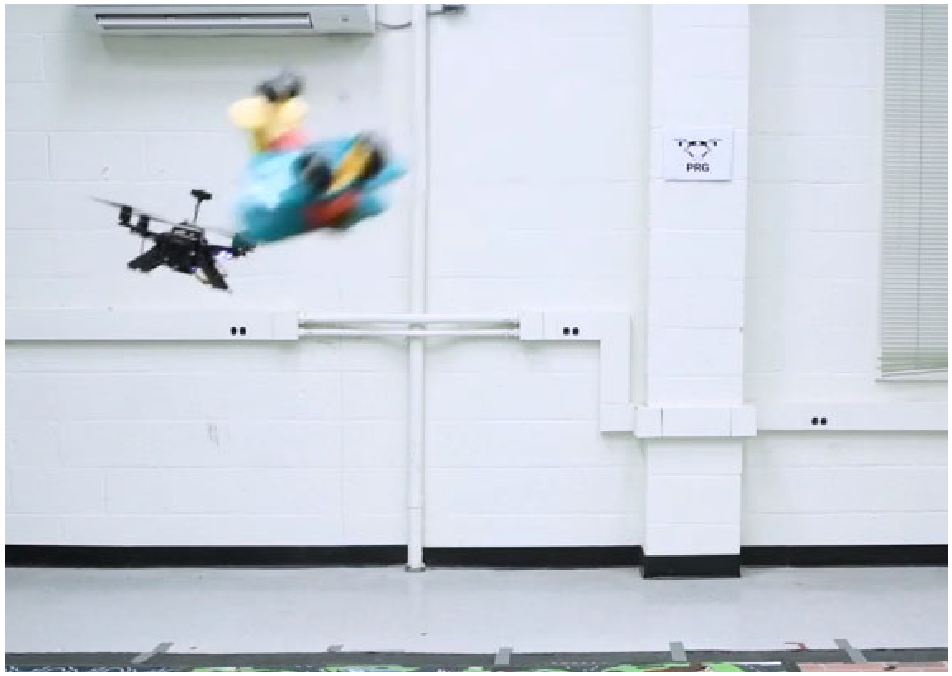
full

small

Event image Ground Truth Depth Estimated Depth Estimated mask

Results

EVDodge

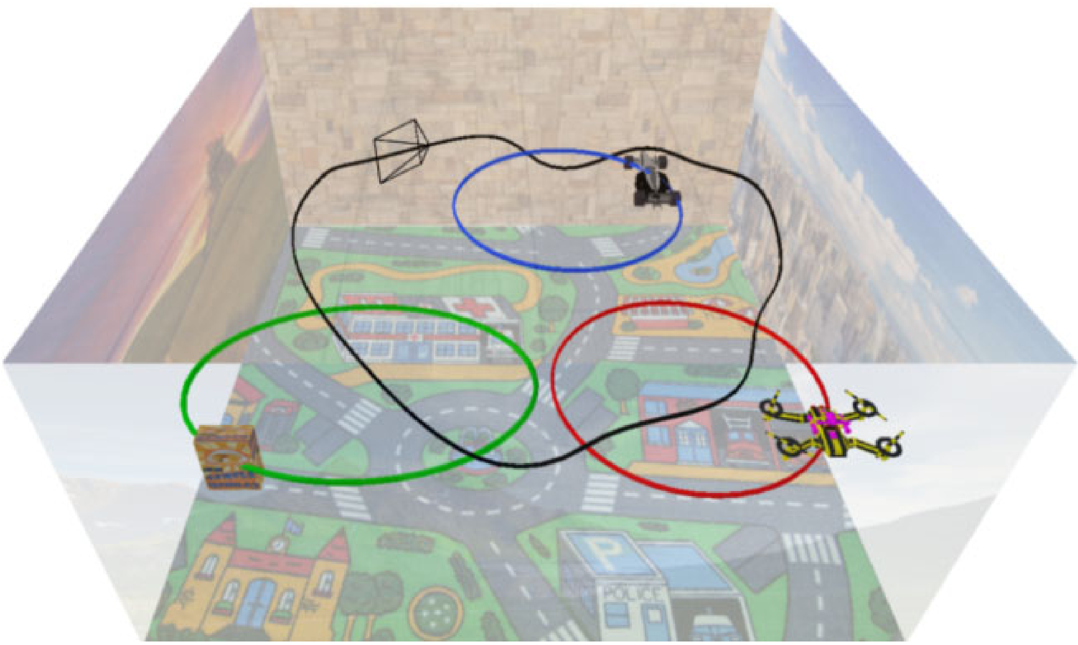


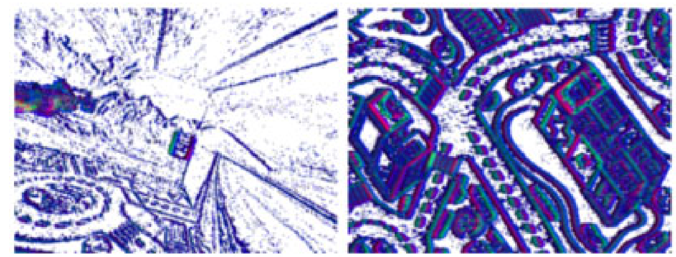
Camera equipped with down- and front-facing DVS, down facing sonar and IMU

All computations done online on a NVIDIA TX2 CPU+GPU

N. Sanket , C.. Parameshwara , C.D.Singh, A.. Kuruttukulam , C., Fermüller, D. Scaramuzza , Y. Aloimonos .
EVDodge: Embodied AI For High-Speed Dodging On A Quadrotor Using Event Cameras. ICRA, 2020
.

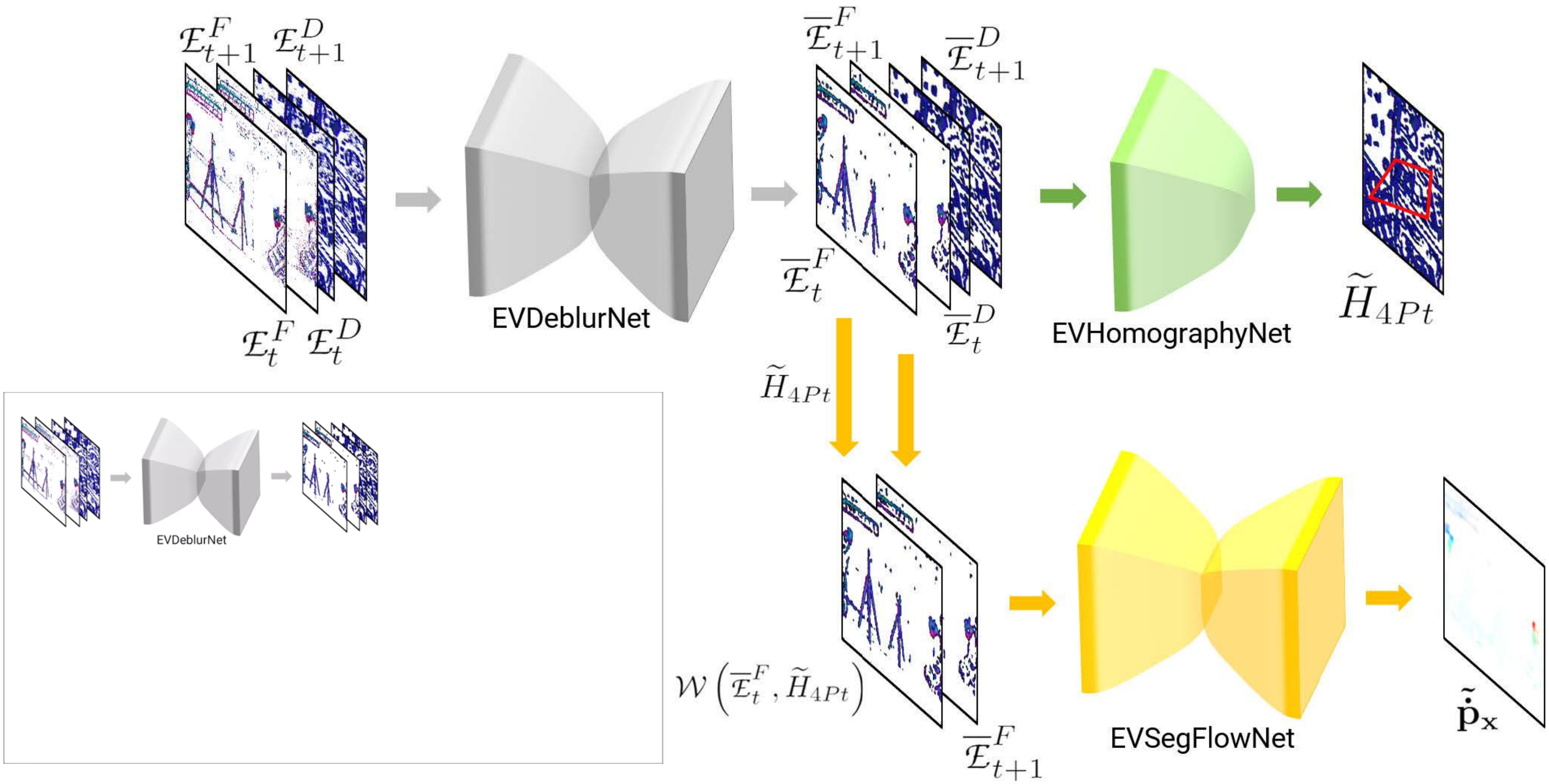
Training in Simulation Environment





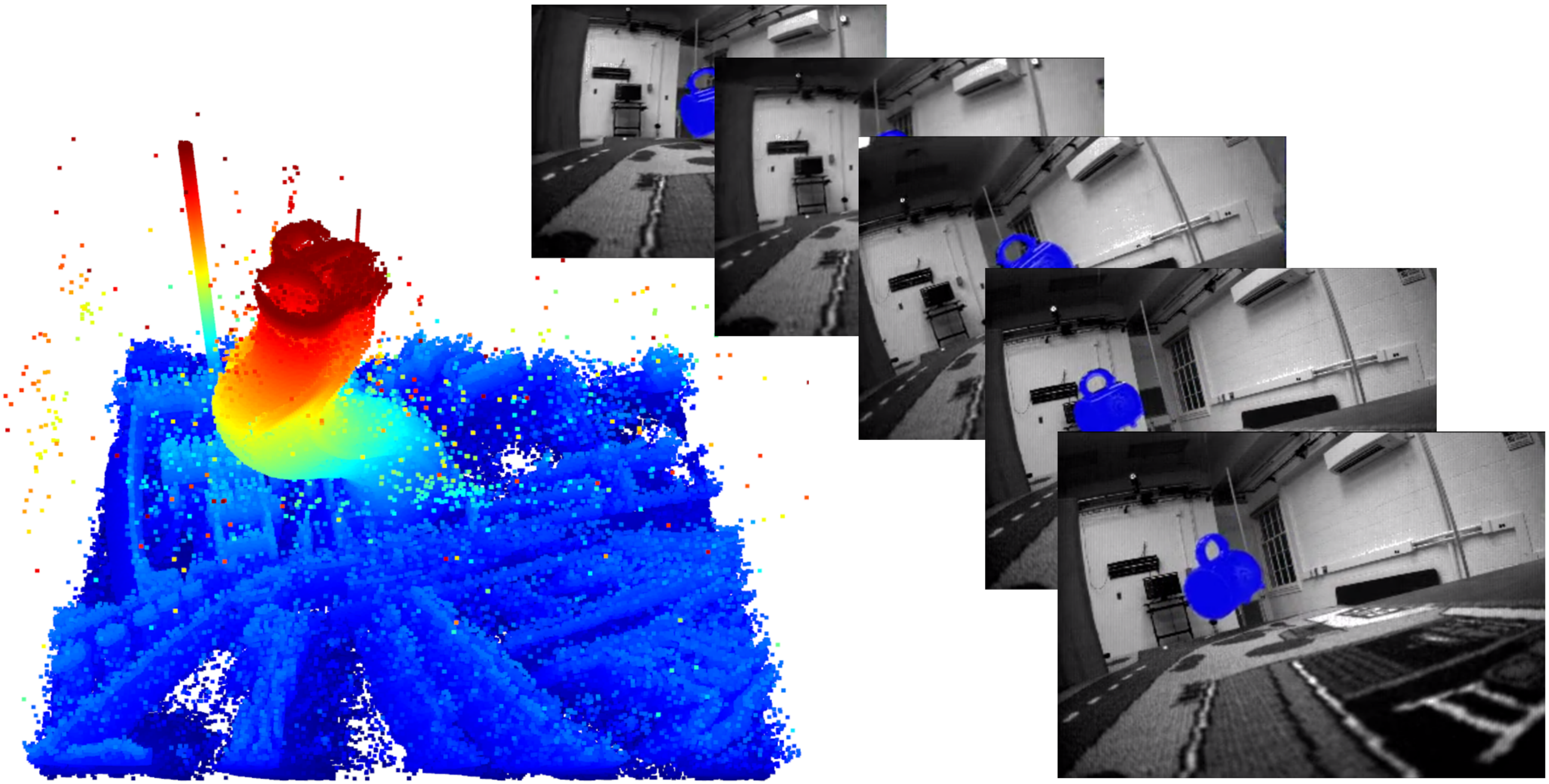
Front and Down-facing simulated events

AI Navigation Stack for Dodging Objects



Front Camera
3D Plot
1.5
1
0.5
0
2
0
-2
-2
0
2
4
Down Camera
PRG
BUS

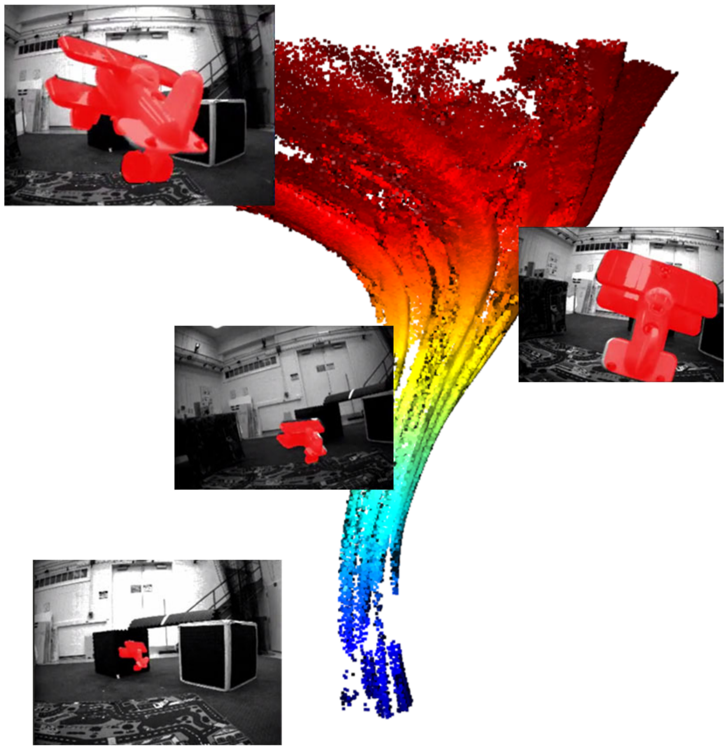
Event Surfaces as a Geometrical Problem



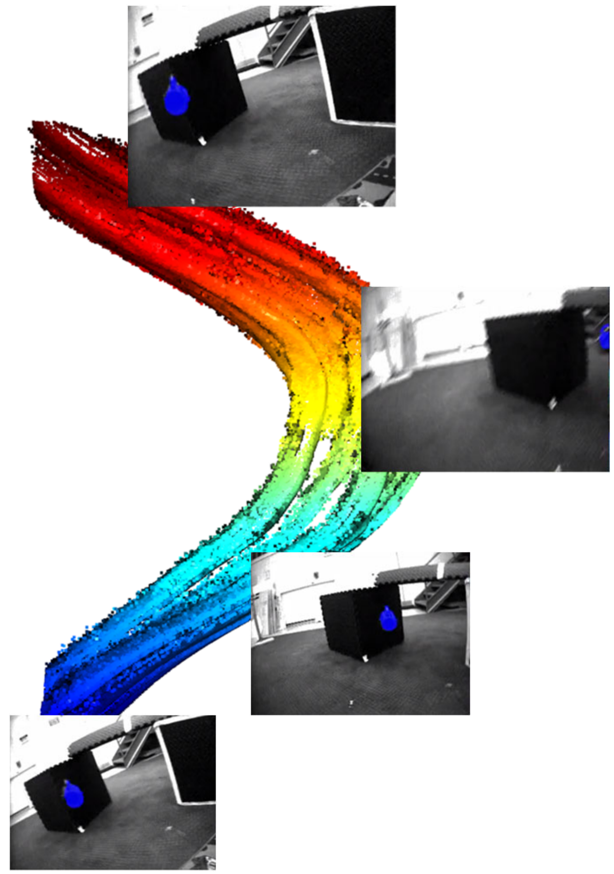
Event Data in 3D

- Motion segmentation is prone to errors when the variation in speed of objects is high
- What we observe is not the speed but the shift in pixels; it becomes greater over large intervals of time
- Objects occlude one another during motion - leaving distinct artifacts in *(x,y,t)* space

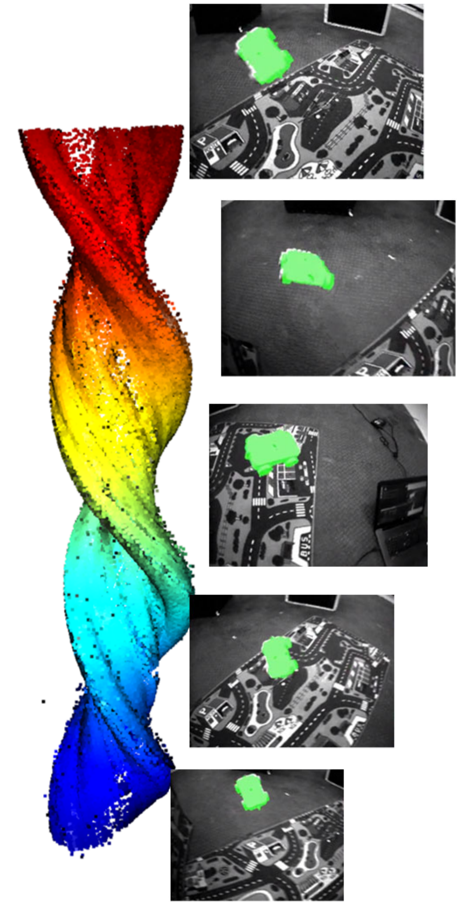
Event Data in 3D



Z-axis translation



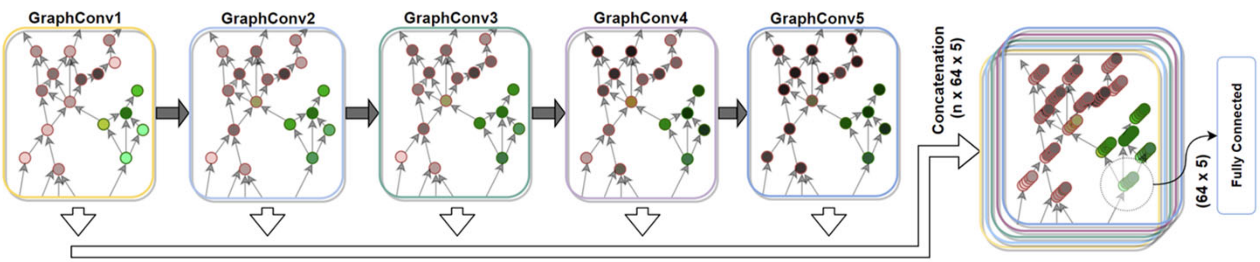
X/Y-axis translation



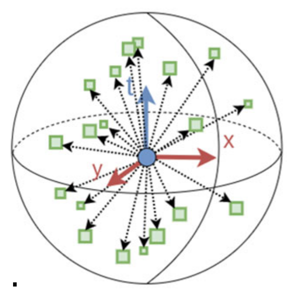
Z-axis rotation

Color = time (~2 sec.)

Learning Segmentation in *(x,y,t)*

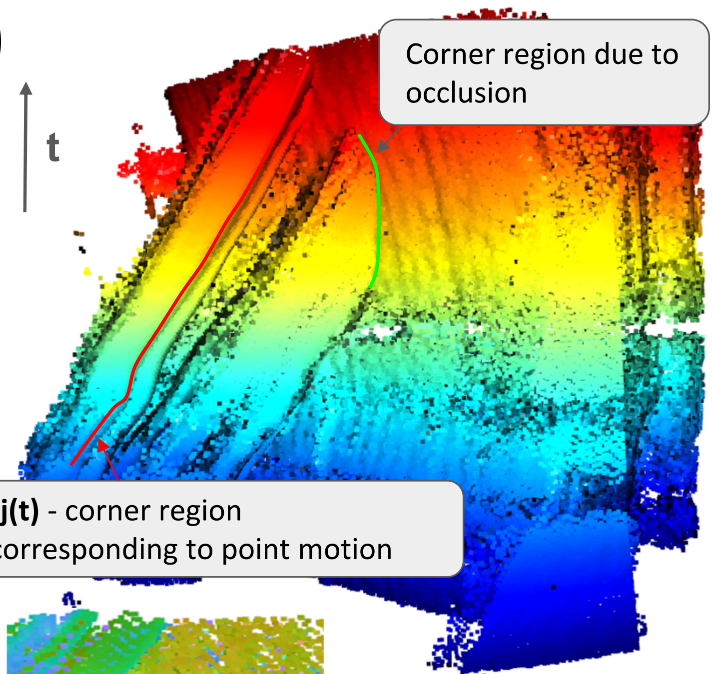


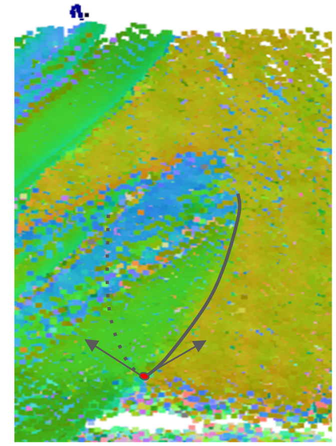
Temporal Augmentation

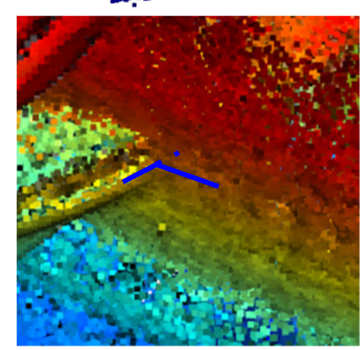


Preprocessing:

- Filtering (radius outlier)
- Normal estimation
- Subsampling
- Graph computation







Mitrokhin et al. "Learning Visual Motion Segmentation Using Event Surfaces." CVPR, 2020

Graph Conv Network - Inference

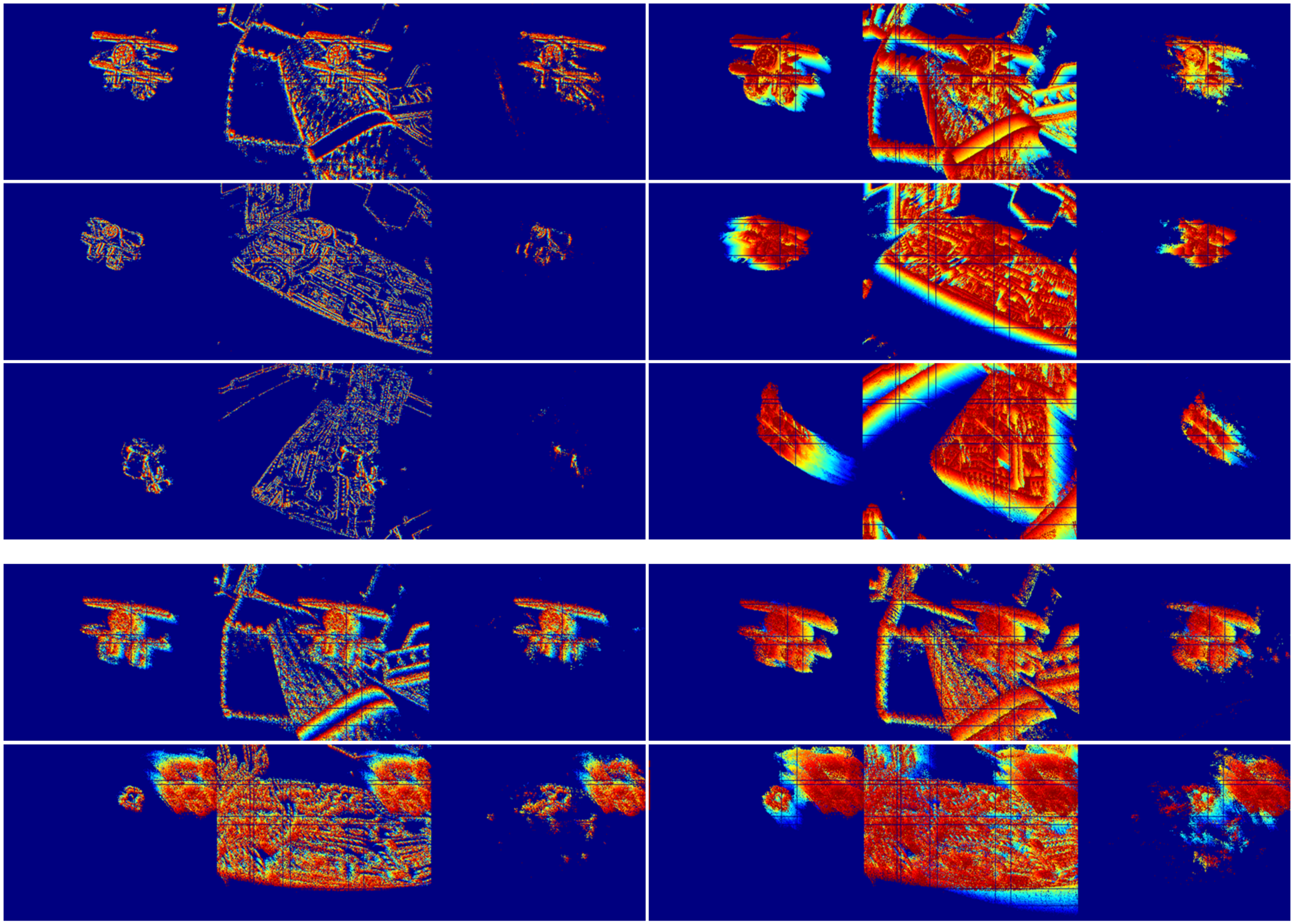
ground truth	input	inference	ground truth	input	inference

no subsampling, w=0.02

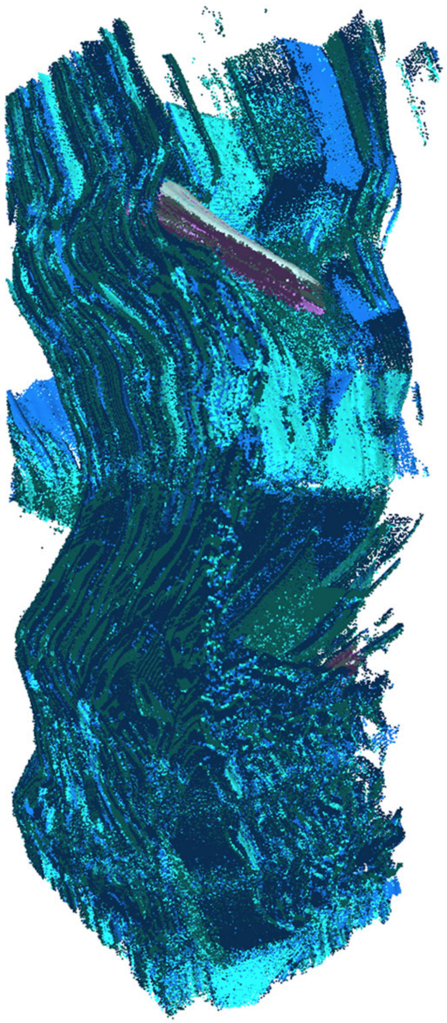
no subsampling, w=0.3

Subsampling x2, w=0.1

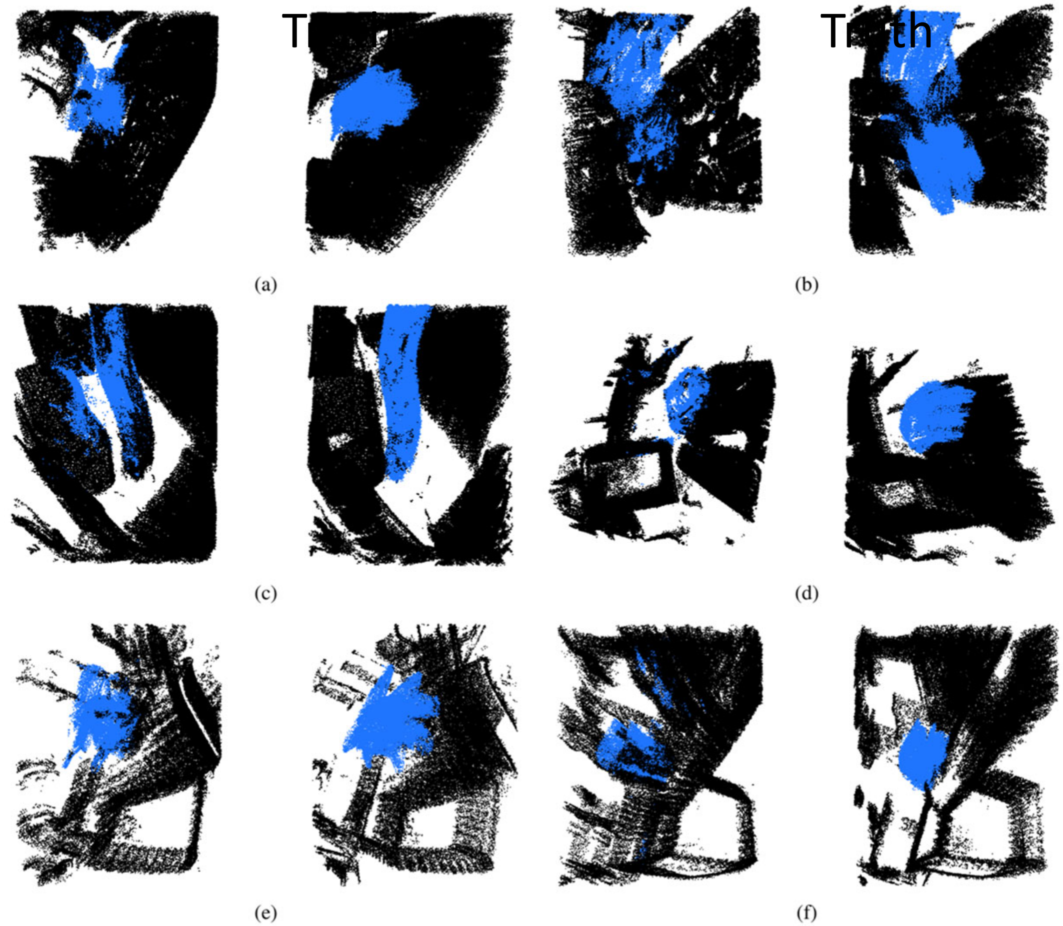
Subsampling x2, w=0.3



Graph Conv Network - Inference



Input Sequence



Next Steps:
Develop the constraints for event-surfaces

Constraint #1:

$$0 = \begin{bmatrix} \dot{\text{x}} & \dot{\text{y}} & 1 \end{bmatrix}^T \cdot \begin{bmatrix} n_x & n_y & n_z \end{bmatrix} \implies \begin{bmatrix} -\frac{n_x}{n_z} & -\frac{n_y}{n_z} \end{bmatrix} \dot{x} = 1$$

Optical flow equation:

$$\boxed{\dot{x} = A(\Omega b + \frac{V}{\text{Z}})}$$

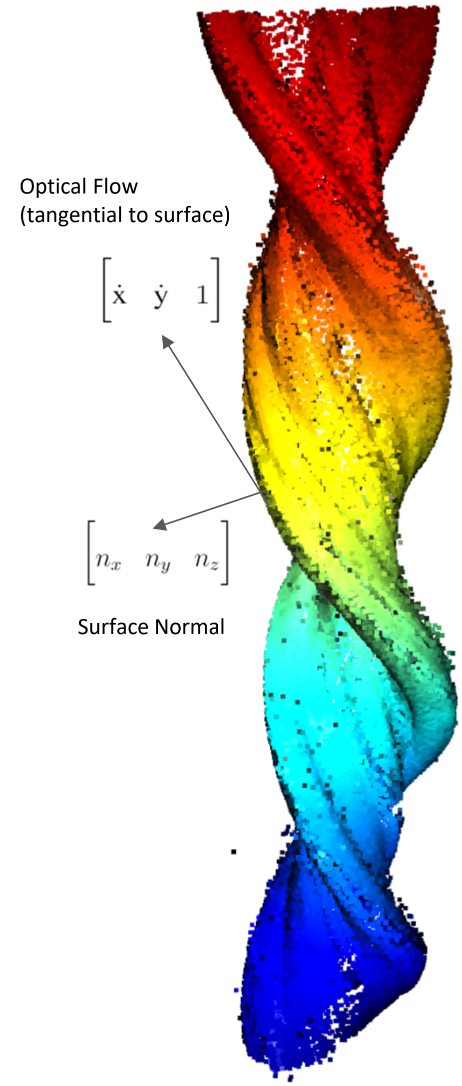
$$A = \begin{bmatrix} 1 & 0 & -\text{x} \\ 0 & 1 & -\text{y} \end{bmatrix} \qquad \Omega = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix}^T_\times = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}$$

$$b = \begin{bmatrix} \text{x} & \text{y} & 1 \end{bmatrix}^T$$

$$c = \begin{bmatrix} -\frac{n_x}{n_z} & -\frac{n_y}{n_z} \end{bmatrix}$$

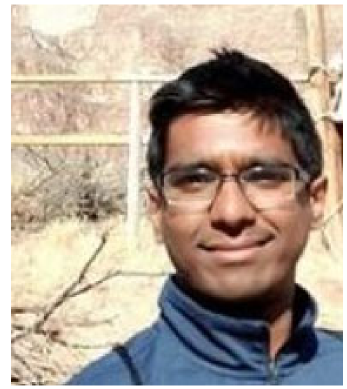
$$\boxed{\dot{x} = A\Omega b + (1 - cA\Omega b)\frac{AV}{cAV}}$$

$$\boxed{\text{Z} = \frac{cAV}{1 - cA\Omega b}}$$



Collaborators

Anton Mitrokhin



Chethan Parameshwara

ChengXi Ye

Francisco Barranco