Visual-Inertial Localization and Mapping for Robot Navigation

Dr. Guillermo Gallego

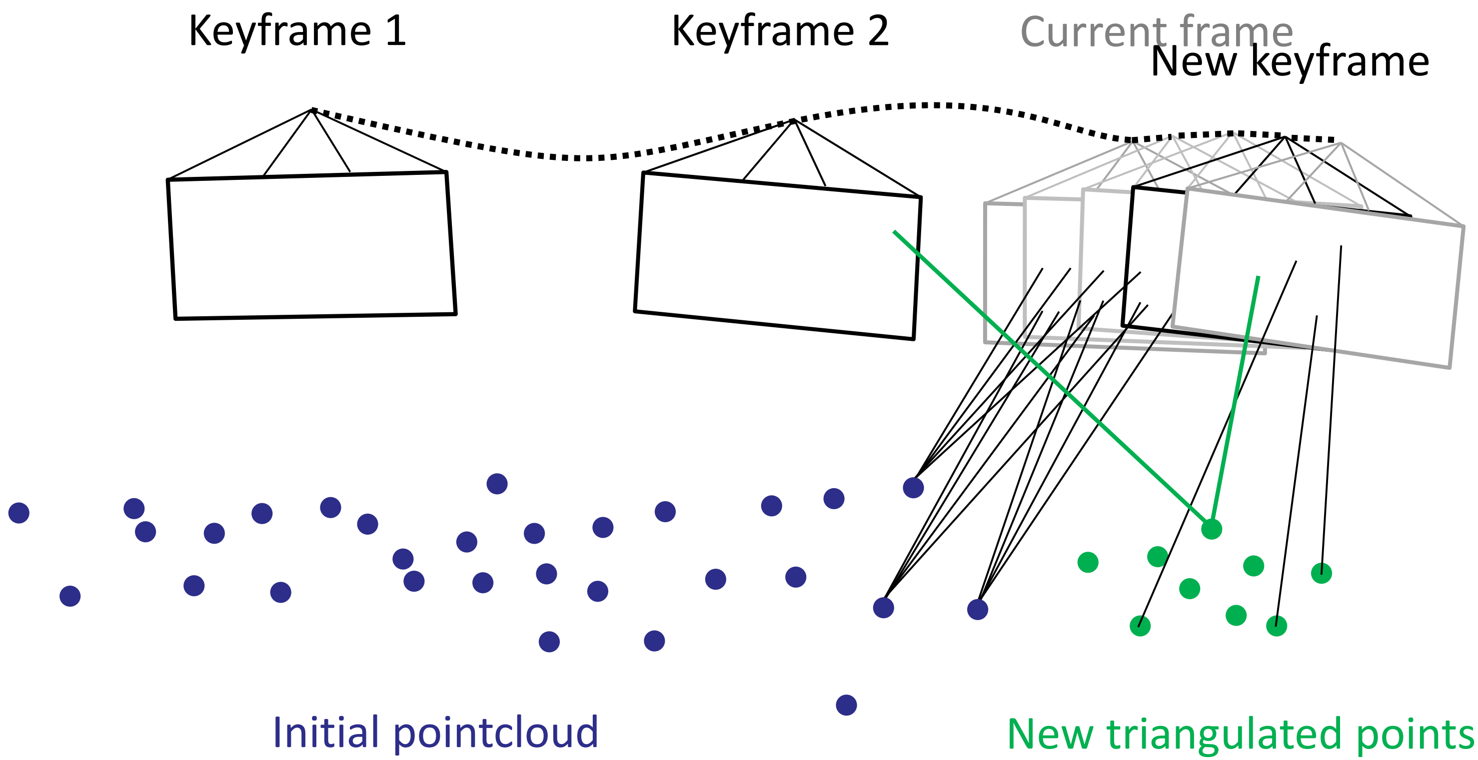
Robotics & Perception Group
University of Zurich

Mocular, Vision-inertial Navigation of Mobile Robots

https://www.youtube.com/watch?v=vHpw8zc7-JQ

[Scaramuzza, Achtelik, Weiss, Fraundorfer, et al., Vision-Controlled Micro Flying Robots: from System Design to Autonomous Navigation and Mapping in GPS-denied Environments, IEEE RAM, 2014]

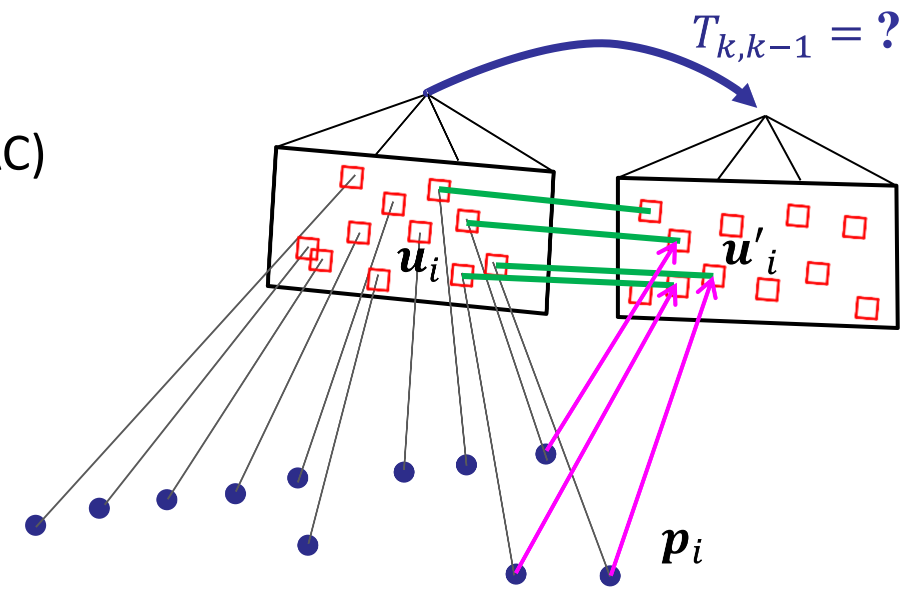
Keyframe-based Visual Odometry



PTAM (Parallel Tracking & Mapping) [Klein, ISMAR'08]

Feature-based methods

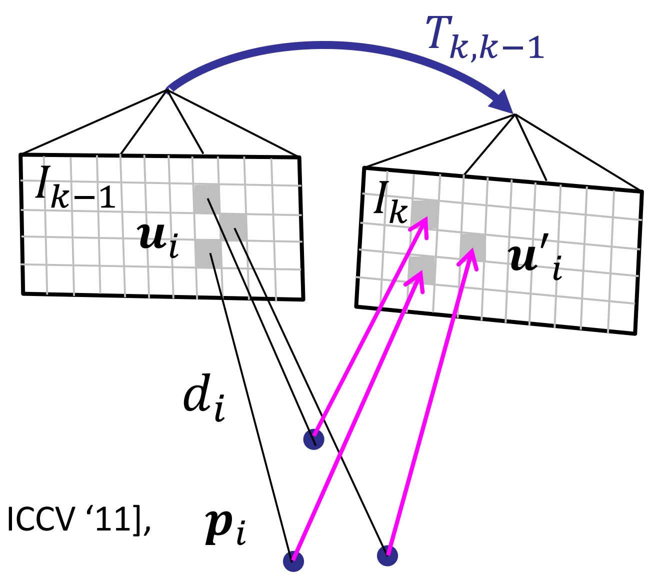
1. Extract & match features (+RANSAC)
2. Minimize **Reprojection error** minimization



$$T_{k,k-1} = \arg\min_{T} \sum_i \| \boldsymbol{u}'_i - \pi(\boldsymbol{p}_i) \|^2_{\Sigma}$$

Direct methods

1. Minimize **photometric error**



$$T_{k,k-1} = \arg\min_{T} \sum_i \| I_k(\boldsymbol{u}'_i) - I_{k-1}(\boldsymbol{u}_i) \|^2_{\sigma}$$

where $\boldsymbol{u}'_i = \pi\big(T \cdot (\pi^{-1}(\boldsymbol{u}_i) \cdot d)\big)$

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

Feature-based methods

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

$$T_{k,k-1} = \arg\min_{T} \sum_{i} \| \boldsymbol{u}'_i - \pi(\boldsymbol{p}_i) \|_{\Sigma}^{2}$$

✓ Large frame-to-frame motions

✗ Slow due to costly feature extraction and matching

✗ Matching Outliers (RANSAC)

Direct methods

1. Minimize **photometric error**

$$T_{k,k-1} = \arg\min_{T} \sum_{i} \| I_k(\boldsymbol{u}'_i) - I_{k-1}(\boldsymbol{u}_i) \|_{\sigma}^{2}$$

$$\text{where} \quad \boldsymbol{u}'_i = \pi\big(T \cdot (\pi^{-1}(\boldsymbol{u}_i) \cdot d)\big)$$

✓ All information in the image can be exploited (precision, robustness)

✓ Increasing camera frame-rate reduces computational cost per frame

✗ Limited frame-to-frame motion

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

Feature-based methods

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

✓ Large frame-to-frame motions

✗ Slow due to costly feature extraction and matching

Our solution:

SVO: *Semi-direct* Visual Odometry [ICRA'14]

Combines feature-based and direct methods

$$T_{k,k-1} = \arg\min_{T} \sum_{i} \| I_k(\boldsymbol{u}'_i) - I_{k-1}(\boldsymbol{u}_i) \|_{\sigma}^{2}$$

where $\boldsymbol{u}'_i = \pi\big(T \cdot (\pi^{-1}(\boldsymbol{u}_i) \cdot d)\big)$

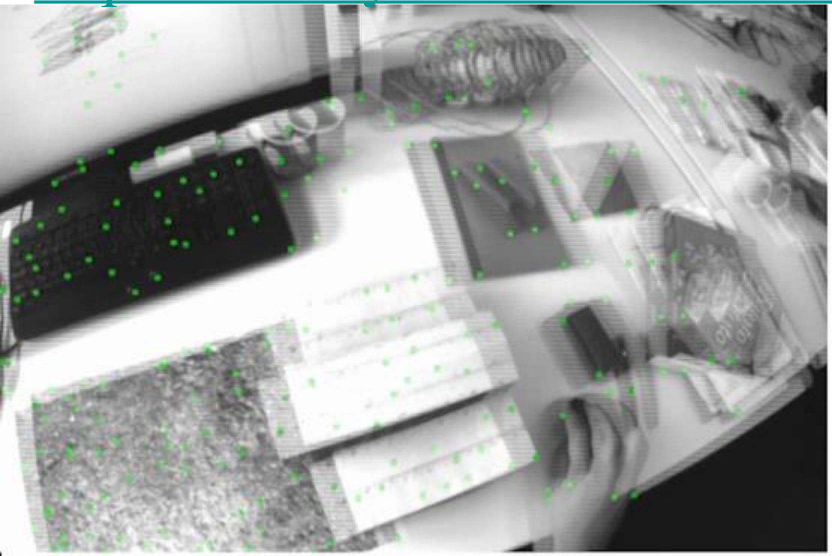
per frame

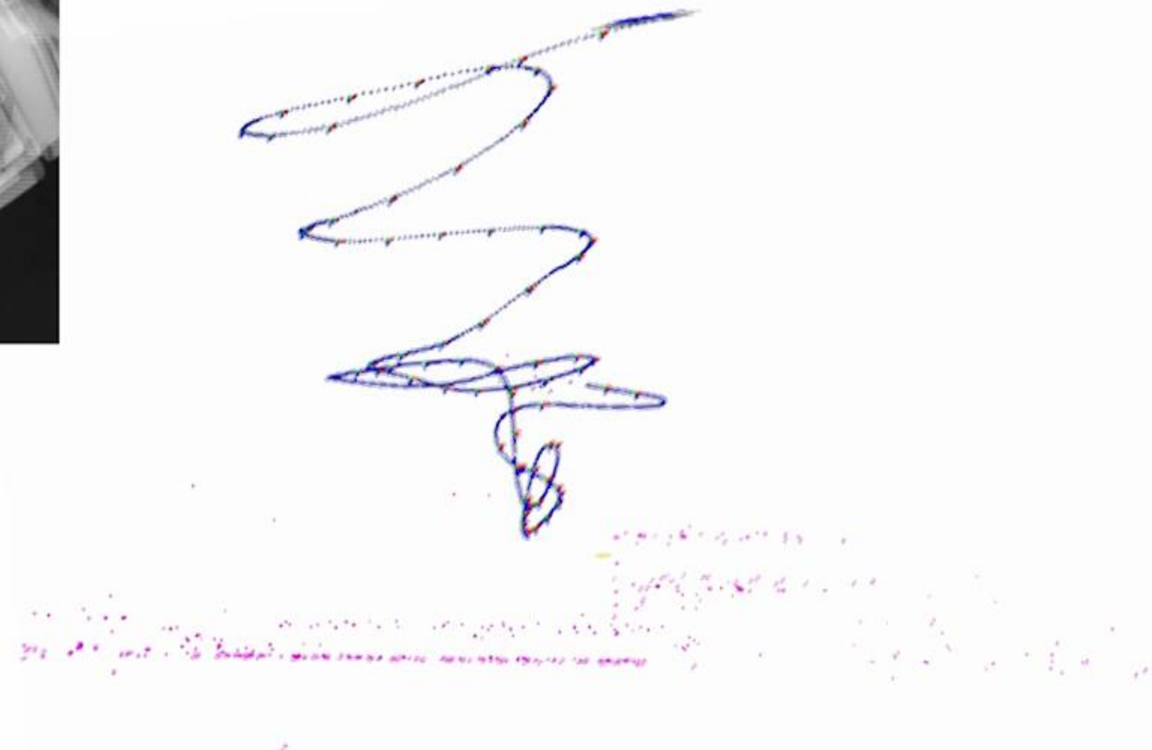
✗ Limited frame-to-frame motion

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

SVO Results: Fast and Abrupt Motions

https://www.youtube.com/watch?v=2YnIMfw6bJY





Open Source

[Forster, Pizzoli, Scaramuzza, «SVO: Semi Direct Visual Odometry», ICRA'14]

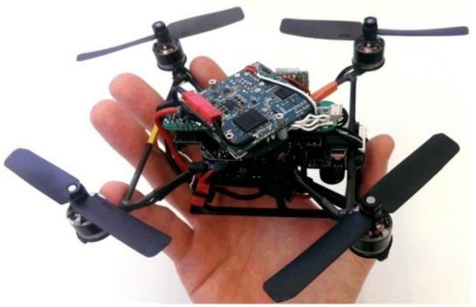
Processing Times of SVO

Laptop (Intel i7, 2.8 GHz)

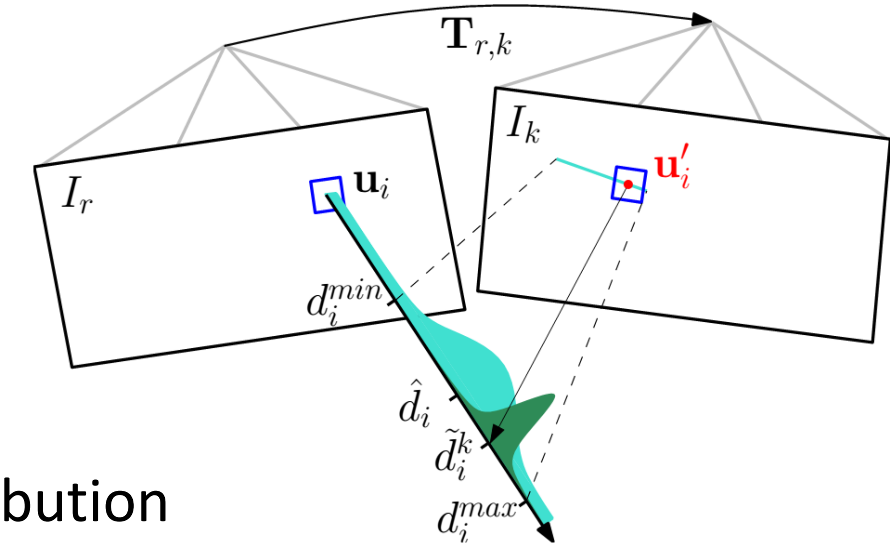
400 frames per second

Embedded ARM Cortex-A9, 1.7 GHz

Up to 70 frames per second



Probabilistic Depth Estimation

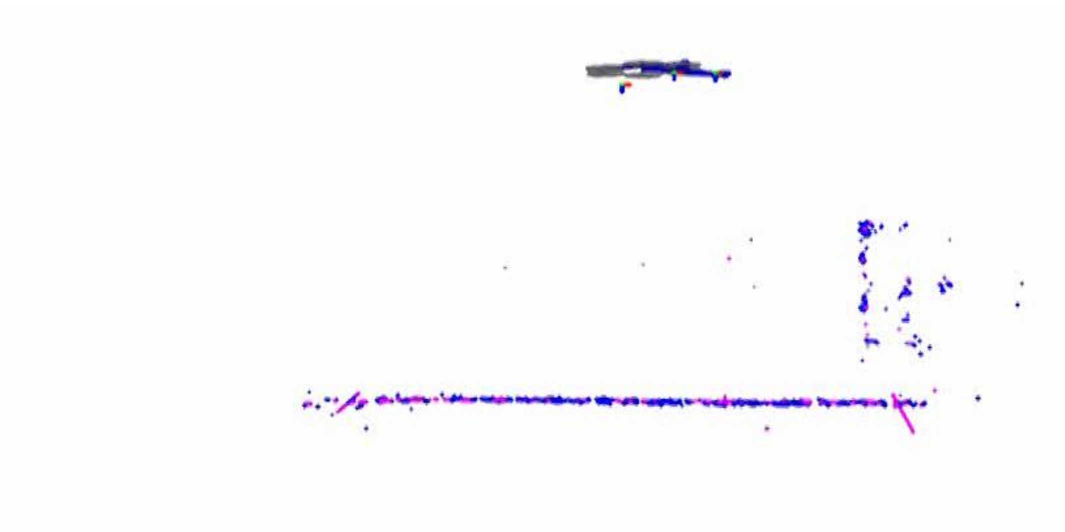


Depth-Filter:

- **Depth Filter** for every feature
- **Recursive Bayesian** depth estimation

Mixture of Gaussian + Uniform distribution

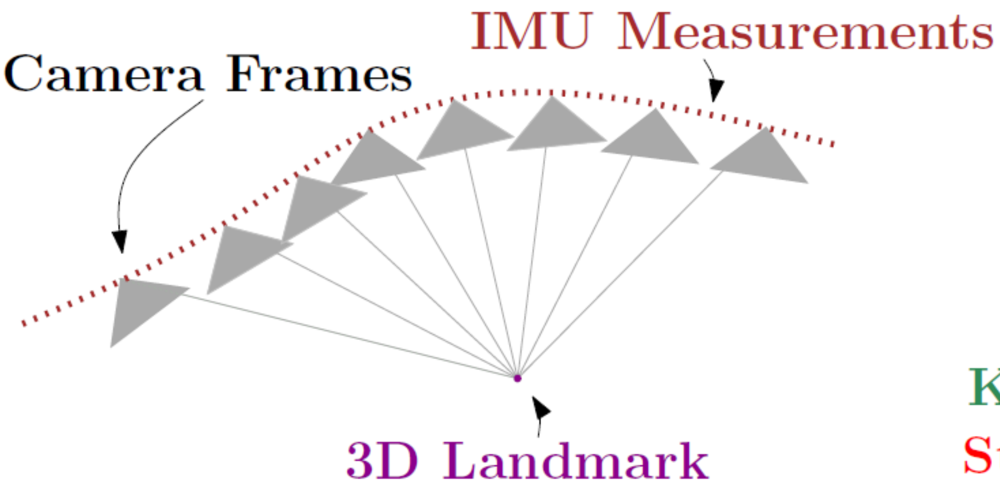
$$p(\tilde{d}_i^k | d_i, \rho_i) = \rho_i \mathcal{N}\left(\tilde{d}_i^k \middle| d_i, \tau_i^2\right) + (1 - \rho_i)\mathcal{U}\left(\tilde{d}_i^k \middle| d_i^{\min}, d_i^{\max}\right)$$

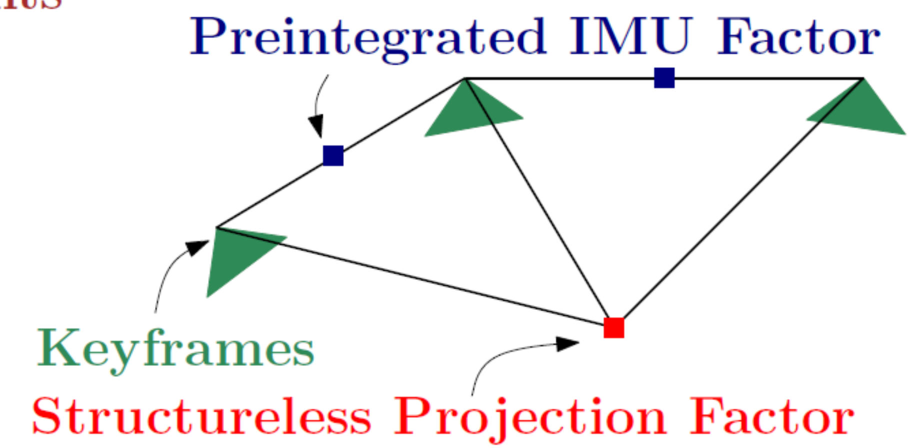


[Forster, Pizzoli, Scaramuzza, SVO: Semi Direct Visual Odometry, IEEE ICRA'14]

Visual-Inertial Fusion via Optimization [RSS'15]

- Fusion solved as a *non-linear optimization problem* (no Kalman filter):
- Increased accuracy over filtering methods





$$\sum_{(i,j)\in\mathcal{K}_k} \|\mathbf{r}_{\mathcal{I}_{ij}}\|^2_{\Sigma_{ij}} + \sum_{i\in\mathcal{K}_k}\sum_{l\in\mathcal{C}_i} \|\mathbf{r}_{\mathcal{C}_{il}}\|^2_{\Sigma_{\mathcal{C}}}$$

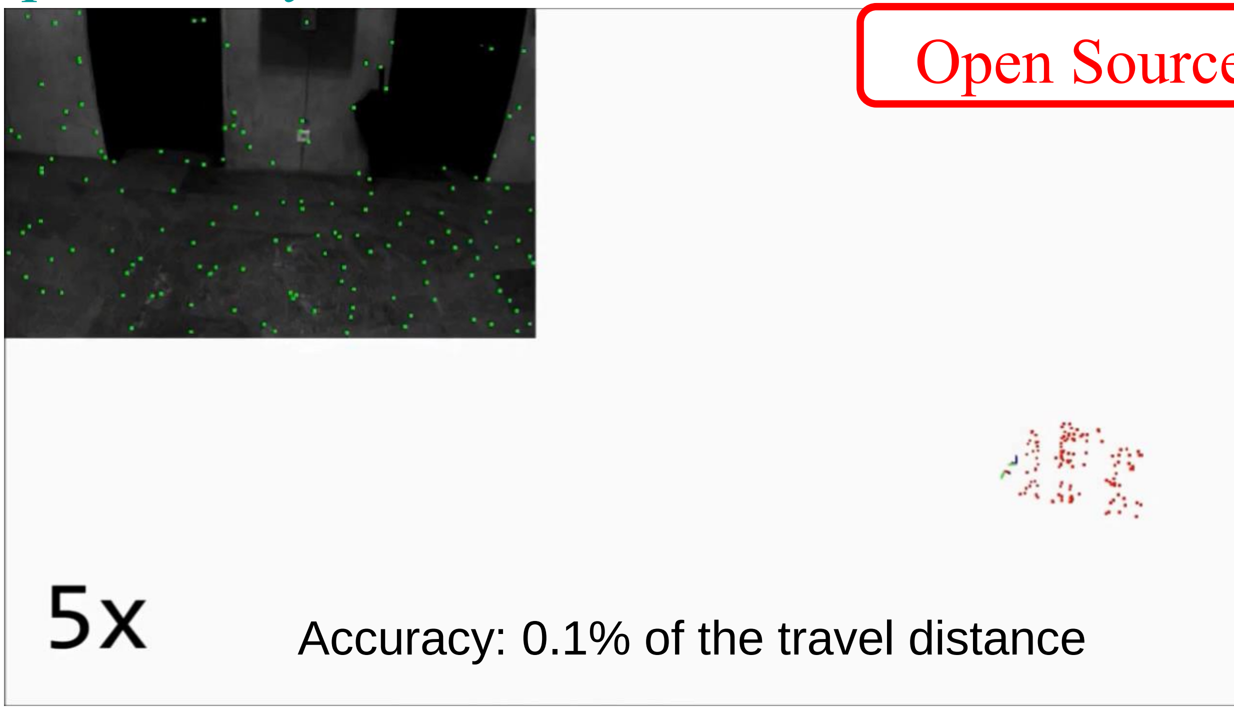
IMU residuals *Reprojection residuals*

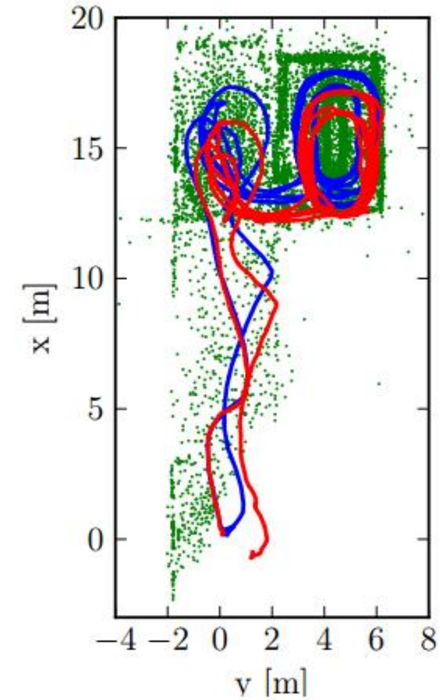
Forster, Carlone, Dellaert, Scaramuzza, IMU Preintegration on Manifold for efficient Visual-Inertial Maximum-a-Posteriori Estimation, *Robotics Science and Systens*'15, **Best Paper Award finalist**

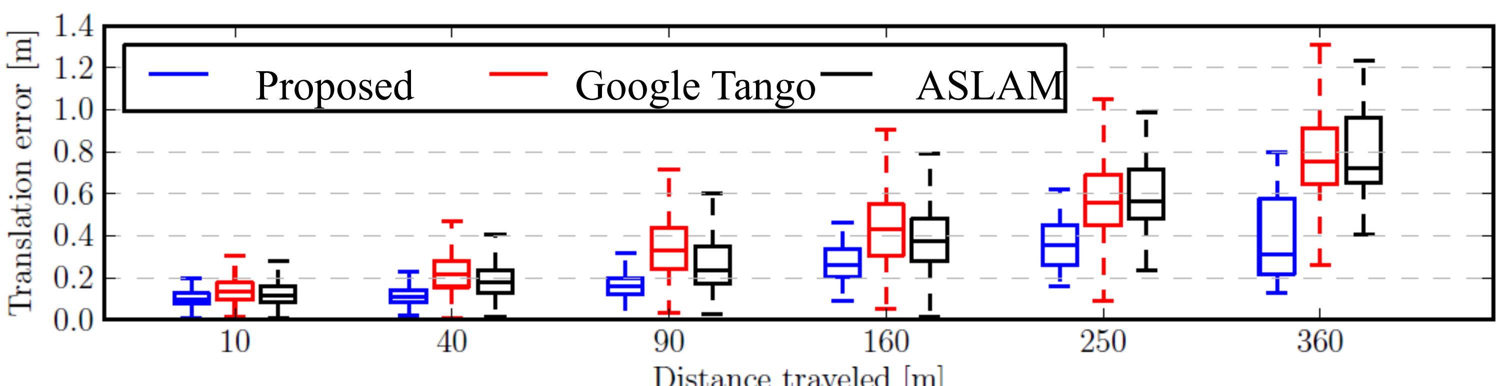
Comparison with Previous Works

https://www.youtube.com/watch?v=CsJkci5lfco

Open Source



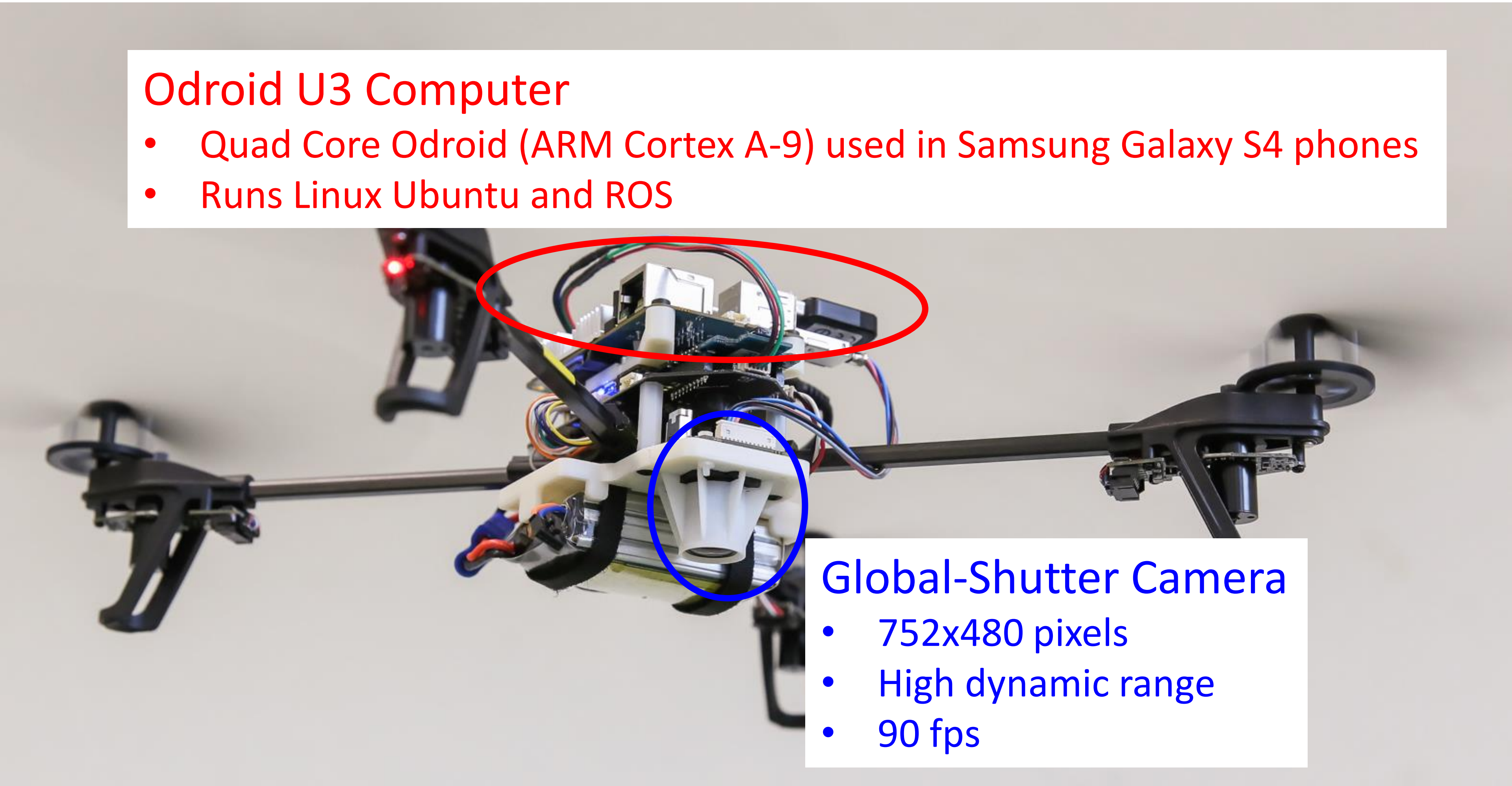




Forster, Carlone, Dellaert, Scaramuzza, IMU Preintegration on Manifold for efficient Visual-Inertial Maximum-a-Posteriori Estimation, *Robotics Science and Systens*'15

Integration on a Quadrotor Platform

Quadrotor System



Odroid U3 Computer

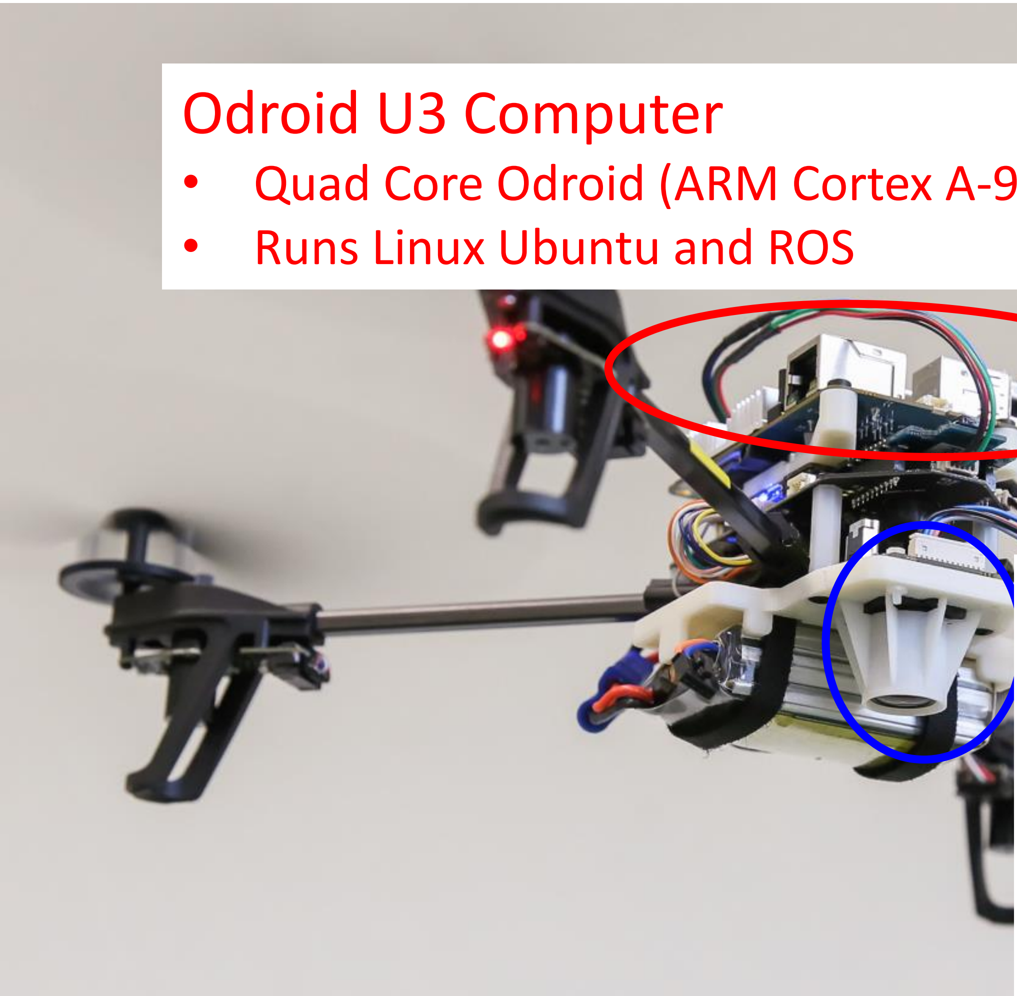
- Quad Core Odroid (ARM Cortex A-9) used in Samsung Galaxy S4 phones
- Runs Linux Ubuntu and ROS

Global-Shutter Camera

- 752x480 pixels
- High dynamic range
- 90 fps

450 grams

Quadrotor System



Odroid U3 Computer

- Quad Core Odroid (ARM Cortex A-9) used in Samsung Galaxy S4 phones
- Runs Linux Ubuntu and ROS

450 grams

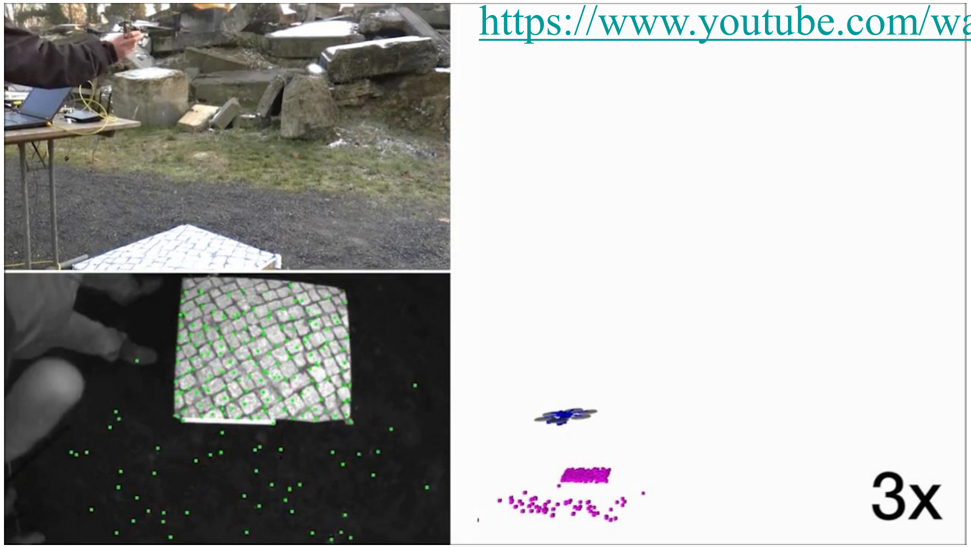
Indoors and outdoors experiments

https://www.youtube.com/watch?v=4X6Voft4Z_0

RMS error: 5 mm, height: 1.5 m – Down-looking camera

Speed: 4 m/s, height: 1.5 m – Down-looking camera

https://www.youtube.com/watch?v=3mNY9-DSUDk






Faessler, Fontana, Forster, Mueggler, Pizzoli, Scaramuzza, Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle, **Journal of Field Robotics, 2015**.

Visual-Inertial Odometry - Results

https://www.youtube.com/watch?v=6KXBoprGaR0

Open Source

Forster, Carlone, Dellaert, Scaramuzza, IMU Preintegration on Manifold for efficient Visual-Inertial Maximum-a-Posteriori Estimation, *Robotics Science and Systens*'15, **Best Paper Award Finalist**

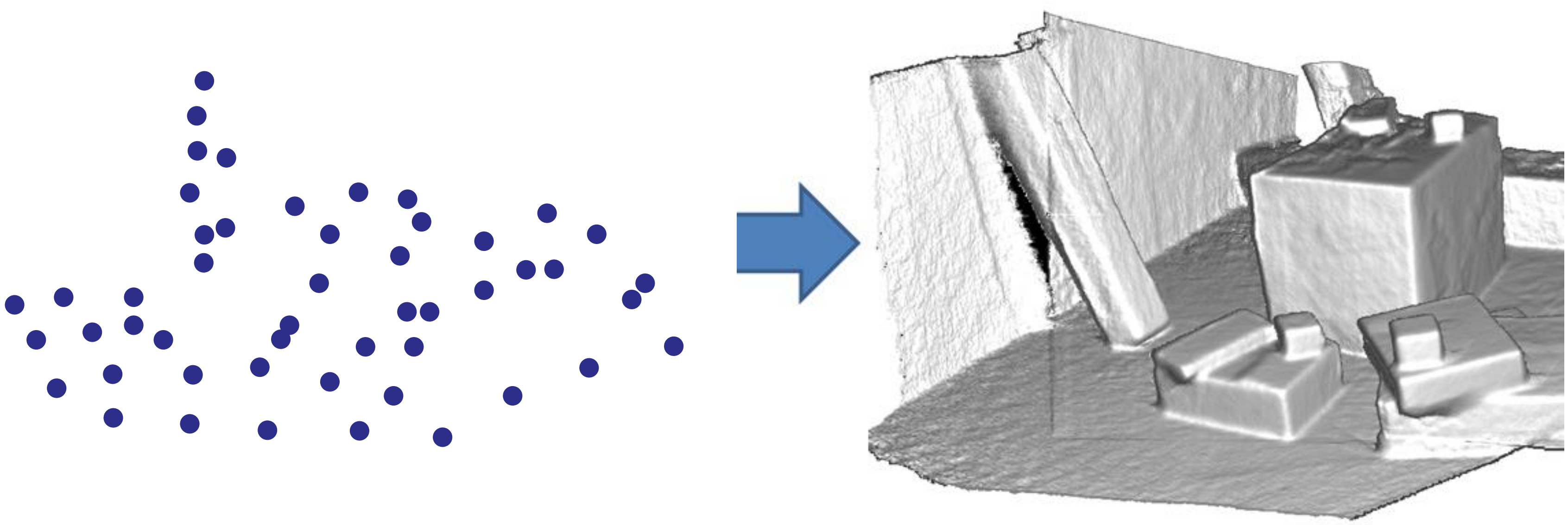
Application: Autonomous Inspection of Bridges and Power Masts

Project with Parrot: Autonomous vision-based navigation
https://www.youtube.com/watch?v=BxVoaLrKJ4U

Parrot senseFly

Albris drone

From Sparse to Dense 3D Models

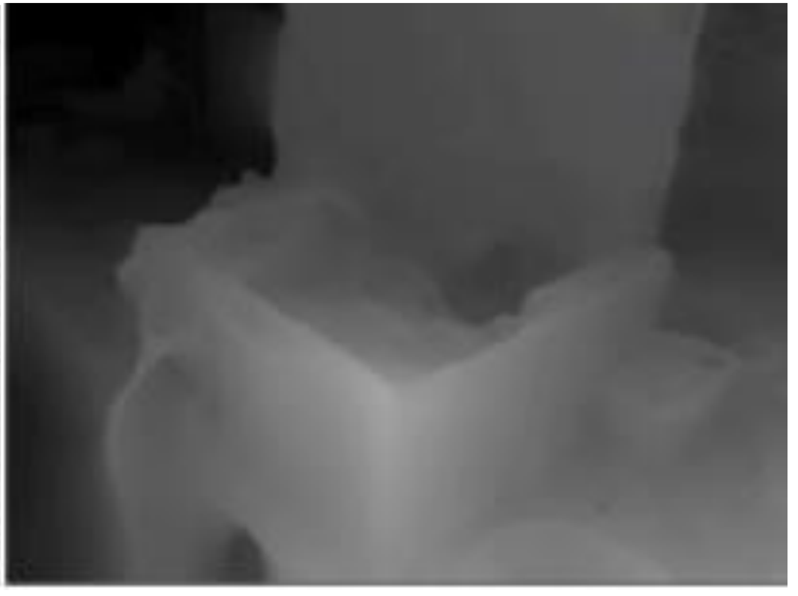


[M. Pizzoli, C. Forster, D. Scaramuzza, REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time, ICRA'14]

Dense Reconstruction Pipeline

- **Local methods**
  - Estimate depth for every pixel independently using **photometric cost aggregation**
- **Global methods**
  - Refine the depth surface as a whole by enforcing smoothness constraint ("*Regularization*")

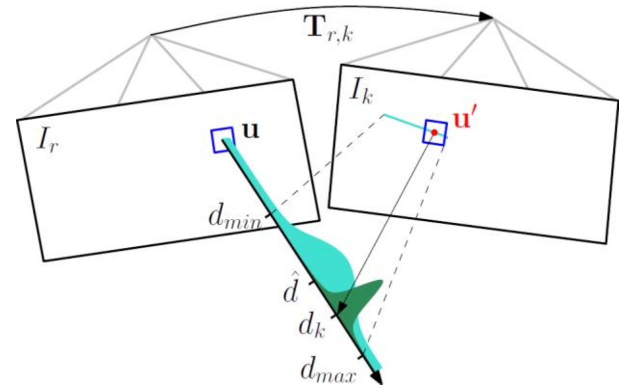
$$E(d) = \underbrace{E_d(d)}_{\text{Data term}} + \underbrace{\lambda E_s(d)}_{\text{Regularization term: penalizes } \textit{non-smooth} \text{ surfaces}}$$



[Newcombe et al. 2011]

REMODE: Probabilistic Monocular Dense Reconstruction [ICRA'14]

- **Track independently** every pixel using the same recursive **Bayesian depth estimation of SVO**



$$p(\tilde{d}_i^k | d_i, \rho_i) = \rho_i \mathcal{N}\left(\tilde{d}_i^k | d_i, \tau_i^2\right) + (1 - \rho_i)\mathcal{U}\left(\tilde{d}_i^k | d_i^{\min}, d_i^{\max}\right)$$

- A ***regularized depth map*** F(u) is computed from the noisy depth map D(u) as

$$\min_F \int_\Omega \{G(\mathbf{u}) \|\nabla F(\mathbf{u})\|_\epsilon + \lambda \|F(\mathbf{u}) - D(\mathbf{u})\|_1\} \, d\mathbf{u}$$

where $$G(\mathbf{u}) = \mathbb{E}_\rho[q](\mathbf{u}) \frac{\sigma^2(\mathbf{u})}{\sigma_{max}^2} + \{1 - \mathbb{E}_\rho[q](\mathbf{u})\}$$

Minimization is done using [Chambolle & Pock, 2011]

[M. Pizzoli, C. Forster, D. Scaramuzza, REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time, ICRA'14]

REMODE: Probabilistic Monocular Dense Reconstruction [ICRA'14]
https://www.youtube.com/watch?v=QTKd5UWCG0Q
Running at 50 Hz on GPU on a Lenovo W530, i7

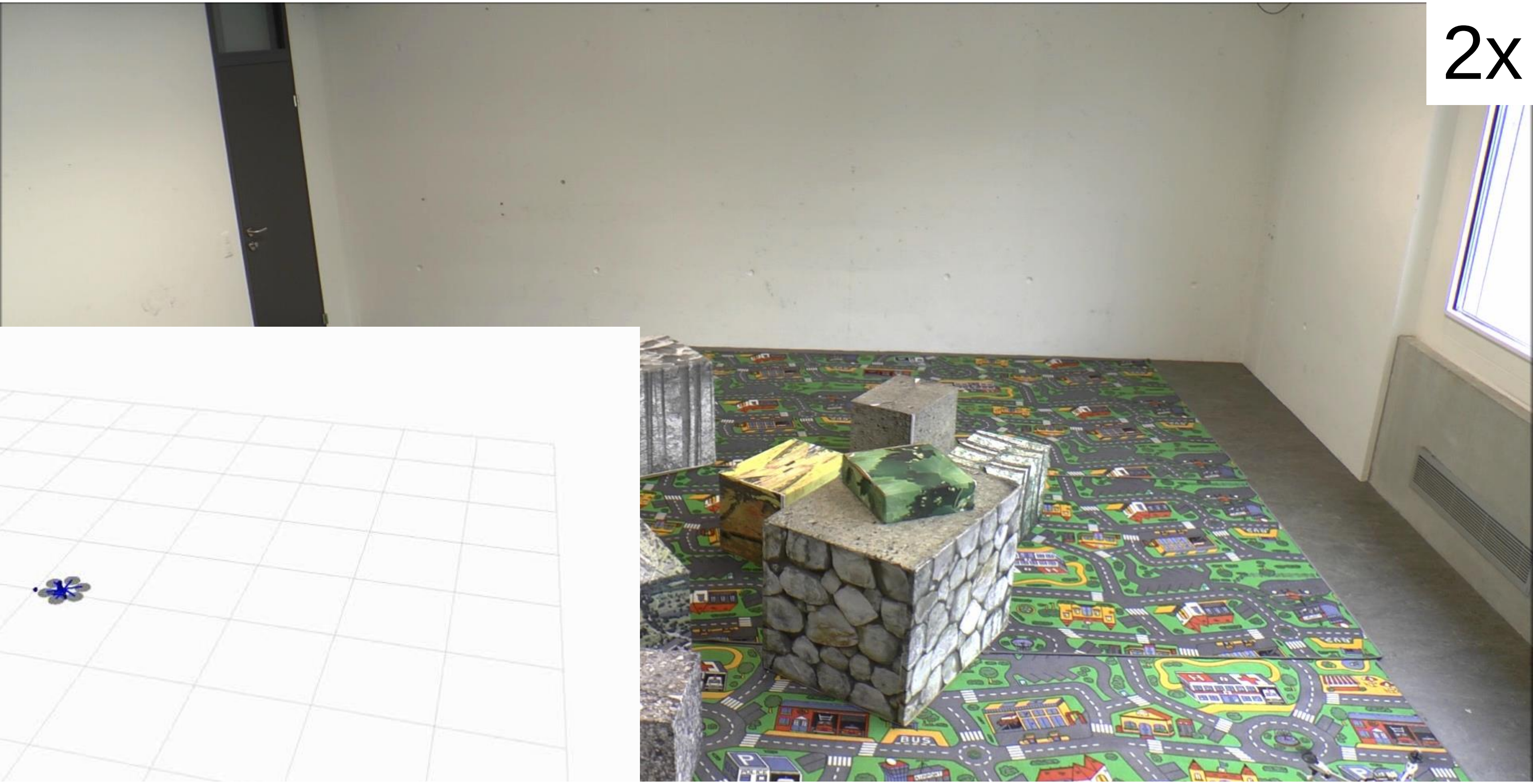
Open Source

[M. Pizzoli, C. Forster, D. Scaramuzza, REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time, ICRA'14]

Autonomus, Flying 3D Scanning

https://www.youtube.com/watch?v=7-kPiWaFYAc

- Sensing, control, state estimation run onboard at 50 Hz (Odroid U3, ARM Cortex A9)
- Dense reconstruction runs live on video streamed to laptop (Lenovo W530, i7)

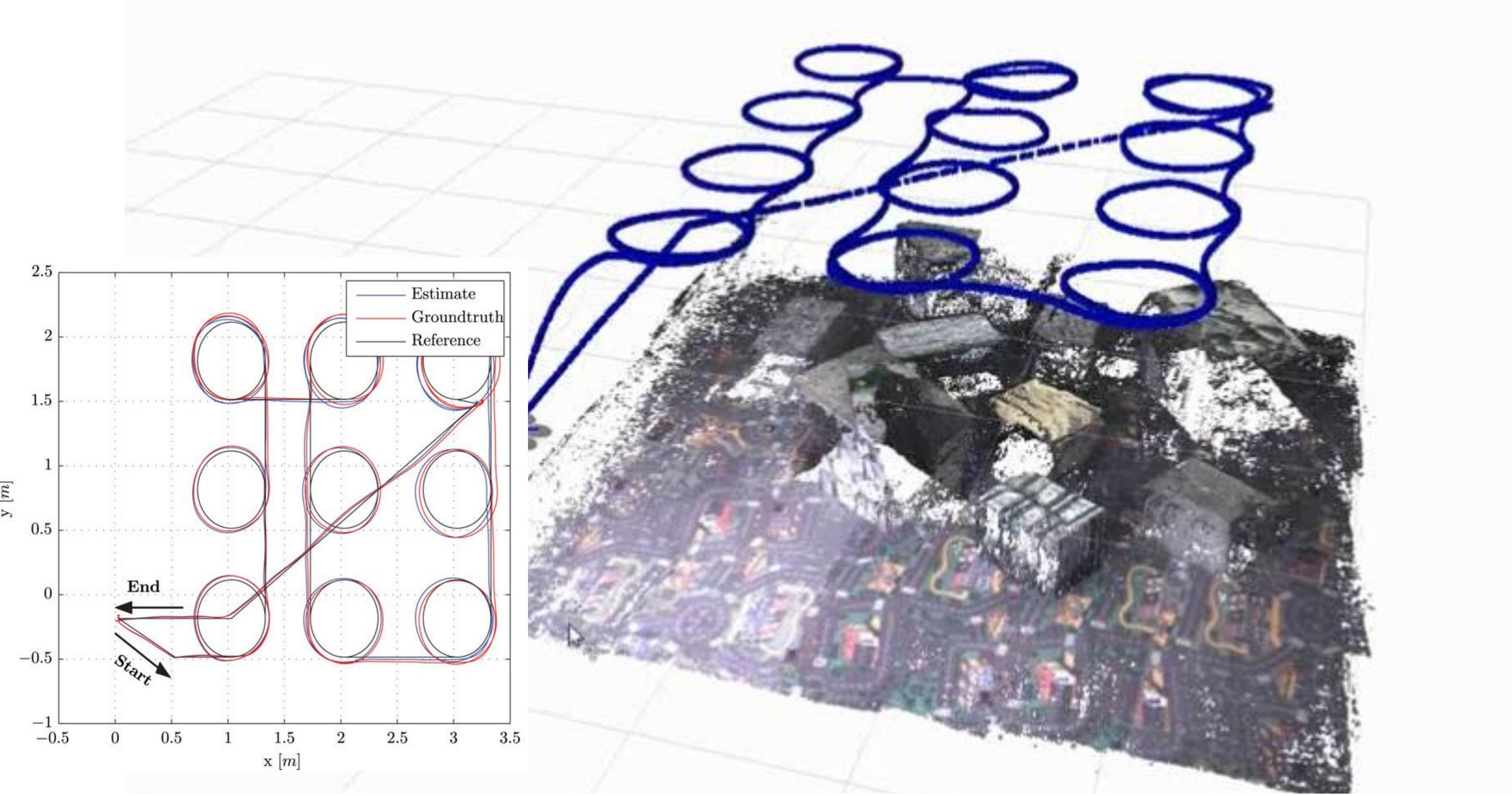


Faessler, Fontana, Forster, Mueggler, Pizzoli, Scaramuzza, Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle, **Journal of Field Robotics, 2015**.

Autonomus, Flying 3D Scanning [JFR'15]

https://www.youtube.com/watch?v=7-kPiWaFYAc

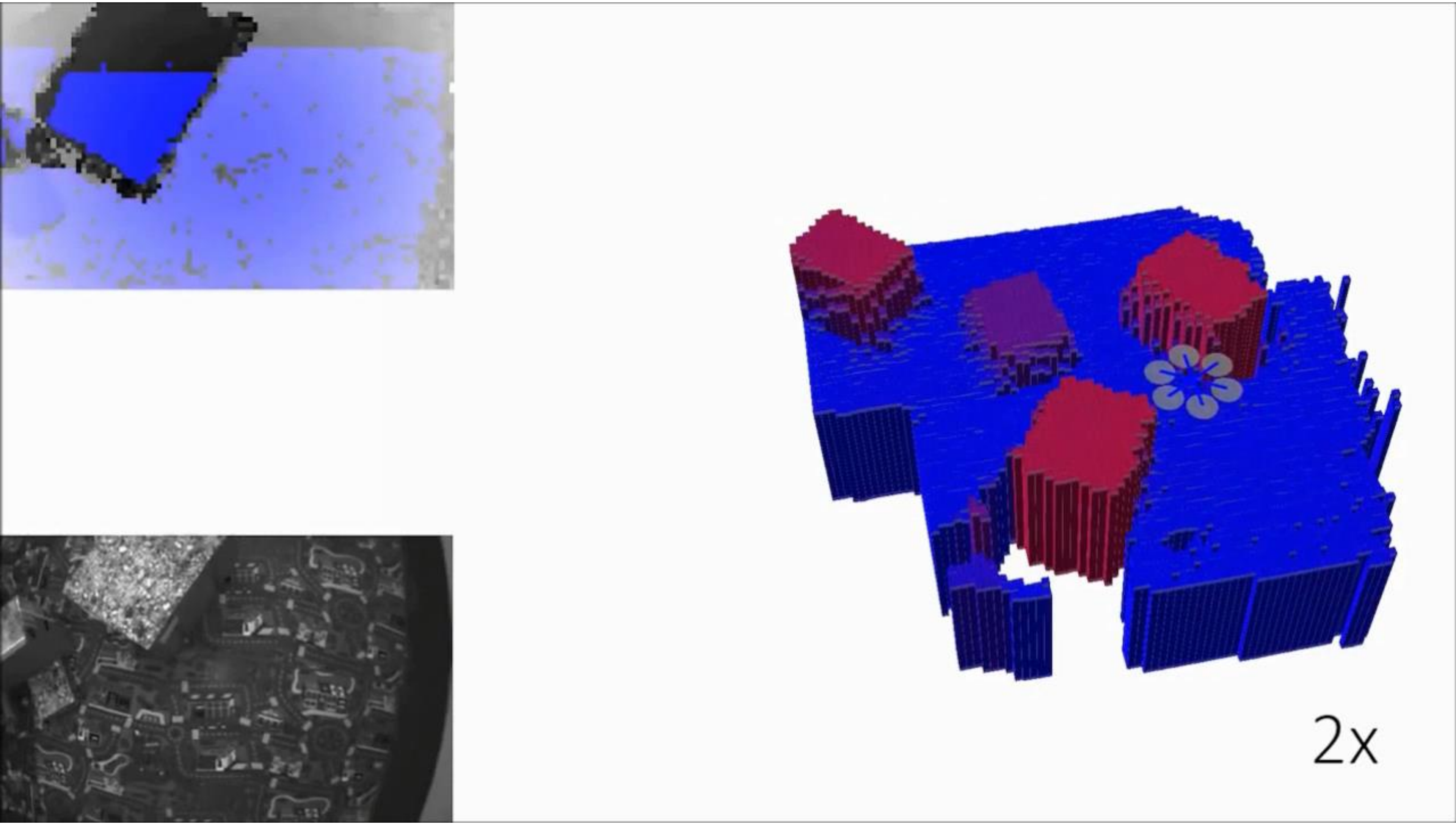
- Sensing, control, state estimation run onboard at 50 Hz (Odroid U3, ARM Cortex A9)
- Dense reconstruction runs live on video streamed to laptop (Lenovo W530, i7)



Faessler, Fontana, Forster, Mueggler, Pizzoli, Scaramuzza, Autonomous, Vision-based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle, **Journal of Field Robotics, 2015**.

Autonomous Landing-Spot Detection and Landing

https://www.youtube.com/watch?v=phaBKFwfcJ4



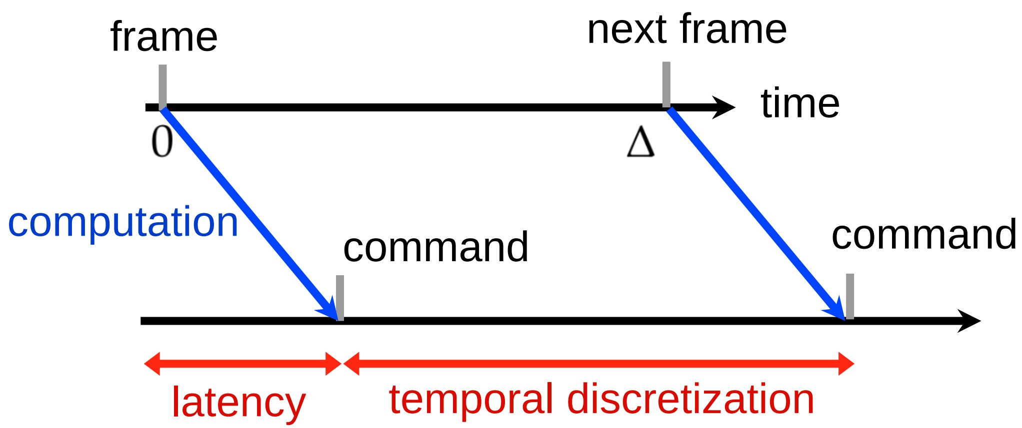
Forster, Faessler, Fontana, Werlberger, Scaramuzza, Continuous *On-Board Monocular-Vision-based Elevation Mapping Applied to Autonomous Landing of Micro Aerial Vehicles*, ICRA'15.

Outlook

To go faster, we need faster sensors!

- At the current state, the agility of a robot is limited by the latency and temporal discretization of its sensing pipeline.

- Currently, the average robot-vision algorithms have latencies of 50-200 ms. This puts a hard bound on the agility of the platform.

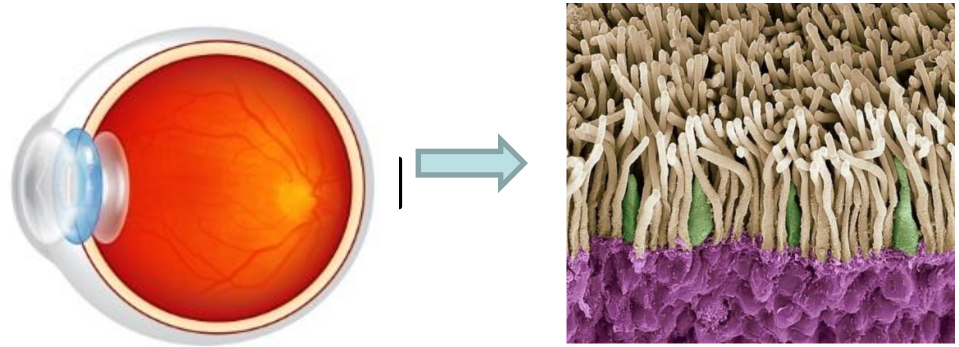


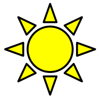
- **Can we create a low-latency, low-discretization perception pipeline?**
  - Yes, if we use **event-based cameras**

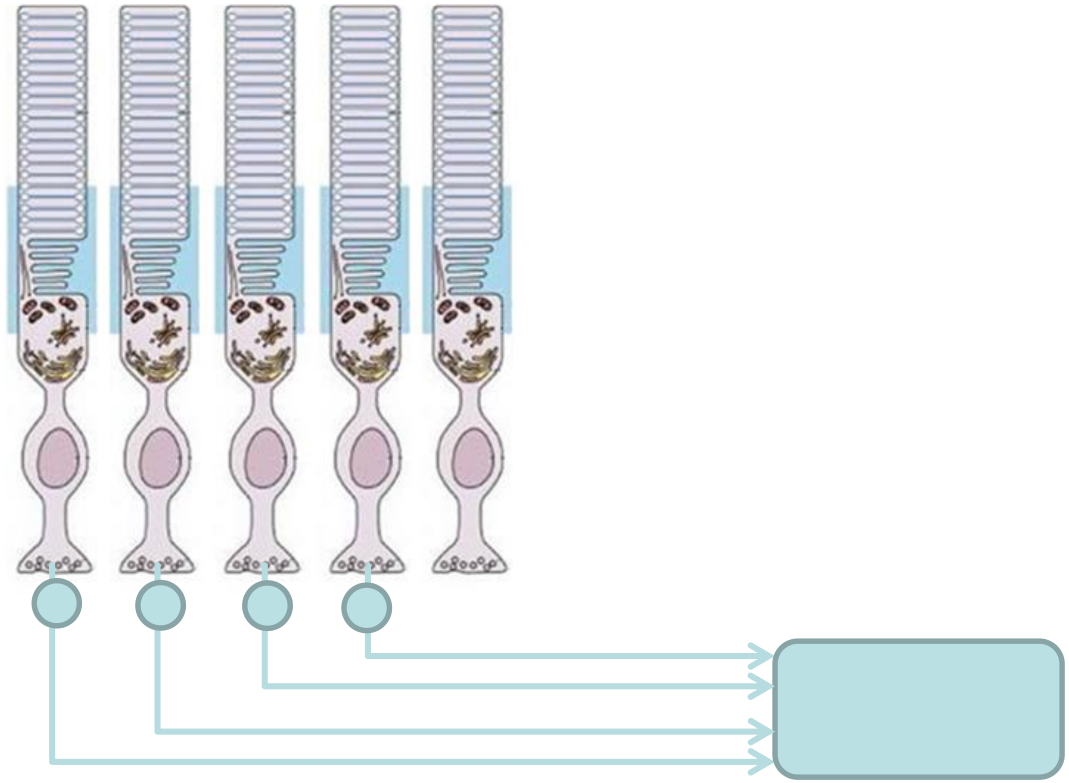
[Censi & Scaramuzza, «Low Latency, Event-based Visual Odometry», ICRA'14]

Human Vision System

- 130 million **photoreceptors**
- But only 2 million **axons**!







Dynamic Vision Sensor (DVS)

- **Event-based camera** developed by Tobi Delbruck's group (ETH & UZH).
- Temporal resolution: **1 μs**
- High dynamic range: **120 dB**
- Low power: **20 mW**
- Cost: 2,500 EUR

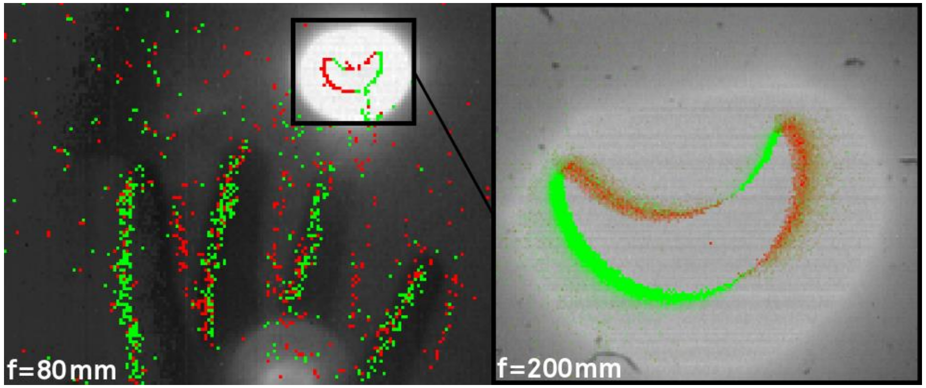
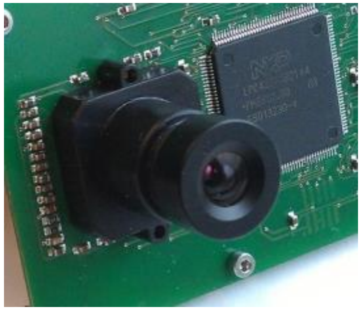
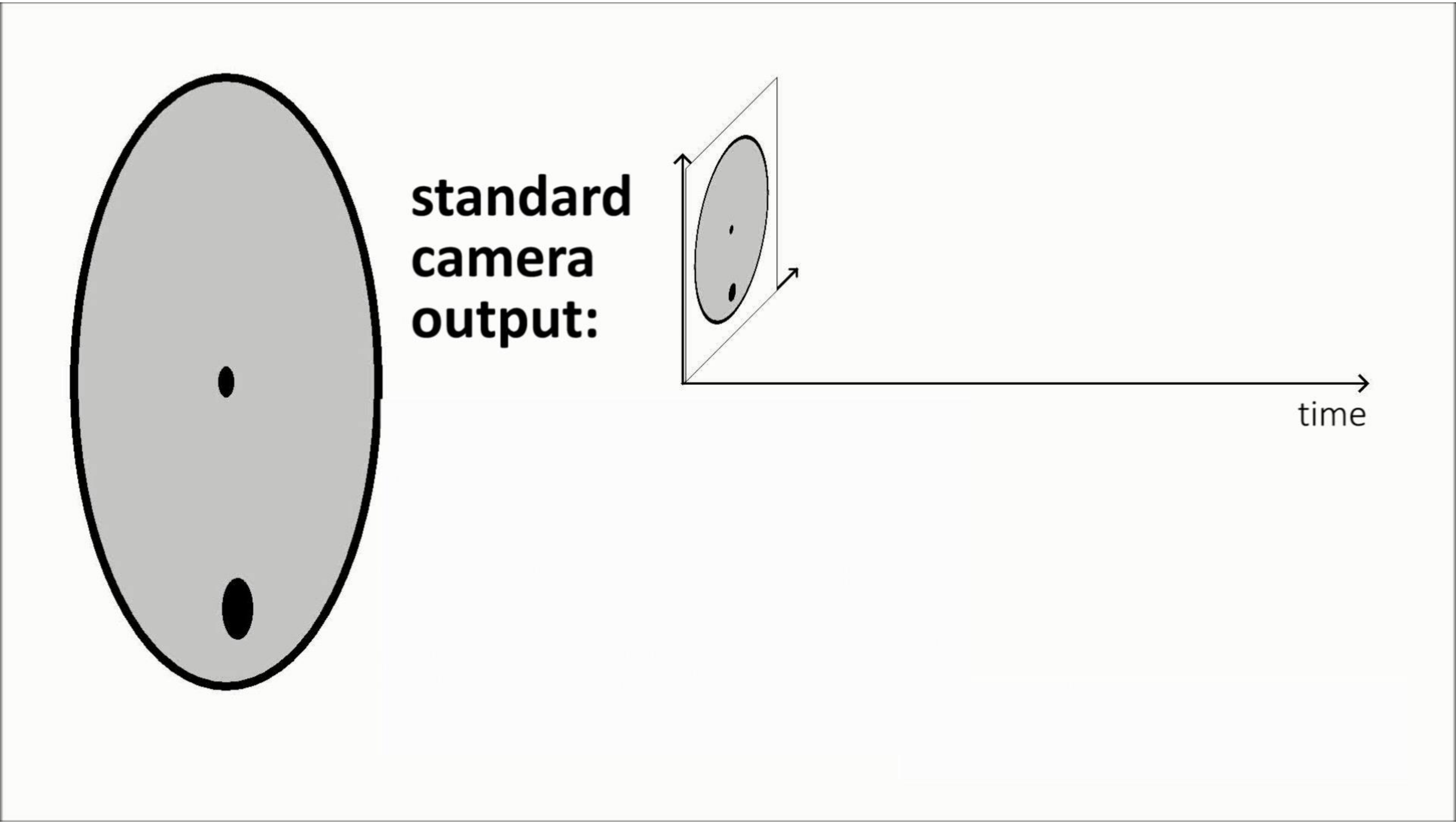


Image of the solar eclipse (March'15) captured by a DVS (courtesy of IniLabs)



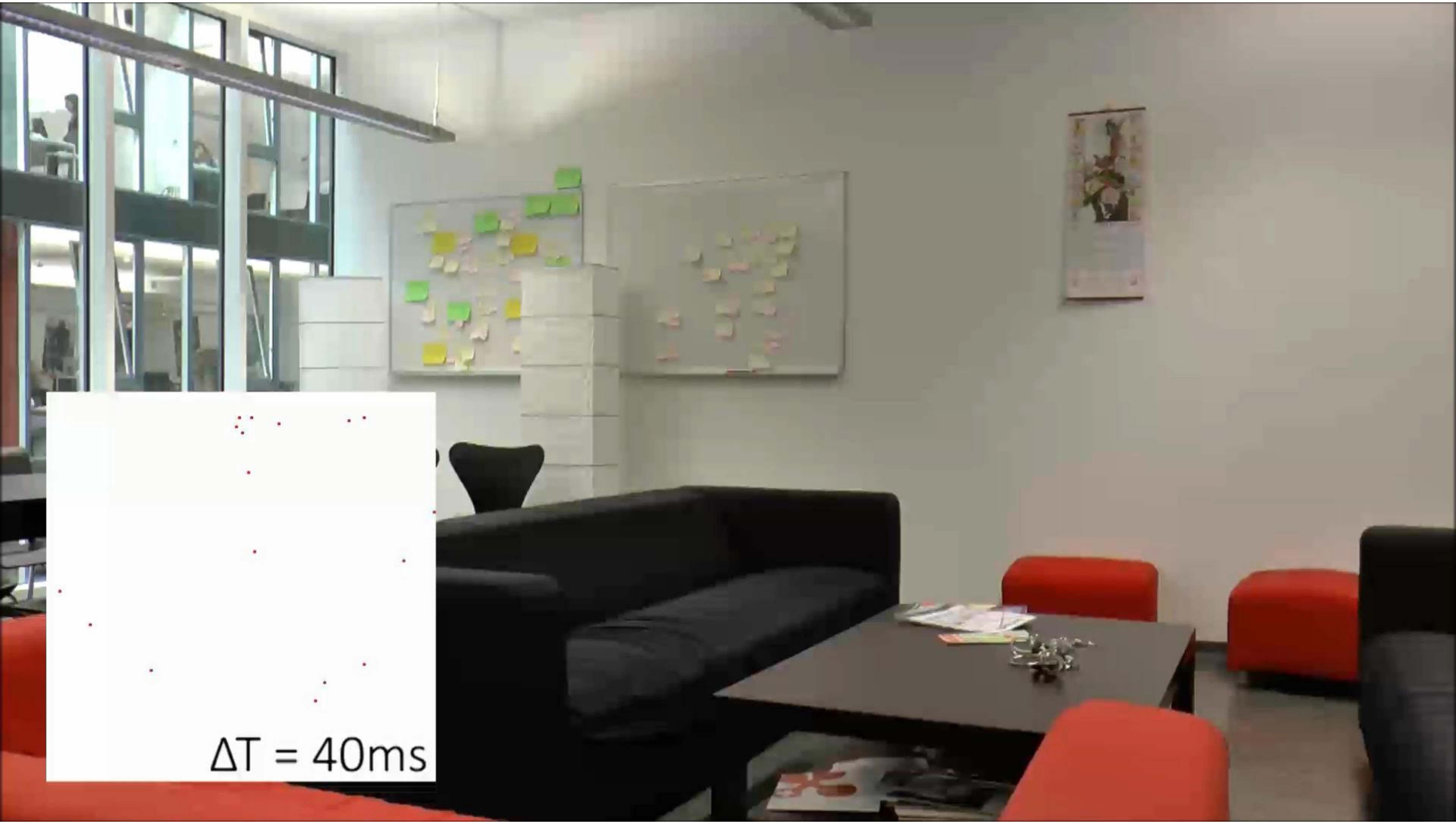
[Lichtsteiner, Posch, Delbruck. A 128x128 120 dB 15μs Latency Asynchronous Temporal Contrast Vision Sensor. 2008]

Camera vs Dynamic Vision Sensor



https://www.youtube.com/watch?v=LauQ6LWTkxM

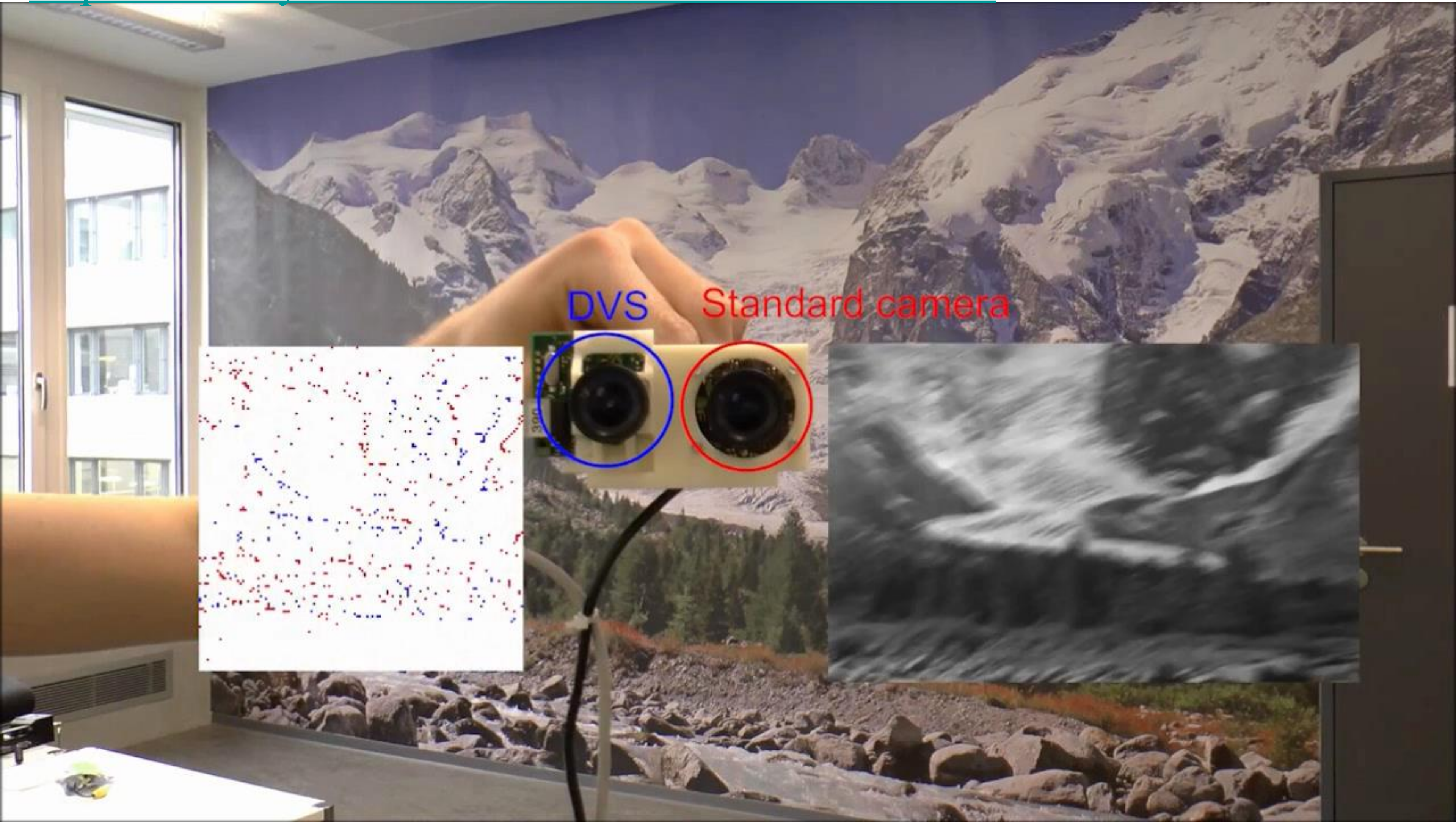
Camera vs Dynamic Vision Sensor



https://www.youtube.com/watch?v=LauQ6LWTkxM

High-speed State Estimation

https://www.youtube.com/watch?v=iZZ77F-hwzs



[Event-based Camera Pose Tracking using a Generative Event Model, Under review. Available at arXiv]

[Censi & Scaramuzza, *Low Latency, Event-based Visual Odometry*, ICRA'14]

Conclusions

- Visual-inertial state estimation is crucial for GPS-denied navigation.
- More accurate than GPS, DGPS, and RTK-GPS.
- Pending challenges: robustness to changing illumination and high-speed motion.
- Event cameras are revolutionary visual sensors that can address such challenges where standard cameras fail.