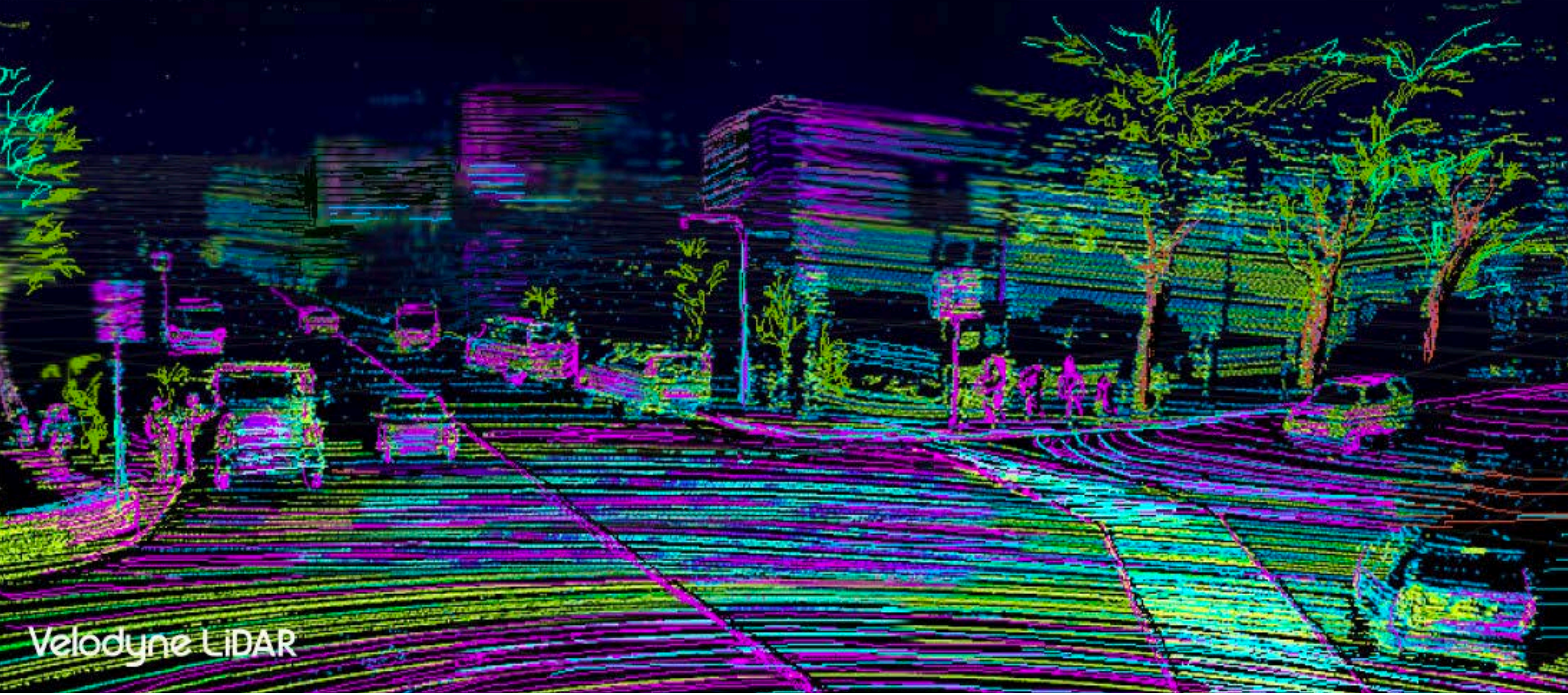
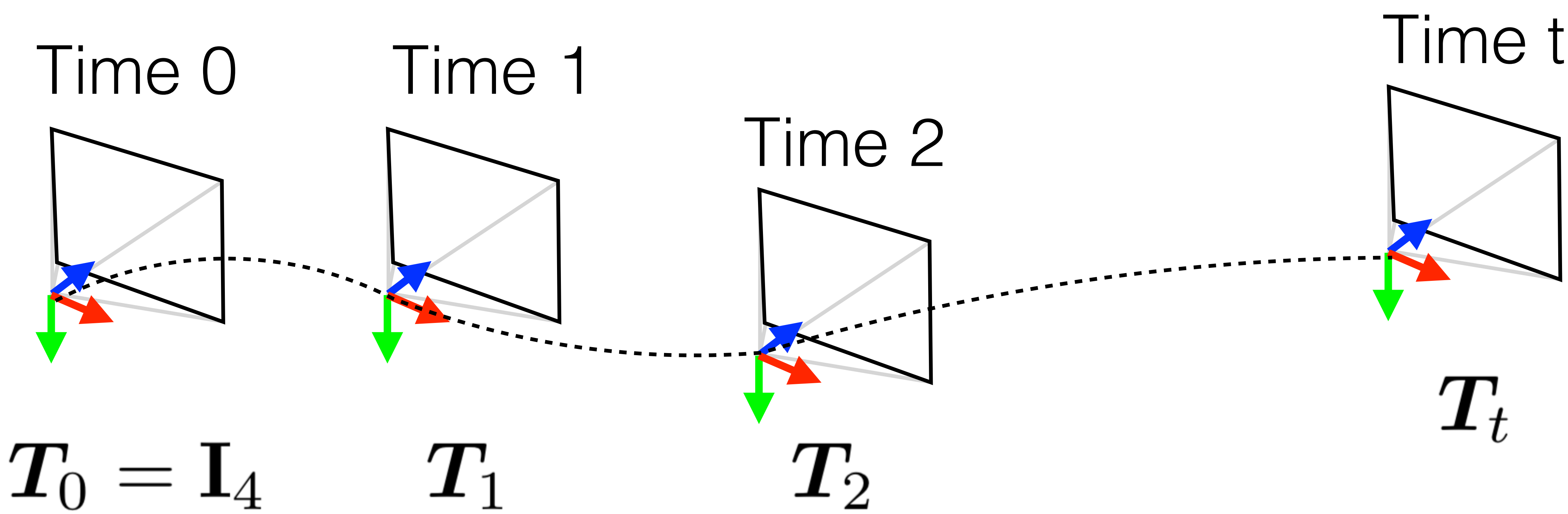


16.485: VNAV - Visual Navigation for Autonomous Vehicles

Luca Carlone

Lecture 26: Advanced Topics - Beyond Cameras

Previously on VNAV: 2-view Geometry and VO



Visual odometry (VO): motion estimation estimation based on cameras (monocular, stereo, RGB-D, …)

others: wheel odometry, inertial, visual-inertial

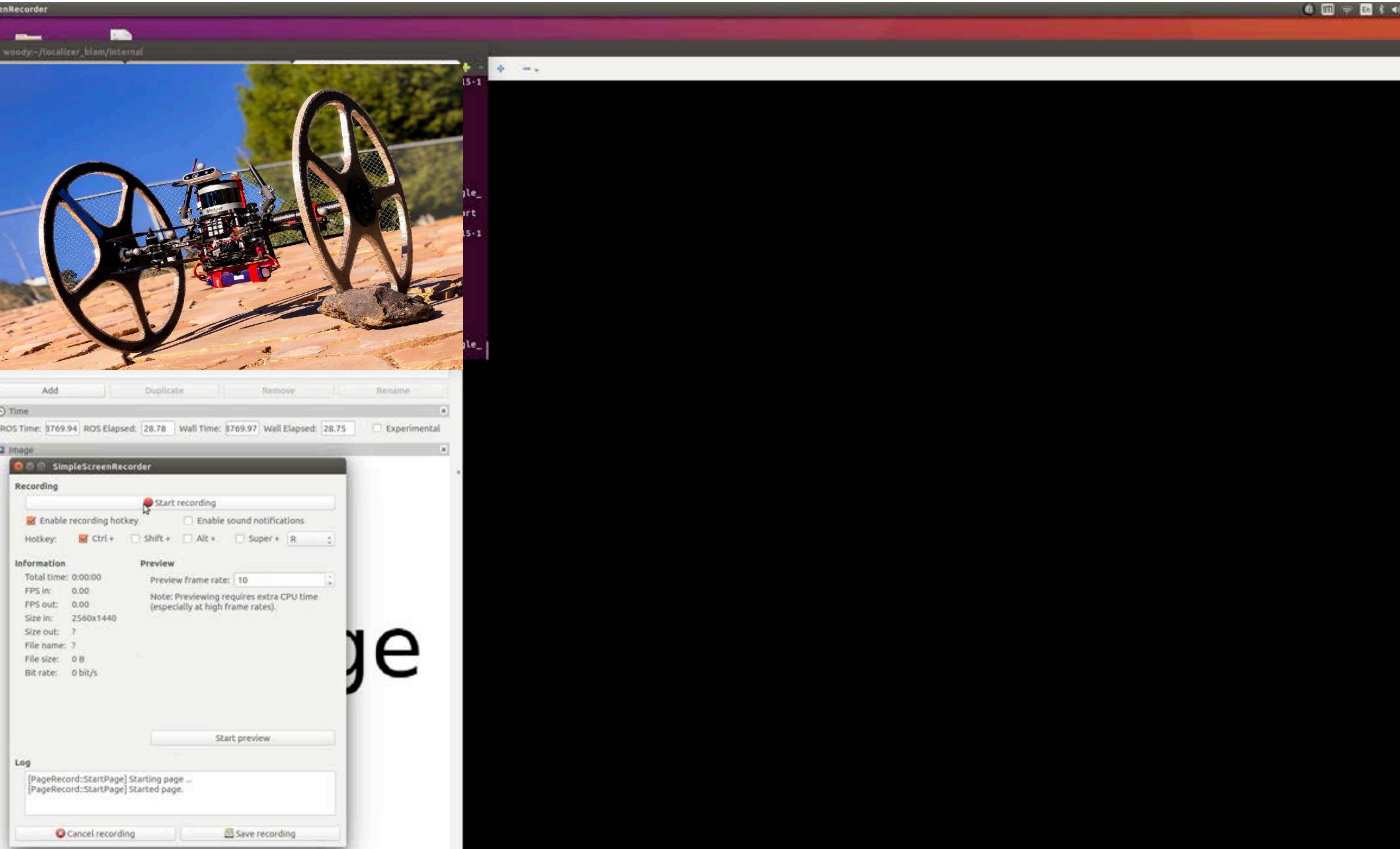
Today: Beyond Cameras

- wheel odometry
- GPS
- Lidar
- Inertial Measurement Unit (IMU)
- Event Cameras



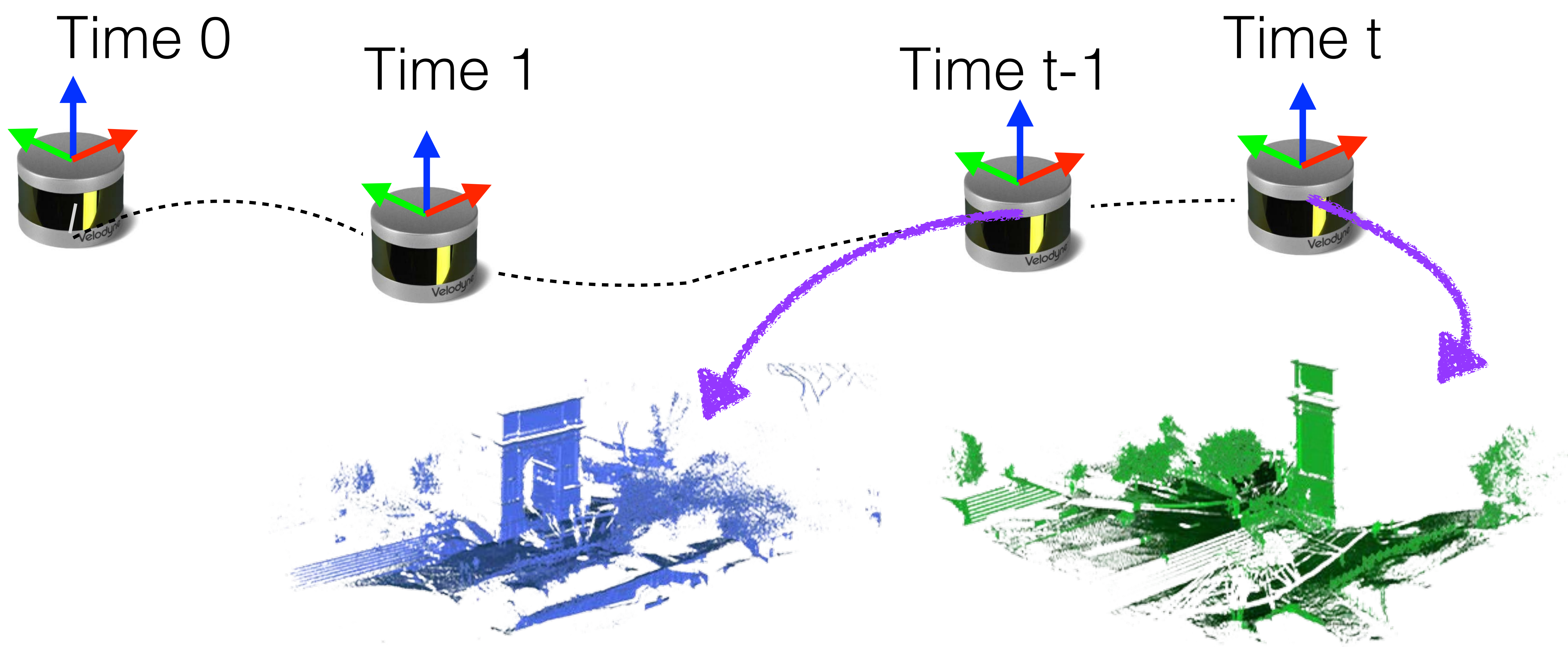
830g	160g	4g	3g
8 W	2.5 W	0.3W	~1 W

Lidar Odometry & Lidar SLAM

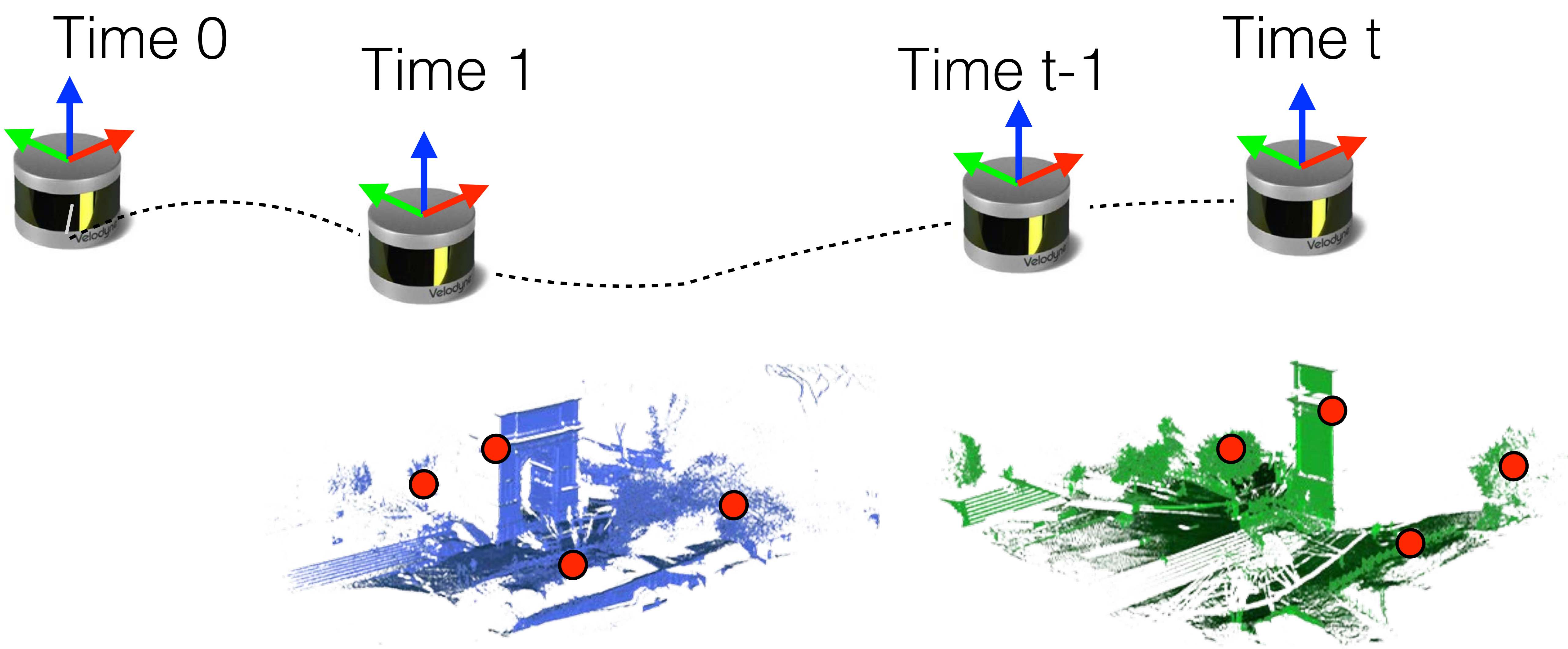


DARPA Subterranean Challenge, in collaboration with JPL

Feature-based Lidar Odometry

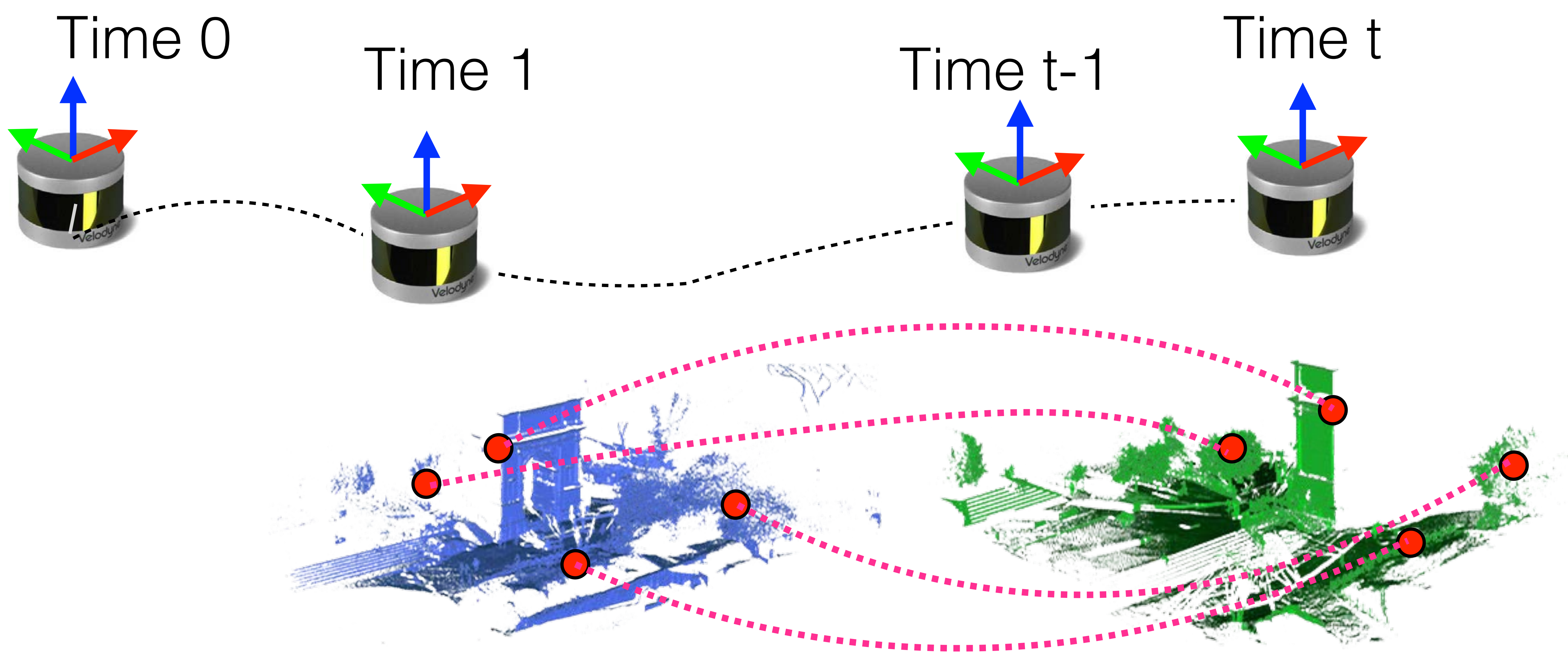


Feature-based Lidar Odometry



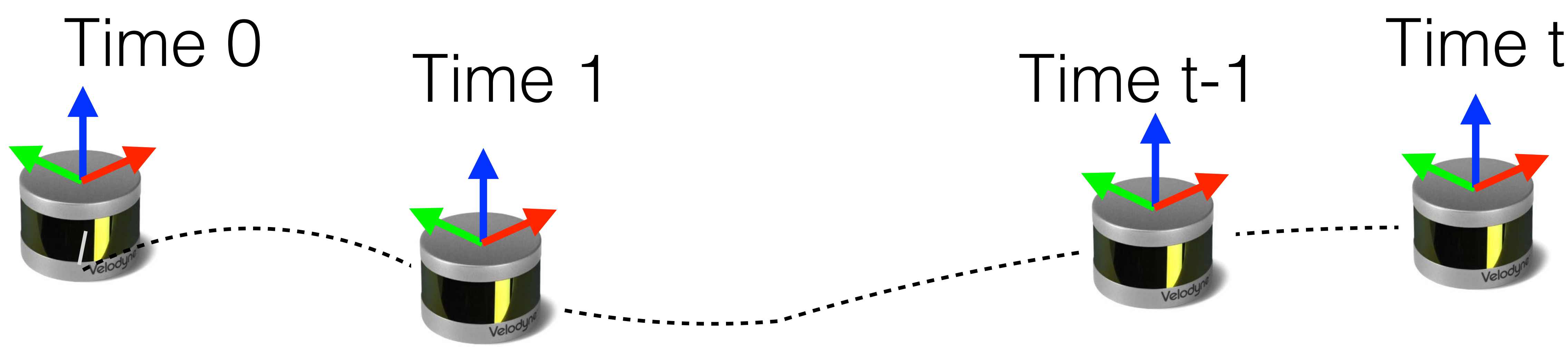
Registration: compute relative pose between scans:
- extract features & descriptors
- use descriptors for matching
- compute relative pose

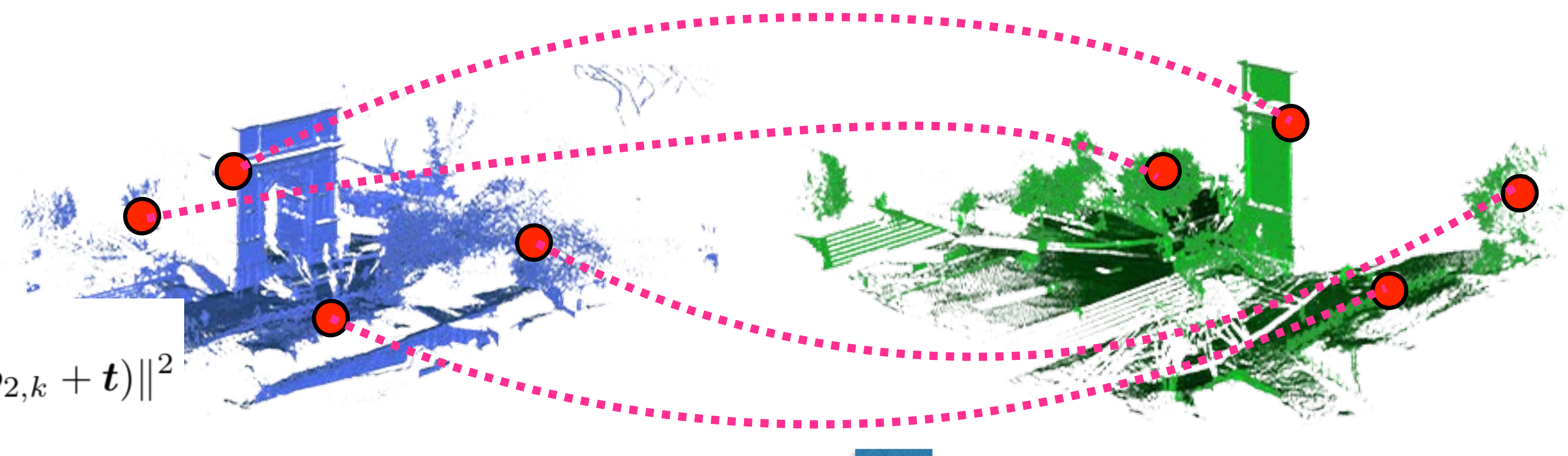
Feature-based Lidar Odometry



Registration: compute relative pose between scans:
- extract features & descriptors
- use descriptors for matching
- compute relative pose

Feature-based Lidar Odometry

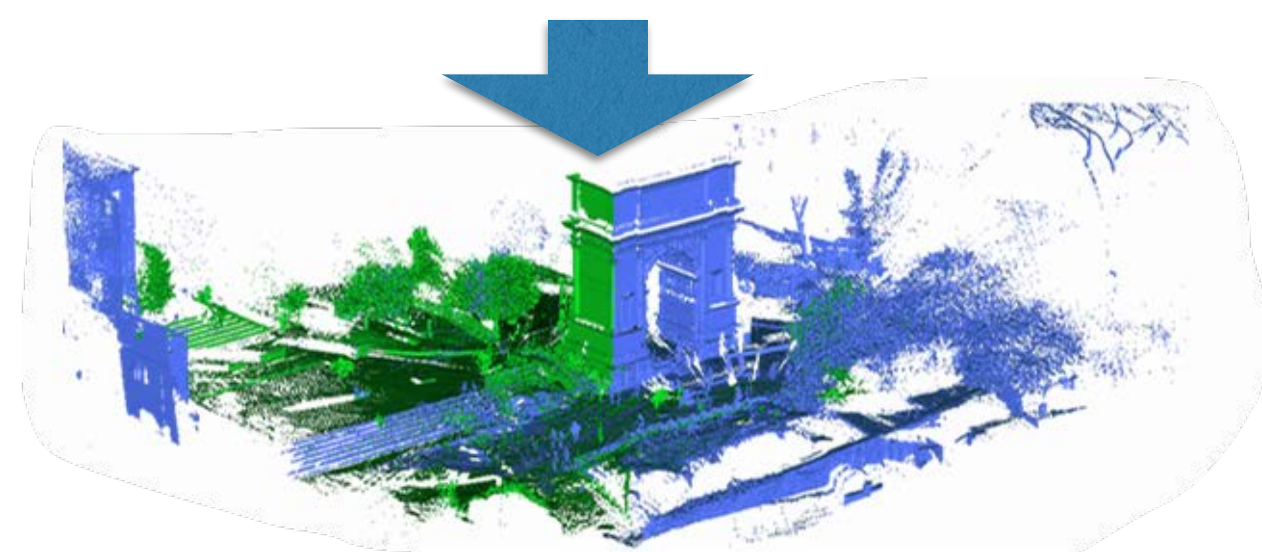




$$\min_{\substack{\boldsymbol{R}\in \mathrm{SO}(3)\\ \boldsymbol{t}\in\mathbb{R}^3}} \sum_{k=1}^{N} \|\boldsymbol{p}_{1,k} - (\boldsymbol{R}\boldsymbol{p}_{2,k} + \boldsymbol{t})\|^2$$

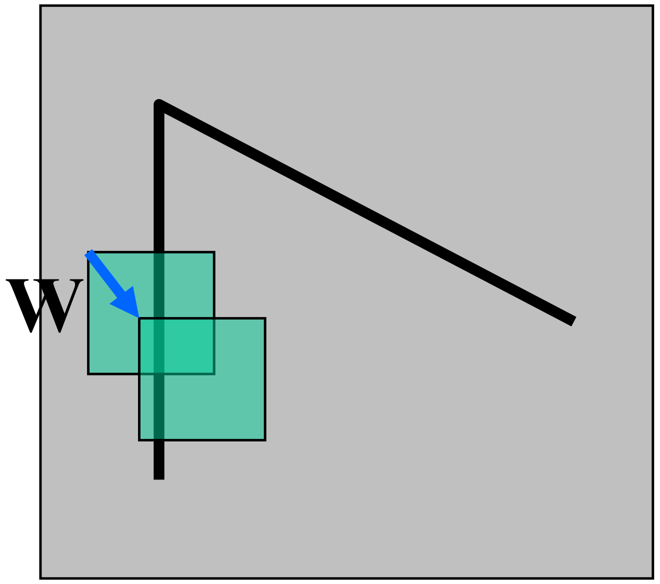
Registration: compute relative pose between scans:

- extract features & descriptors
- use descriptors for matching
- compute relative pose



Feature Detection: 3D Harris Corners

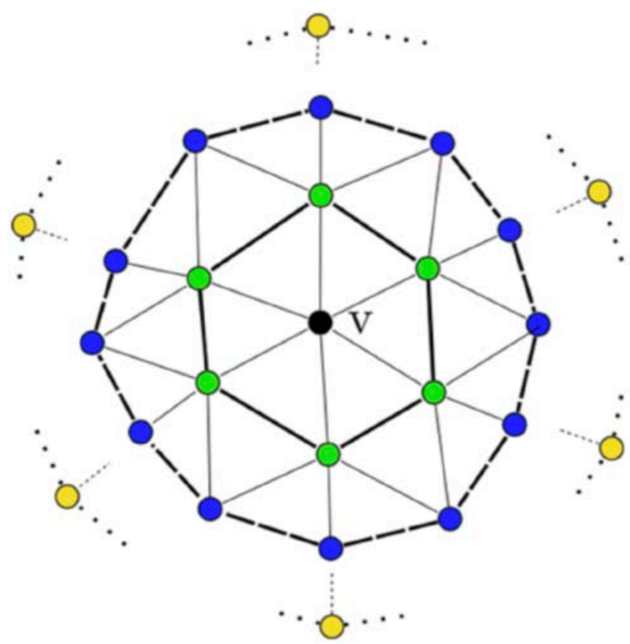
2D Harris Corner Detector



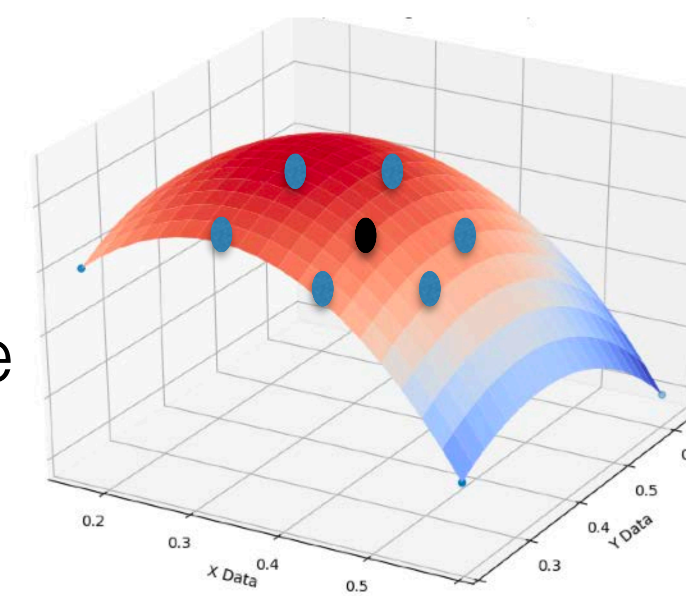
$$\boldsymbol{G} = \sum_{\boldsymbol{x} \in W(\bar{\boldsymbol{x}})} \nabla \mathcal{I}(\boldsymbol{x}) \nabla \mathcal{I}(\boldsymbol{x})^{\mathsf{T}}$$

$$C(\boldsymbol{G}) = \det(\boldsymbol{G}) - k \operatorname{tr}(\boldsymbol{G})^2$$

3D Harris Corner Detector



- Consider neighborhood of v
- Fit paraboloid to sets of points
- Evaluate gradients (+some magic)
- Apply Harris corner-ness score



Vis Comput
DOI 10.1007/s00371-011-0610-y

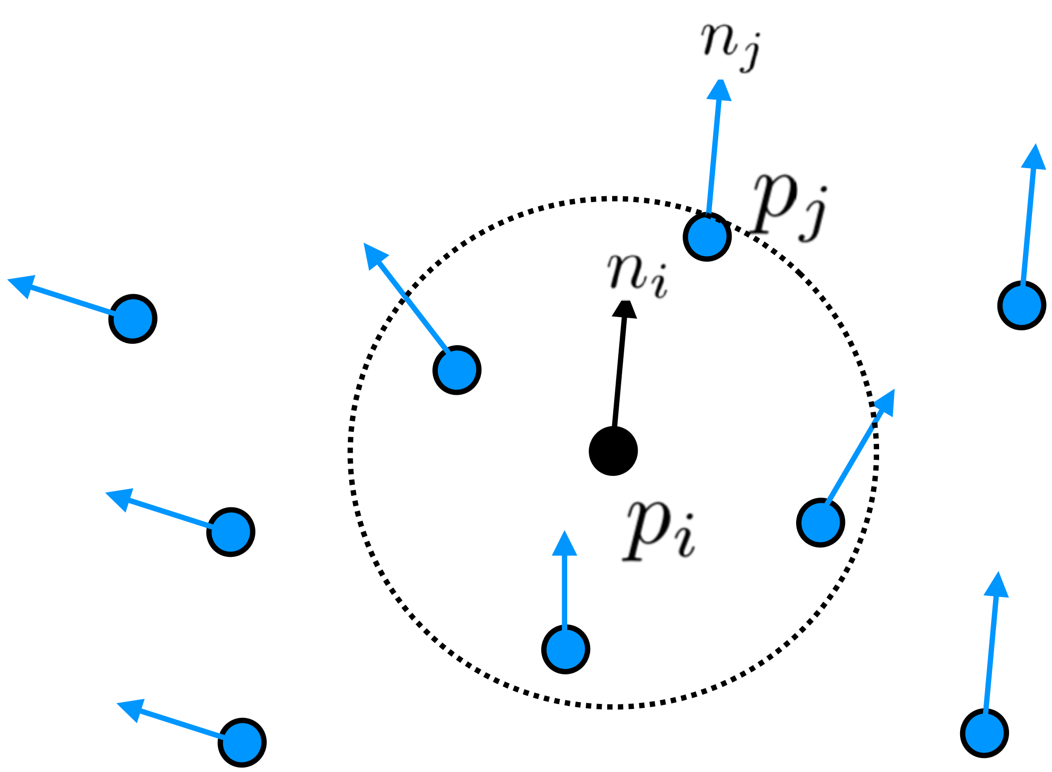
ORIGINAL ARTICLE

Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes

Others: SIFT3D, SUSAN, ISS3D

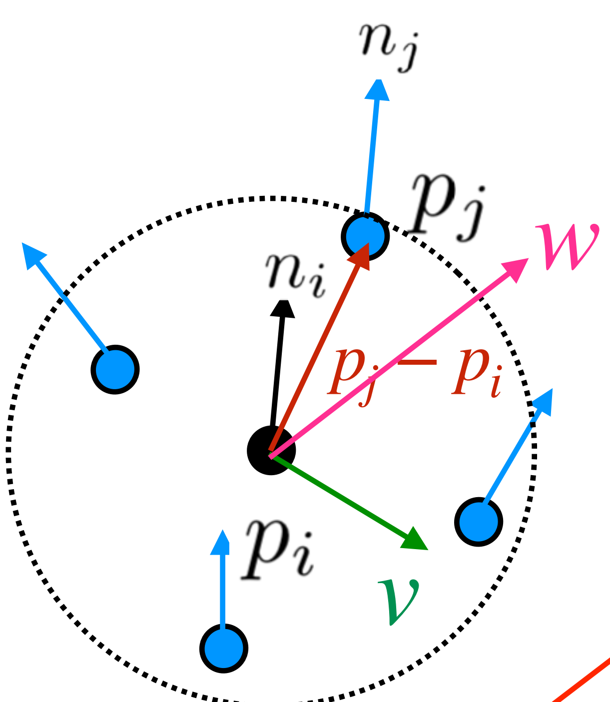
Feature Descriptors: Point Feature Histograms (PFH)

Idea: describe neighborhood of a point in the point cloud as a multi-dimensional histogram



Feature Descriptors: Point Feature Histograms (PFH)

Idea: describe neighborhood of a point in the point cloud as a multi-dimensional histogram



Define:

$$u = n_i, \;\; v = (p_j - p_i) \times u, \;\; w = u \times v$$

For points (i,j), compute:

$$\alpha = v \cdot n_j$$

$$\phi = (u \cdot (p_j - p_i))/||p_j - p_i||$$

$$\theta = \arctan(w \cdot n_j, u \cdot n_j)$$

~ Angle between normals

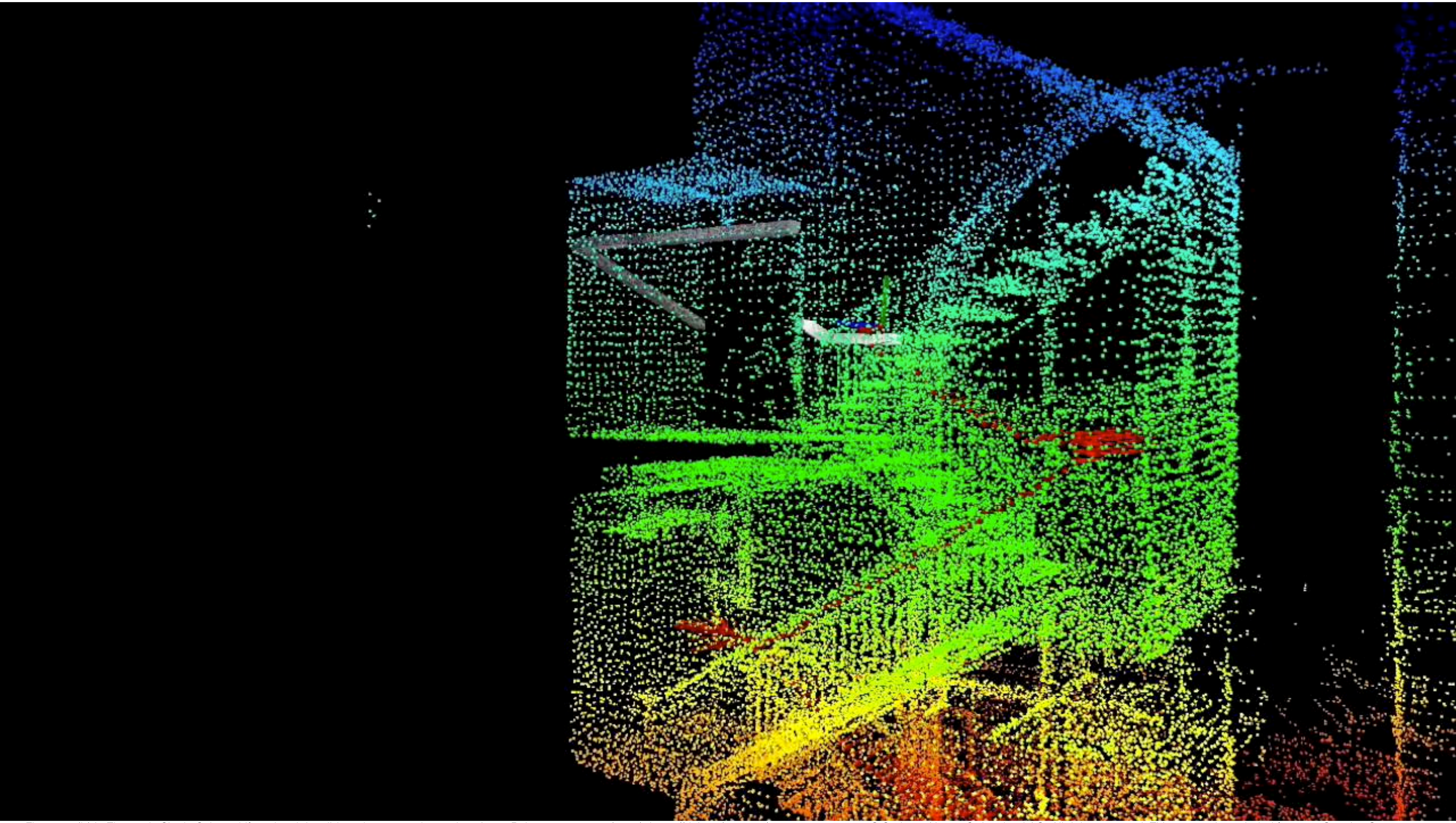
~ Angle between normal "i" and vector between points

"Local" direction of normal "j"

Bin results into a 3D histogram

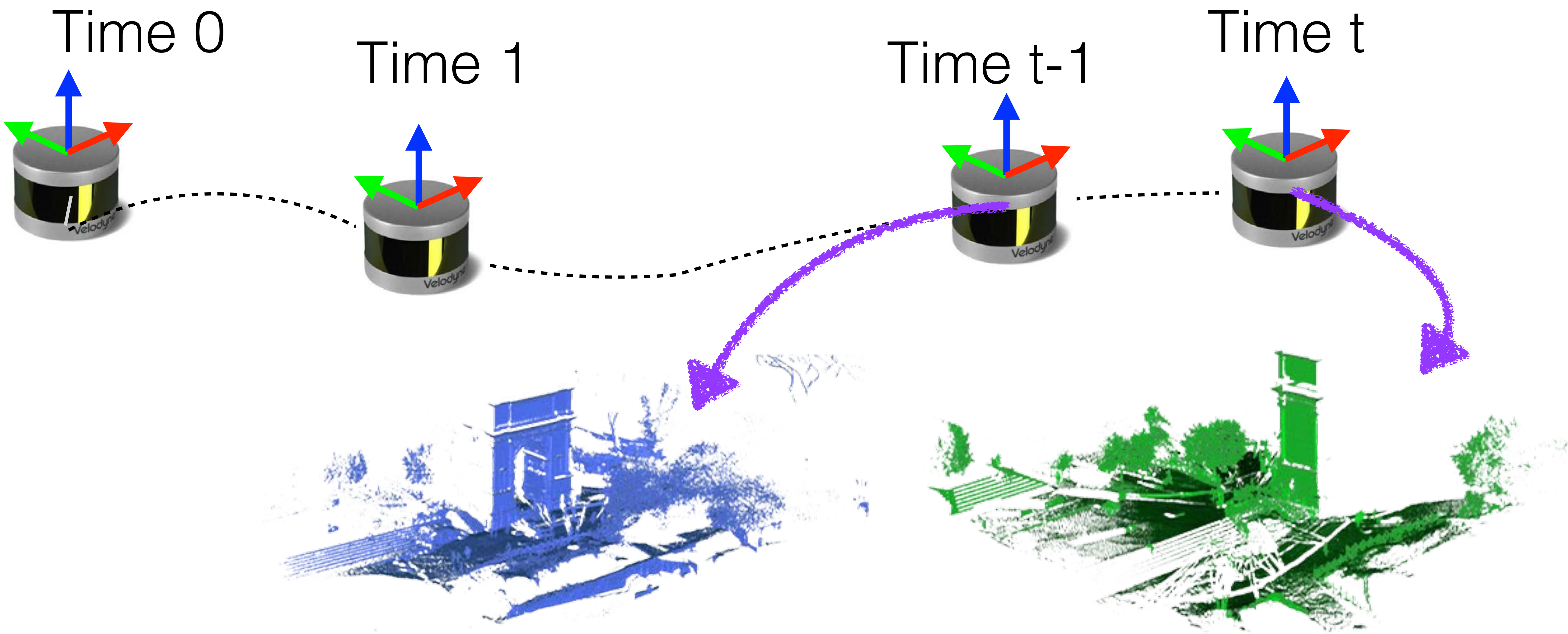
(Others: FPFH, learning-based,…)

Feature-based Lidar Odometry



[Zhang and Singh: LOAM: Lidar Odometry and Mapping in Real-time, 2014]

Dense Lidar Odometry



Iterative Closest Point (ICP)

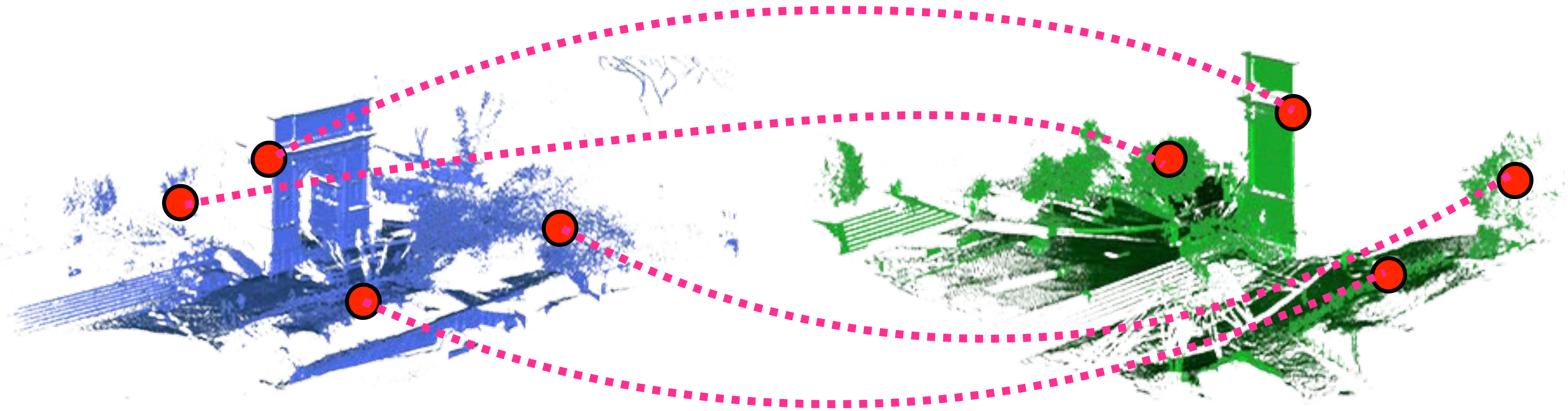
- Alternative to feature-based approaches
- Simultaneous Pose and Correspondences

Iterative Closest Point (ICP)

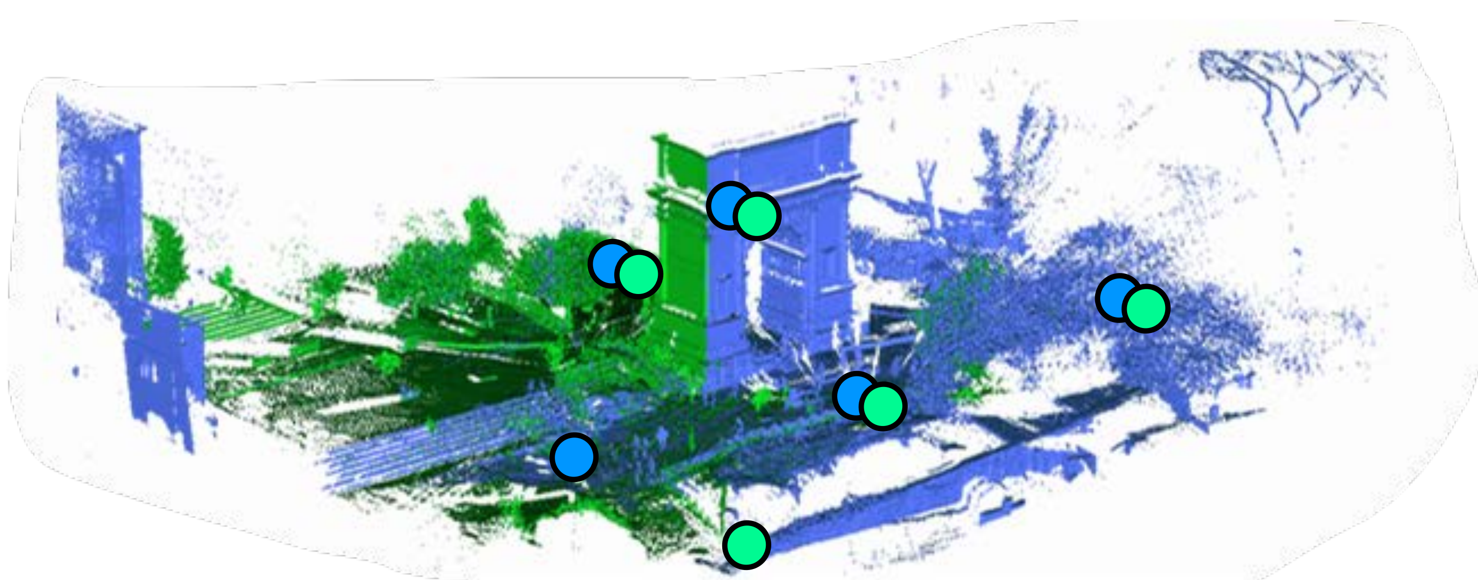
- **Observations:**

$$\min_{\substack{\boldsymbol{R} \in \mathrm{SO}(3) \\ \boldsymbol{t} \in \mathbb{R}^3}} \sum_{k=1}^{N} \|\boldsymbol{p}_{1,k} - (\boldsymbol{R}\boldsymbol{p}_{2,k} + \boldsymbol{t})\|^2$$

1. Easy to compute alignment given ground-truth correspondences



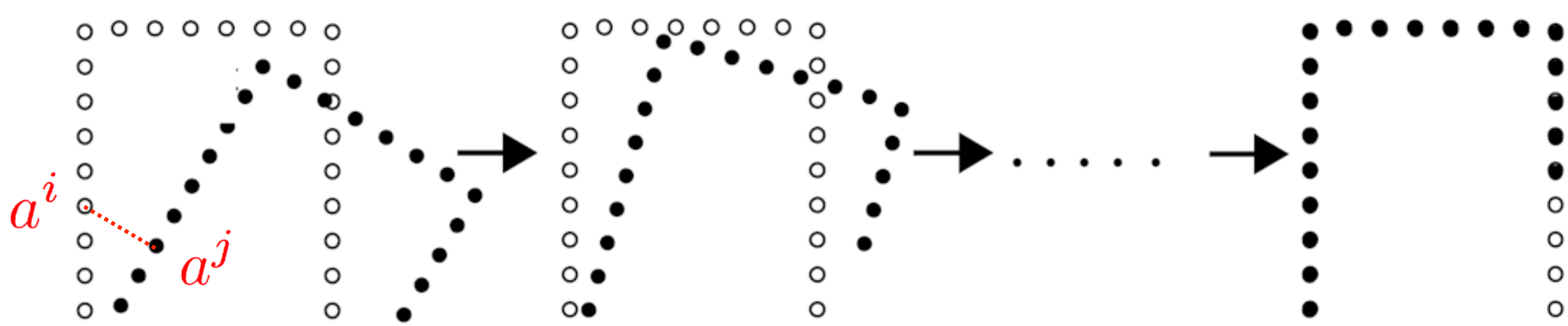
2. Easy to compute correspondences given ground-truth alignment



Iterative Closest Point (ICP)

ICP algorithm: given initial guess, perform the following:

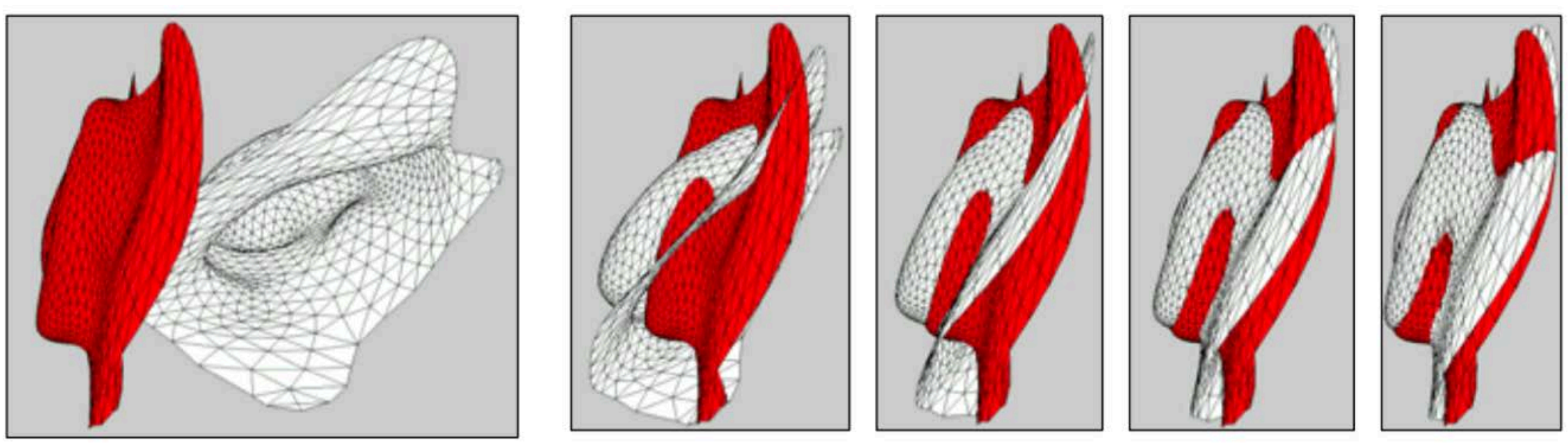
1. **Establish correspondences**: associate to each point in Cloud 1 the closest point in Cloud 2
2. **Compute relative pose given correspondences** (e.g., using Horn's or Arun's method)
3. **Transform point cloud and repeat** (stop when alignment does not improve or after max iter.)

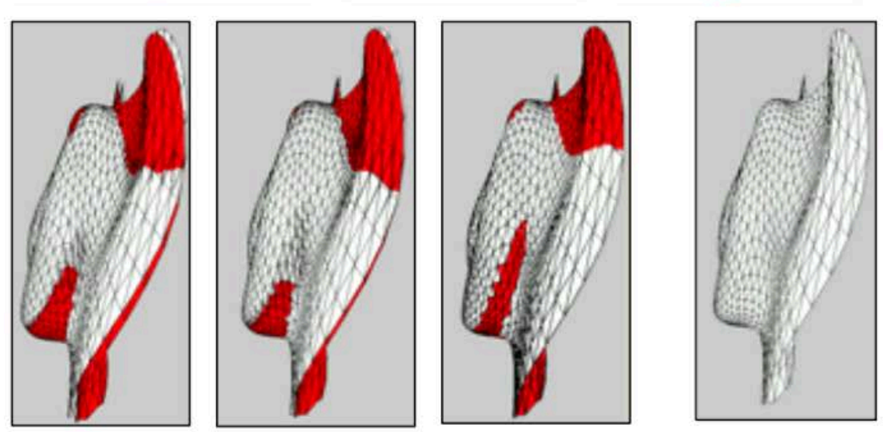


Iterative Closest Point (ICP)

ICP algorithm: given initial guess, perform the following:

1. **Establish correspondences**: associate to each point in Cloud 1 the closest point in Cloud 2
2. **Compute relative pose given correspondences** (e.g., using Horn's or Arun's method)
3. **Transform point cloud and repeat** (stop when alignment does not improve or after max iter.)





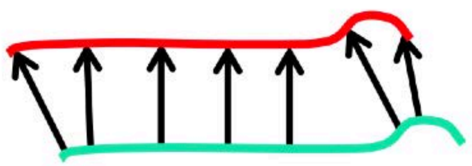
ICP Iterations

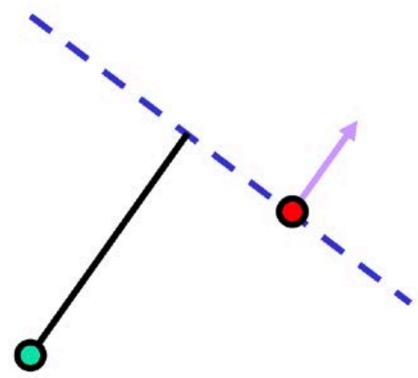
[courtesy: http://www.cs.technion.ac.il/~cs236329/tutorials/ICP.pdf]

Iterative Closest Point (ICP): Issues and Extensions

Extensions

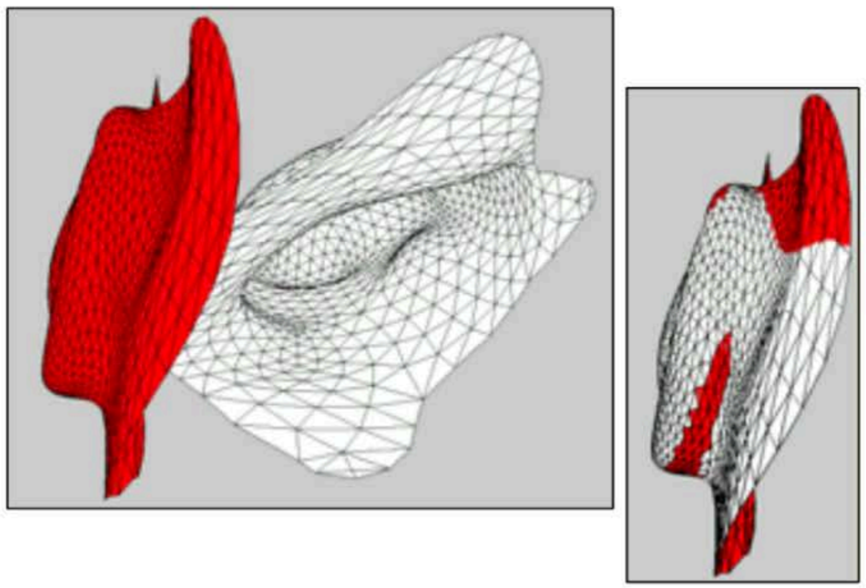
- Kd-tree spatial subdivision
- Different error metrics (e.g., point to plane)
- Reject outliers



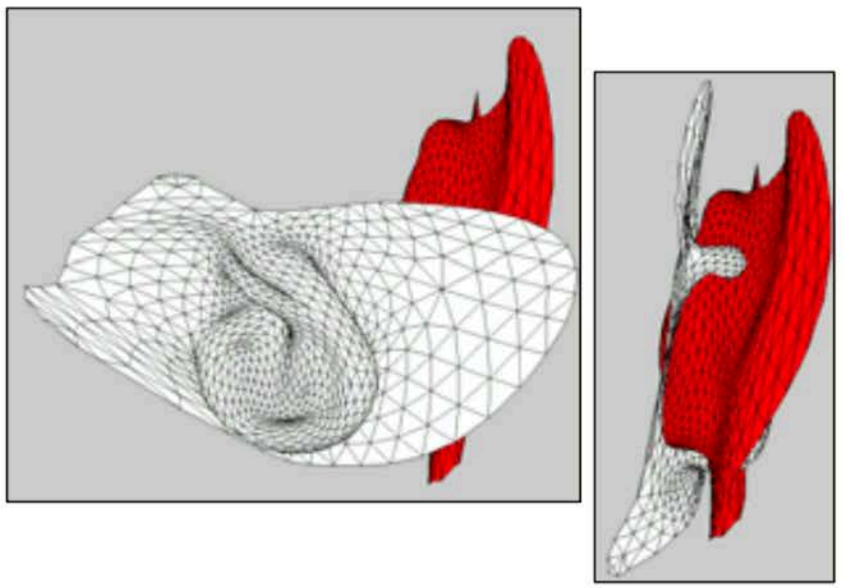


Local convergence

Initial guess 1

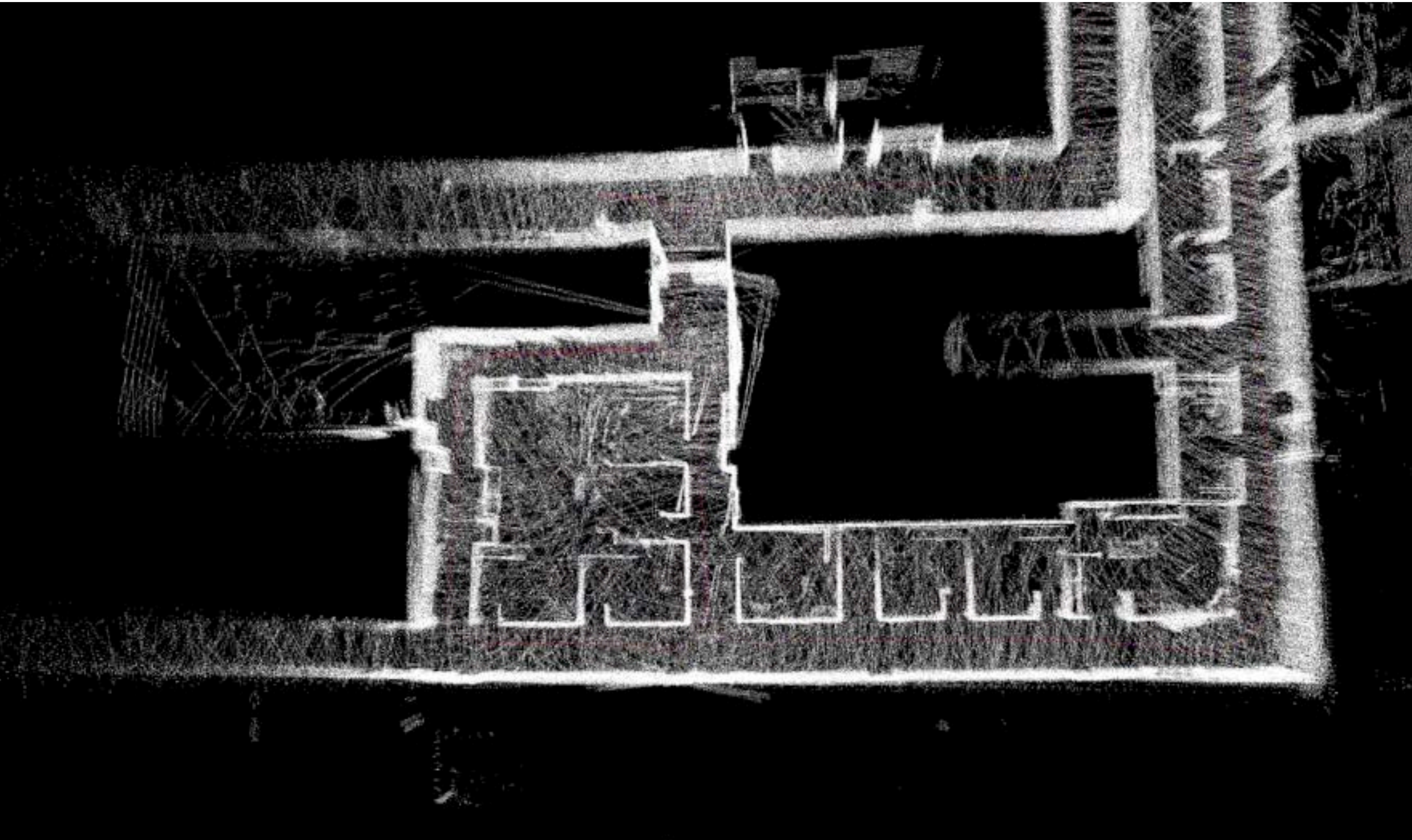


Initial guess 2



[courtesy: http://www.cs.technion.ac.il/~cs236329/tutorials/ICP.pdf]

ICP-based SLAM: Failure Mode



DARPA Subterranean Challenge, in collaboration with JPL

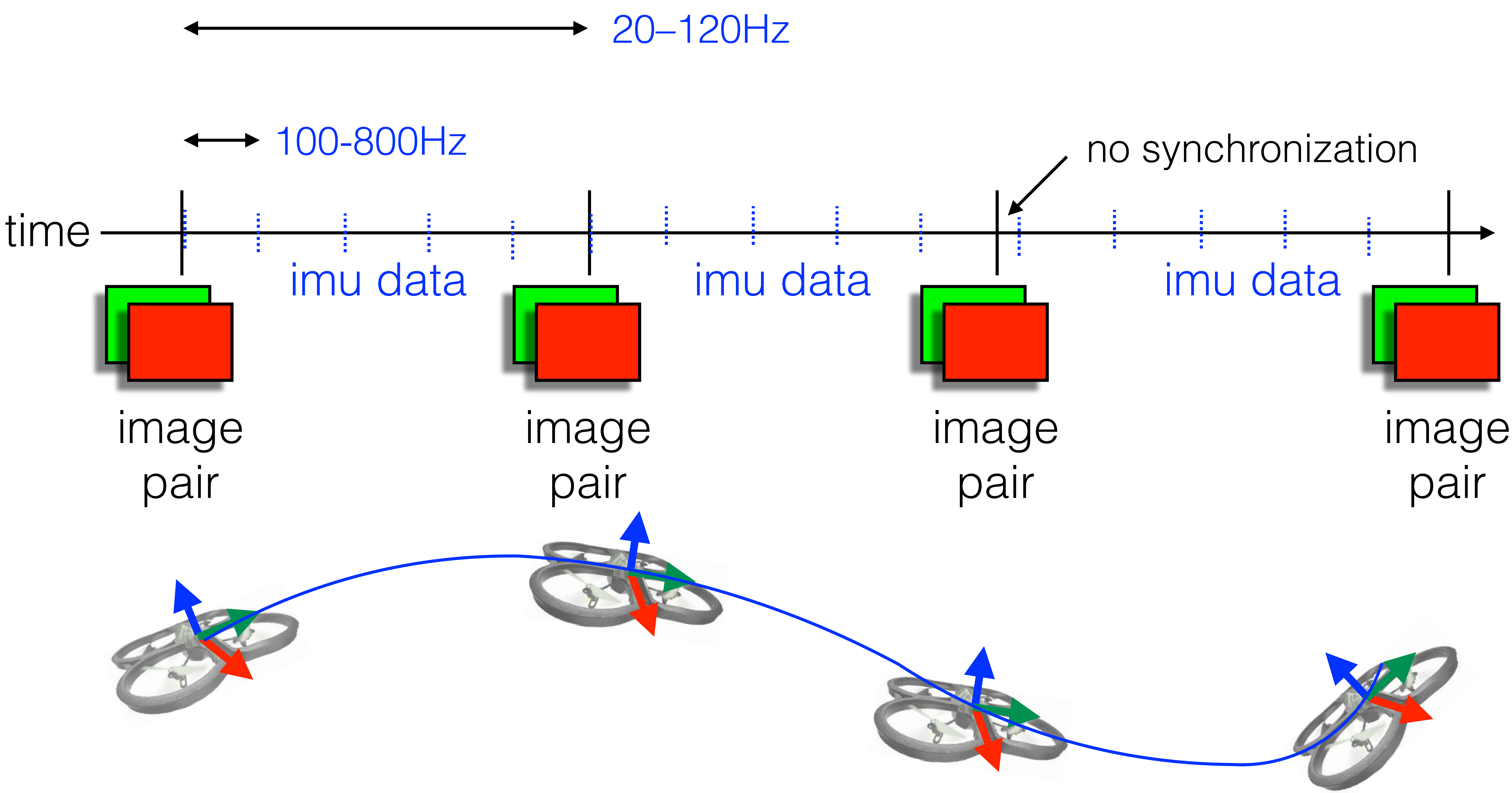
Today: Beyond Cameras

- wheel odometry
- GPS
- Lidar
- Inertial Measurement Unit (IMU)
- Event Cameras



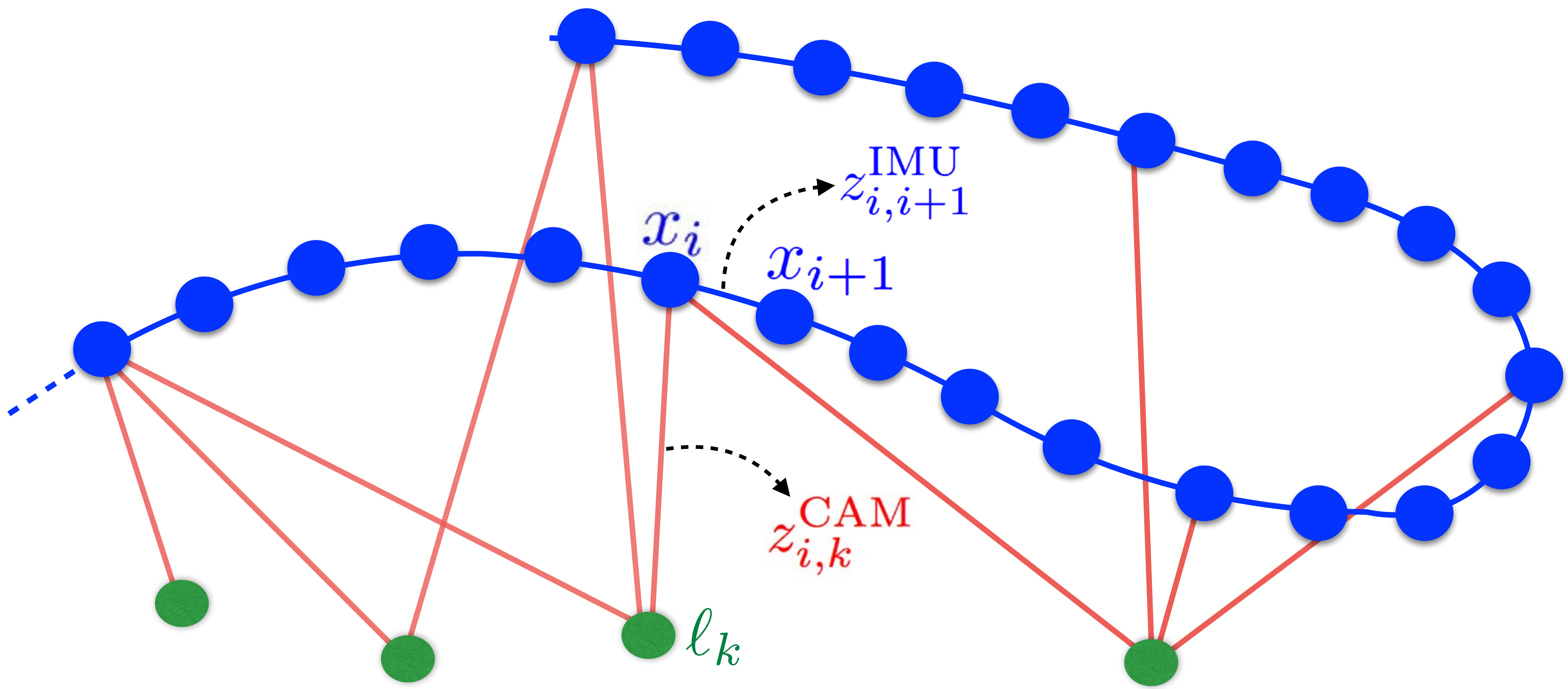
830g	160g	4g	3g
8 W	2.5 W	0.3W	~1 W

Visual-Inertial Odometry



- **Fixed-lag smoother**: estimate a fixed window of recent states from time k-T, k-T+1, .. k (sliding window)

MAP Estimation



Challenges:

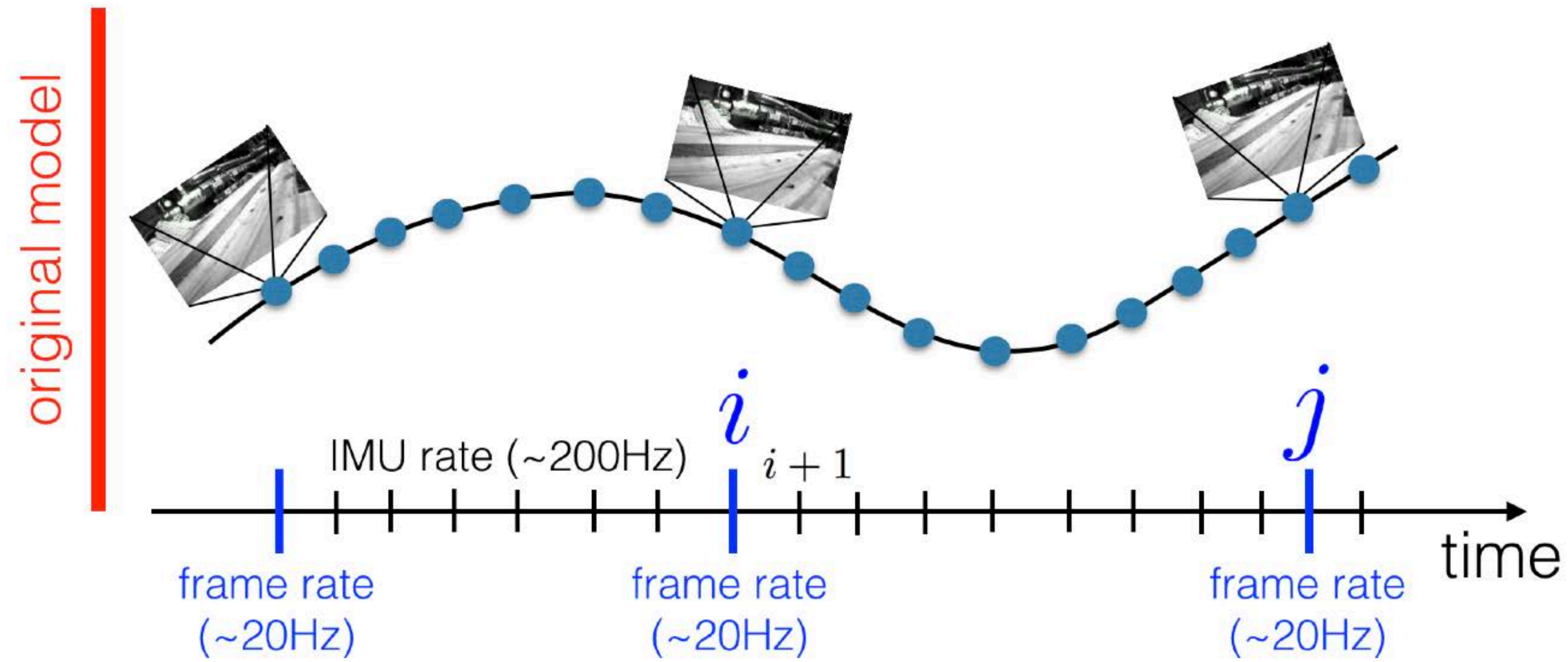
- IMU measurements arrive at high-rate (~200Hz) ⇨ **IMU preintegration**
- camera observes hundreds of landmarks per frame ⇨ **structureless vision factors**
- need to solve optimization problem quickly

IMU Preintegration

Key idea: integrate IMU measurements between frames

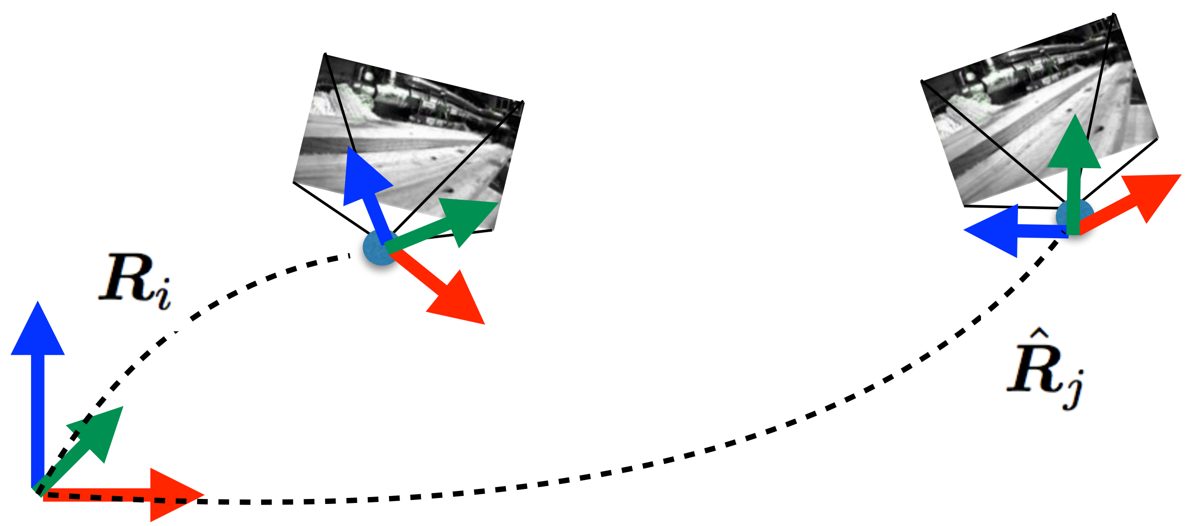
many measurements & states

$$
\begin{aligned}
z^{\mathrm{IMU}}_{i,i+1} &= f(x_i, x_{i+1}) + \epsilon \\
z^{\mathrm{IMU}}_{i+1,i+2} &= f(x_{i+1}, x_{i+2}) + \epsilon \\
&\vdots \\
z^{\mathrm{IMU}}_{j-1,j} &= f(x_{j-1}, x_j) + \epsilon
\end{aligned}
$$



IMU Preintegration

Standard integration

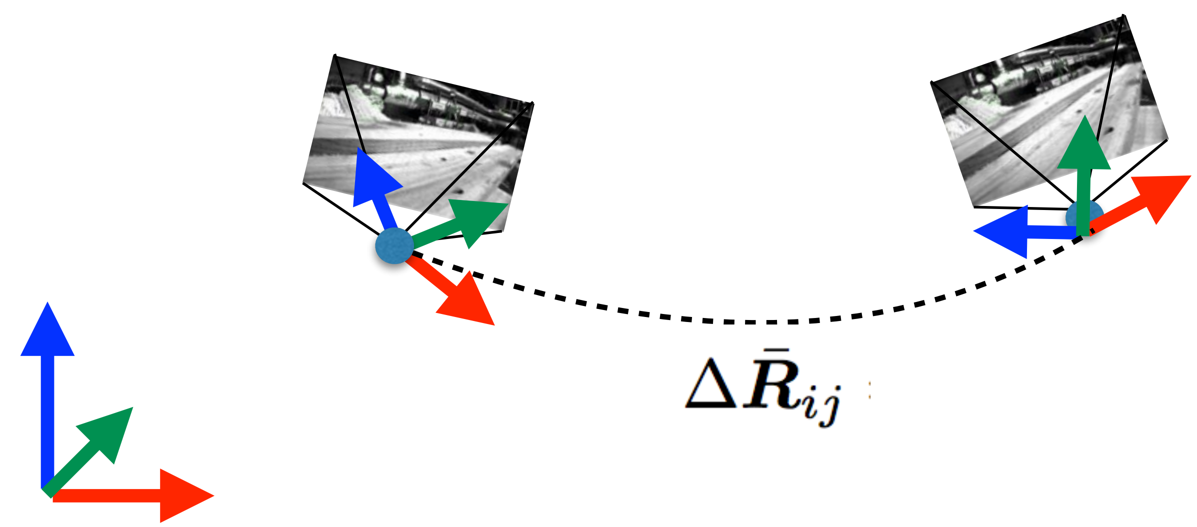


$$\hat{\boldsymbol{R}}_j = \boldsymbol{R}_i \cdot \mathrm{Exp}\left(w_{i,i+1}\delta t\right) \cdots \mathrm{Exp}\left(w_{j-1,j}\delta t\right)$$

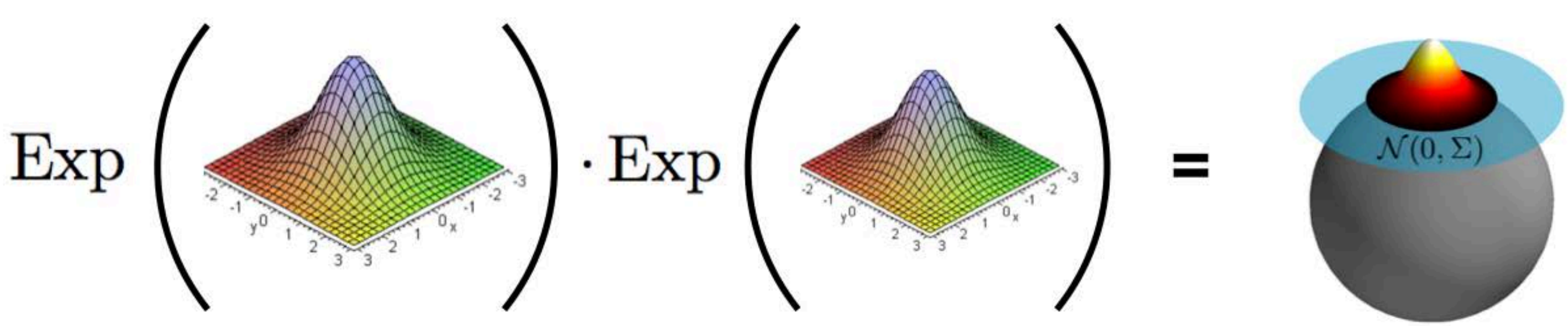
initial rotation

rotation rate measurements

Preintegration

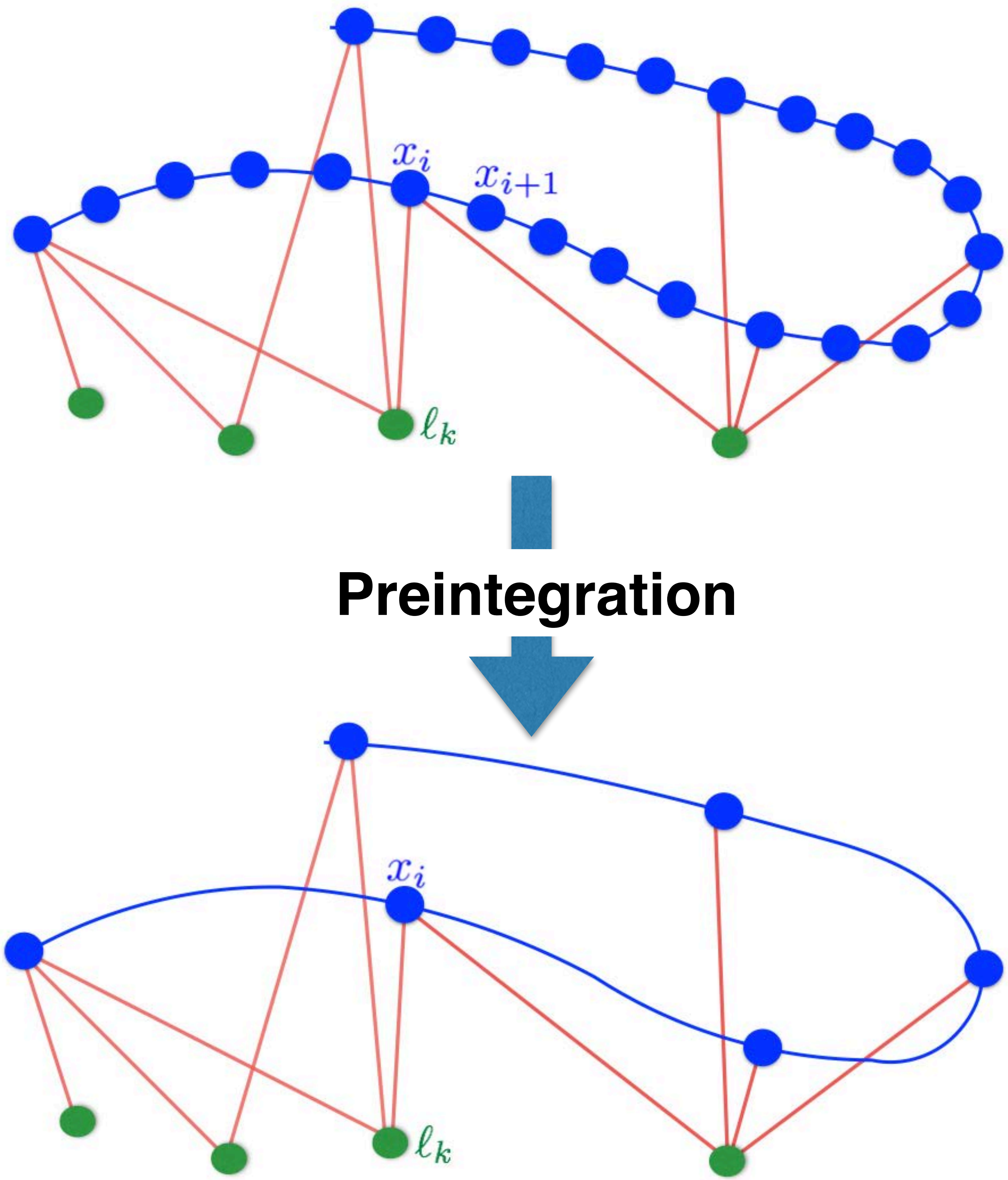


$$\Delta\bar{\boldsymbol{R}}_{ij} = \mathrm{Exp}\left(w_{i,i+1}\delta t\right) \cdots \mathrm{Exp}\left(w_{j-1,j}\delta t\right)$$



Carlone, Kira, Beall, Indelman, Dellaert, Eliminating conditionally independent sets in factor graphs: a unifying perspective based on smart factors, ICRA'14.
Forster, Carlone, Dellaert, Scaramuzza, *IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation*, RSS'15 (best paper finalist)

Pre-integration



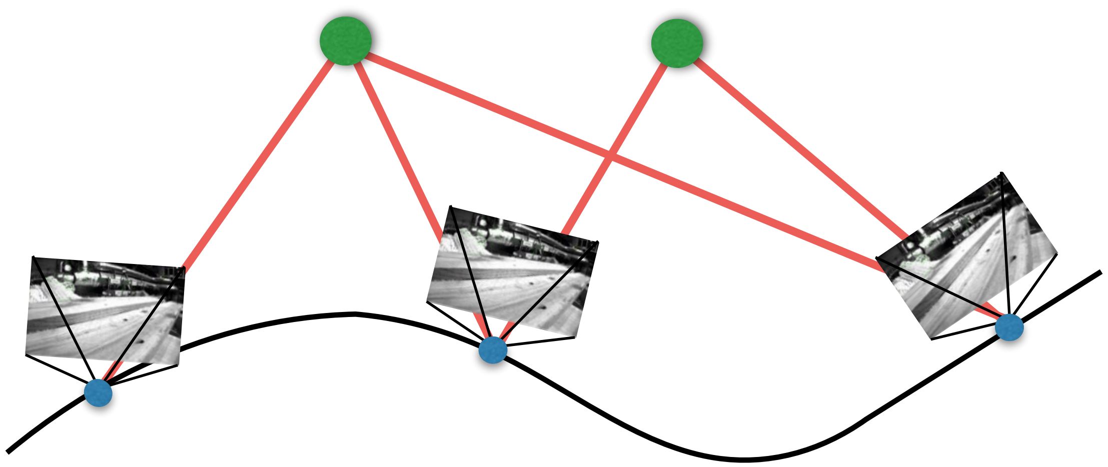
After 10 seconds, original problem has ~10^4 states

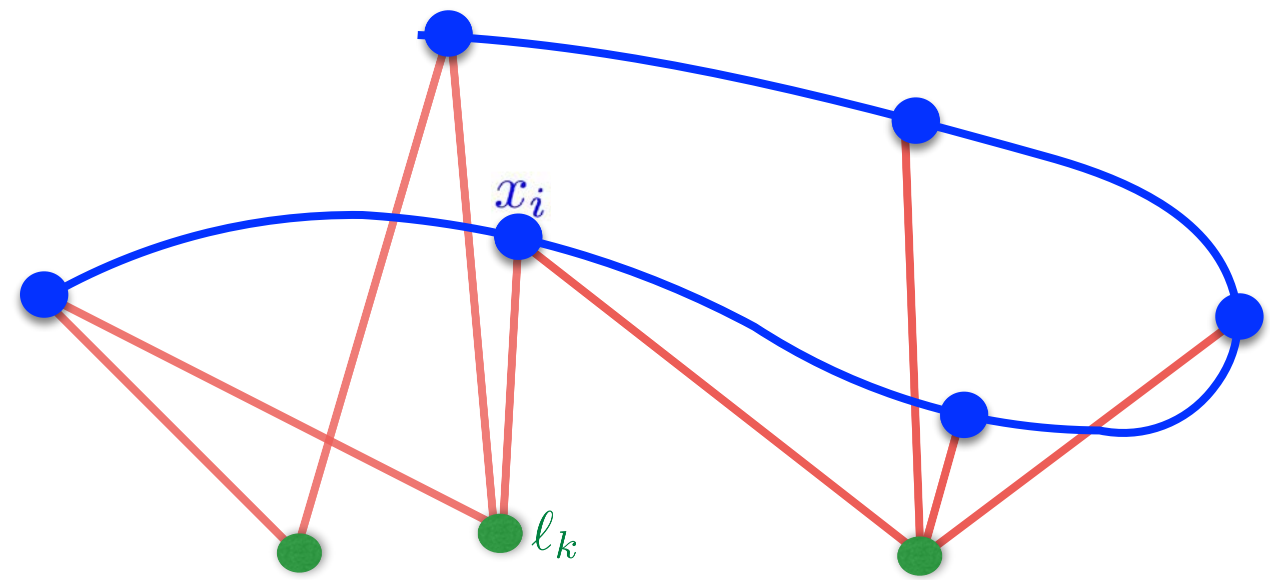
After 10 seconds, preintegrated problem has ~10^2 states

[Forster, Carlone, Dellaert, Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *TRO* 2017]

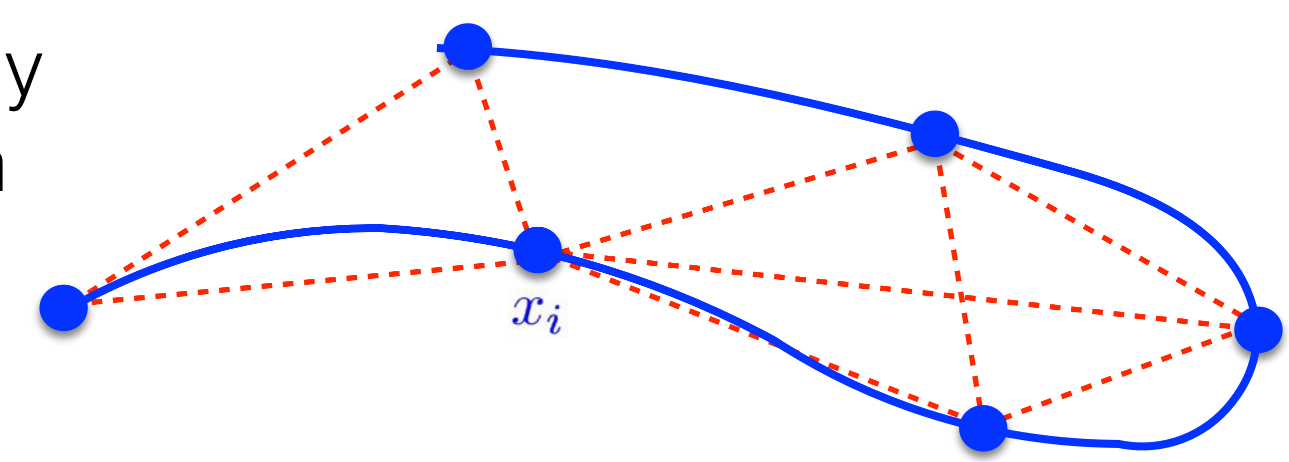
Structureless Vision Model

Marginalization
of 3D landmarks





Schur complement

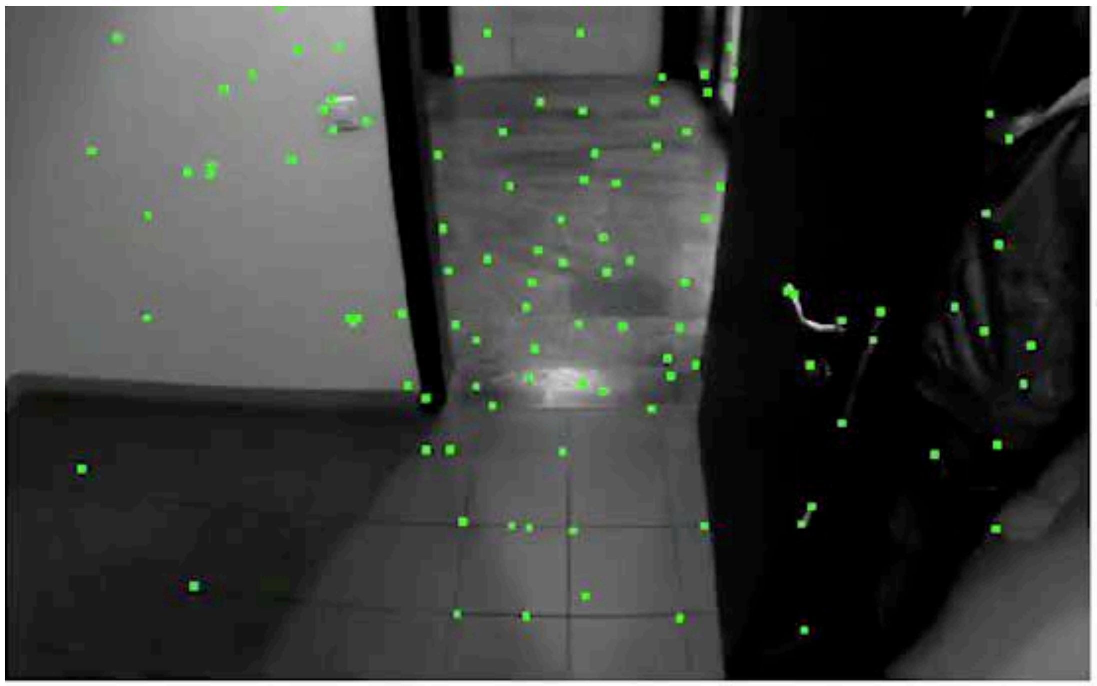


Schur complement trick:
- solve for each landmark separately
- substitute back in the optimization

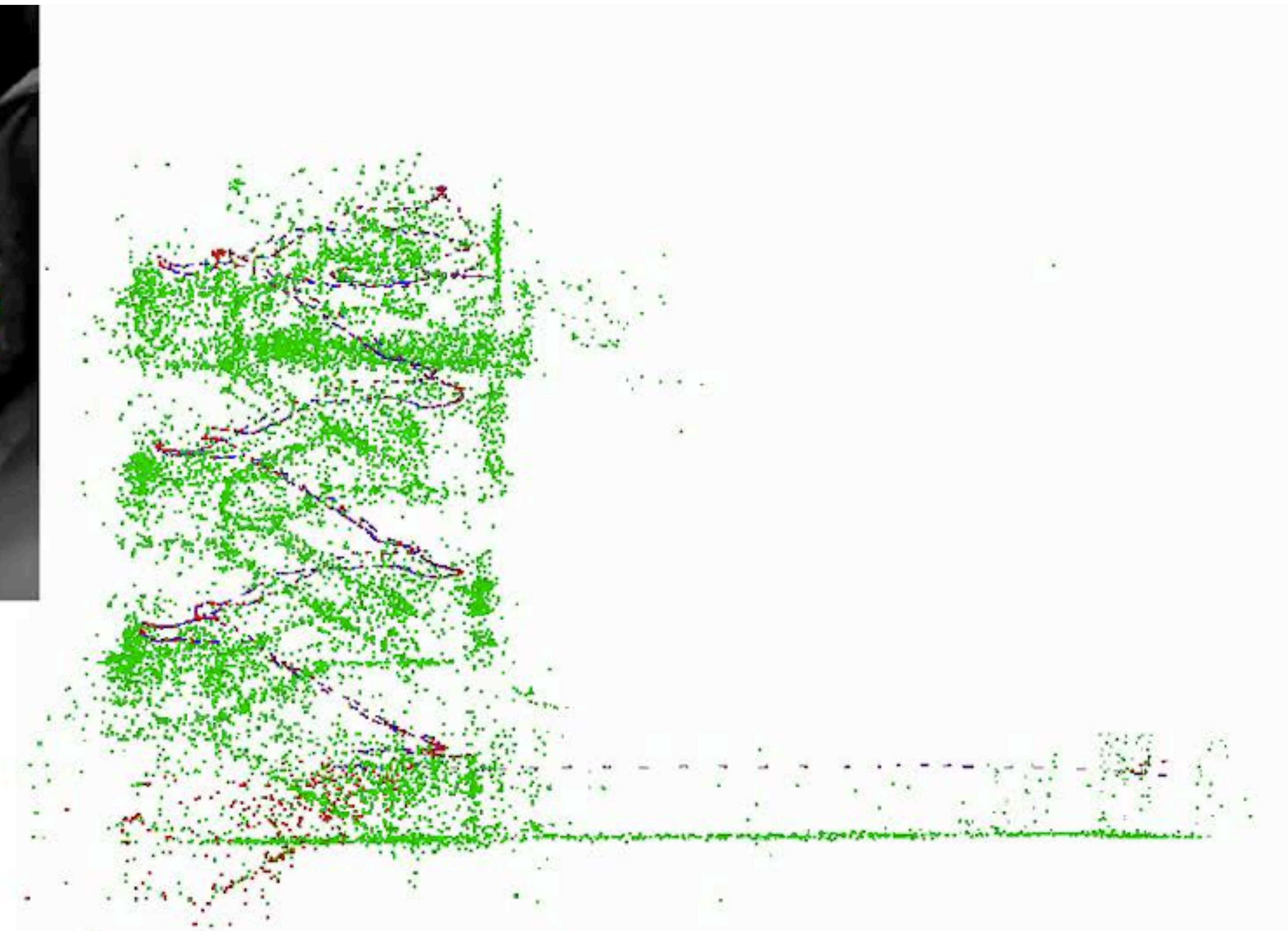
Further reduction of the number of variables in the optimization!

Carlone, Alcantarilla, Chiu, Zsolt, Dellaert, *Mining structure fragments for smart bundle adjustment*, BMVC'14.

Visual-Inertial Odometry



Hand-held sensor

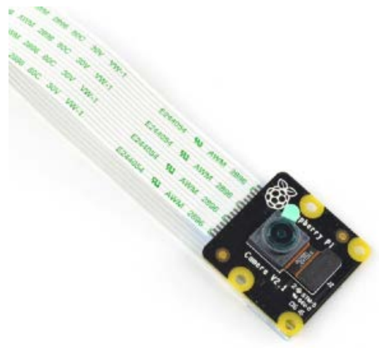


[Forster, Carlone, Dellaert, Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *TRO* 2017]

Today: Beyond Cameras

- wheel odometry
- GPS
- Lidar
- Inertial Measurement Unit (IMU)
- Event Cameras





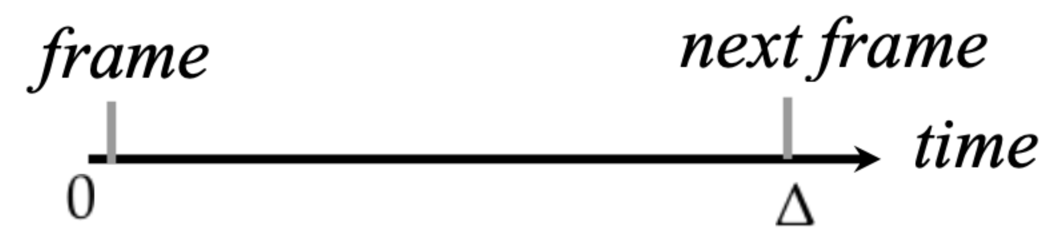
830g	160g	4g	3g
8 W	2.5 W	0.3W	~1 W

Event-based Vision: A Survey

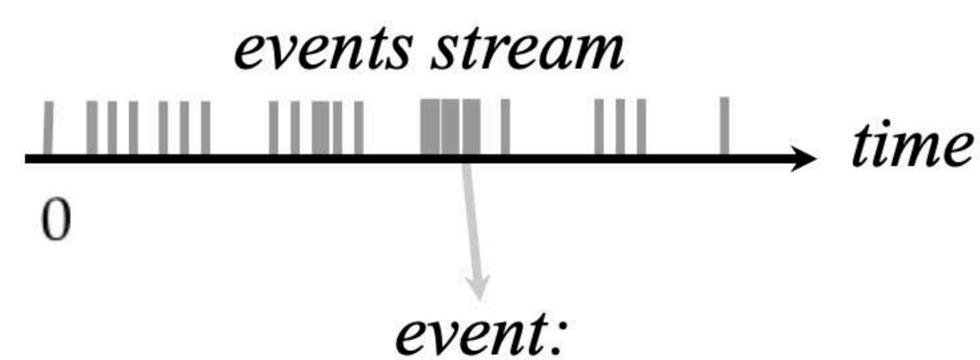
Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, Davide Scaramuzza

Event-based Cameras

- Speed of robot is constrained by speed at which it can sense (and compute)
- Common cameras: 20-120fps



- event-based cameras (e.g., Dynamic Vision Sensor, DVS)
  - Temporal resolution: 1 μs
  - High dynamic range: 120 dB
  - Low power: 20 mW
  - Cost: 2,500 EUR

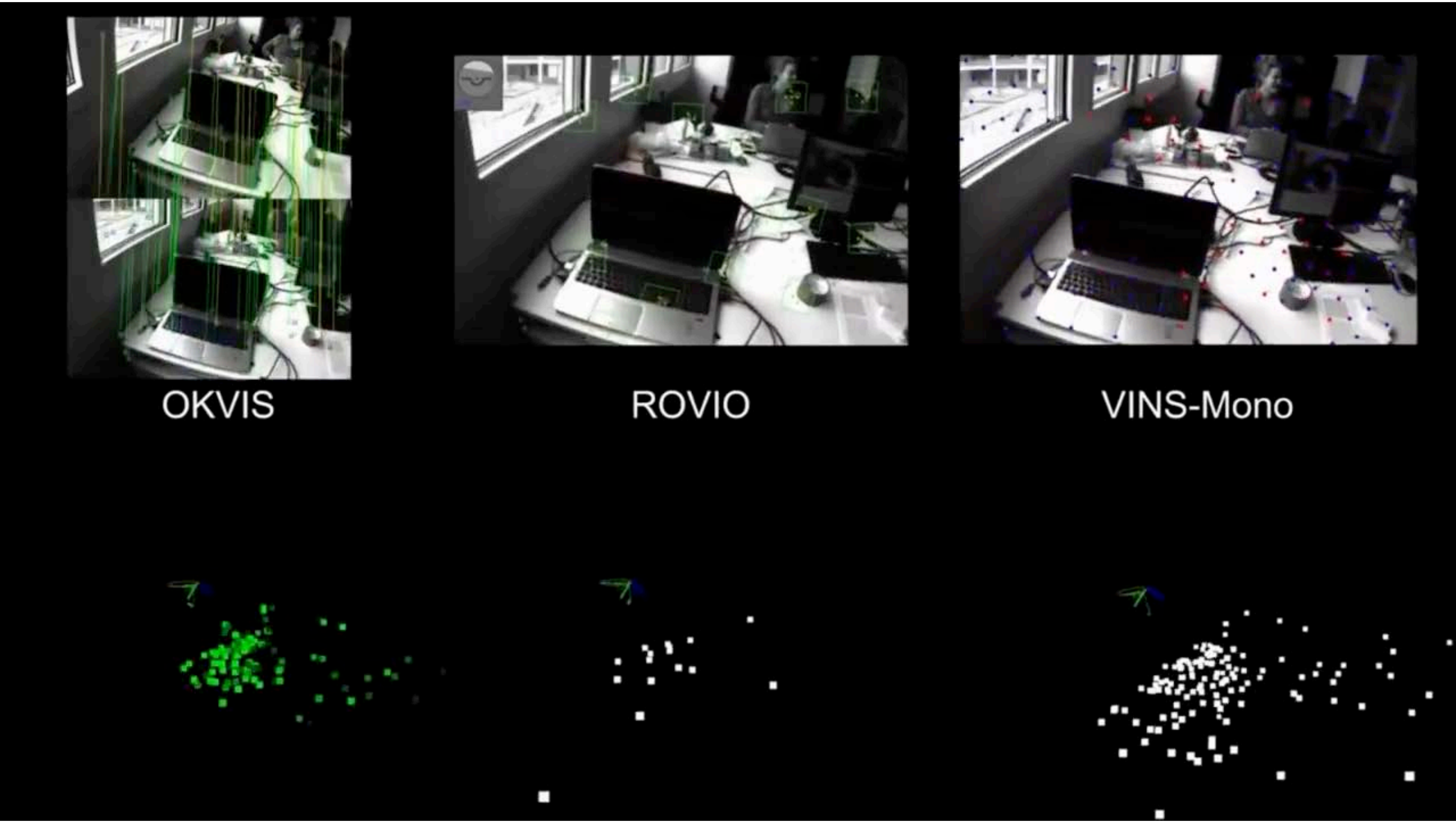


$$\left\langle t, \langle x, y \rangle, sign\left(\frac{d}{dt}\log(I_t(x,y))\right)\right\rangle$$

Event-based Cameras

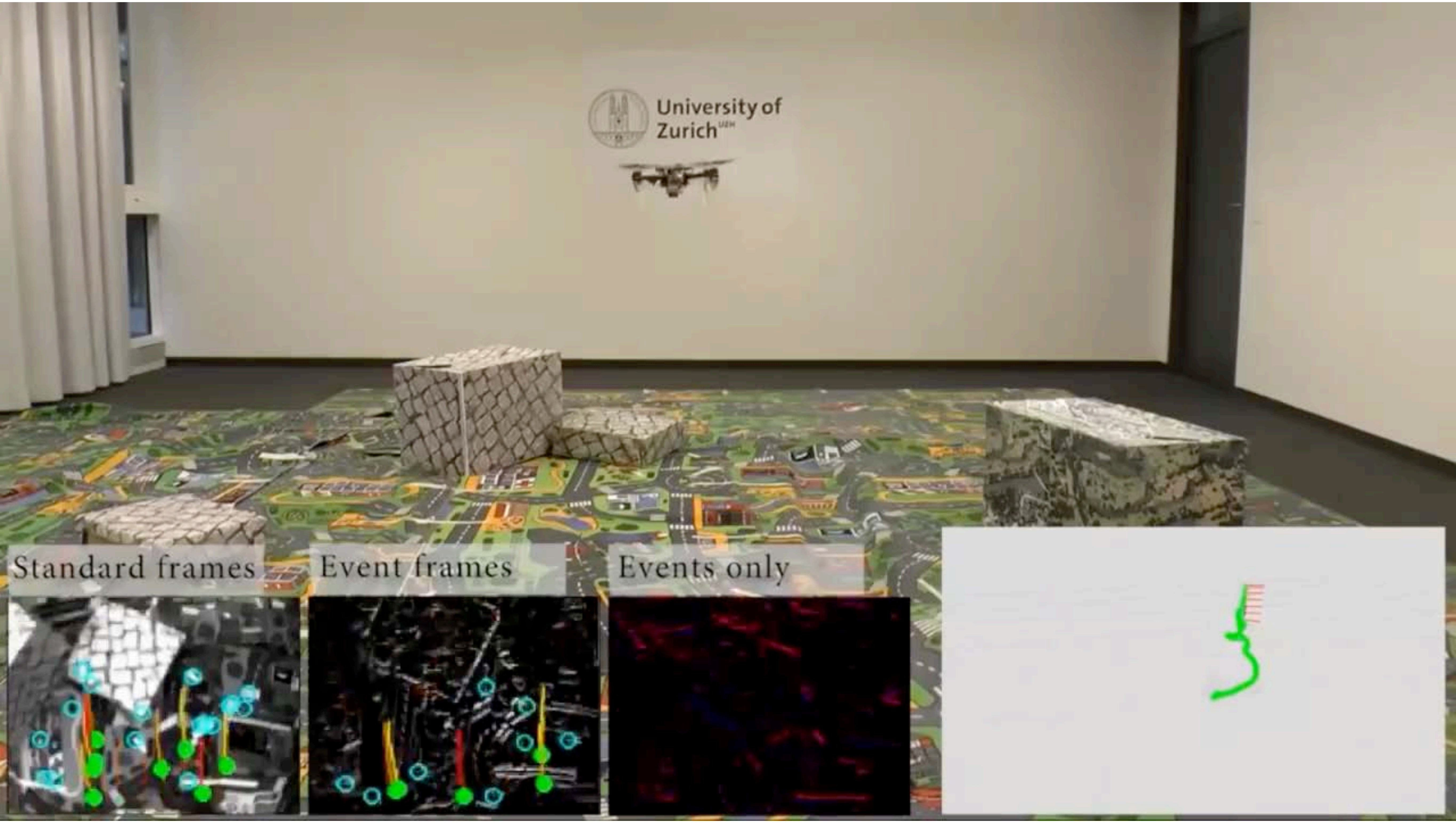
Scaramuzza et al. 2014: Neuromorphic camera

Event-based Cameras for SLAM



Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, Davide Scaramuzza Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High Speed Scenarios R-AL 2018.[30]

Event-based Cameras for SLAM



Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, Davide Scaramuzza Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High Speed Scenarios R-AL 2018.[31]

Next Lecture

Overview of Open Problems
in Robot Perception

32

Zoom poll

Feedback on this Zoom lecture (single-choice):

- A: audio and video are good!
- B: audio and video are adequate (sometimes you break up)
- C: audio quality is very bad
- D: audio AND video quality is bad :-(