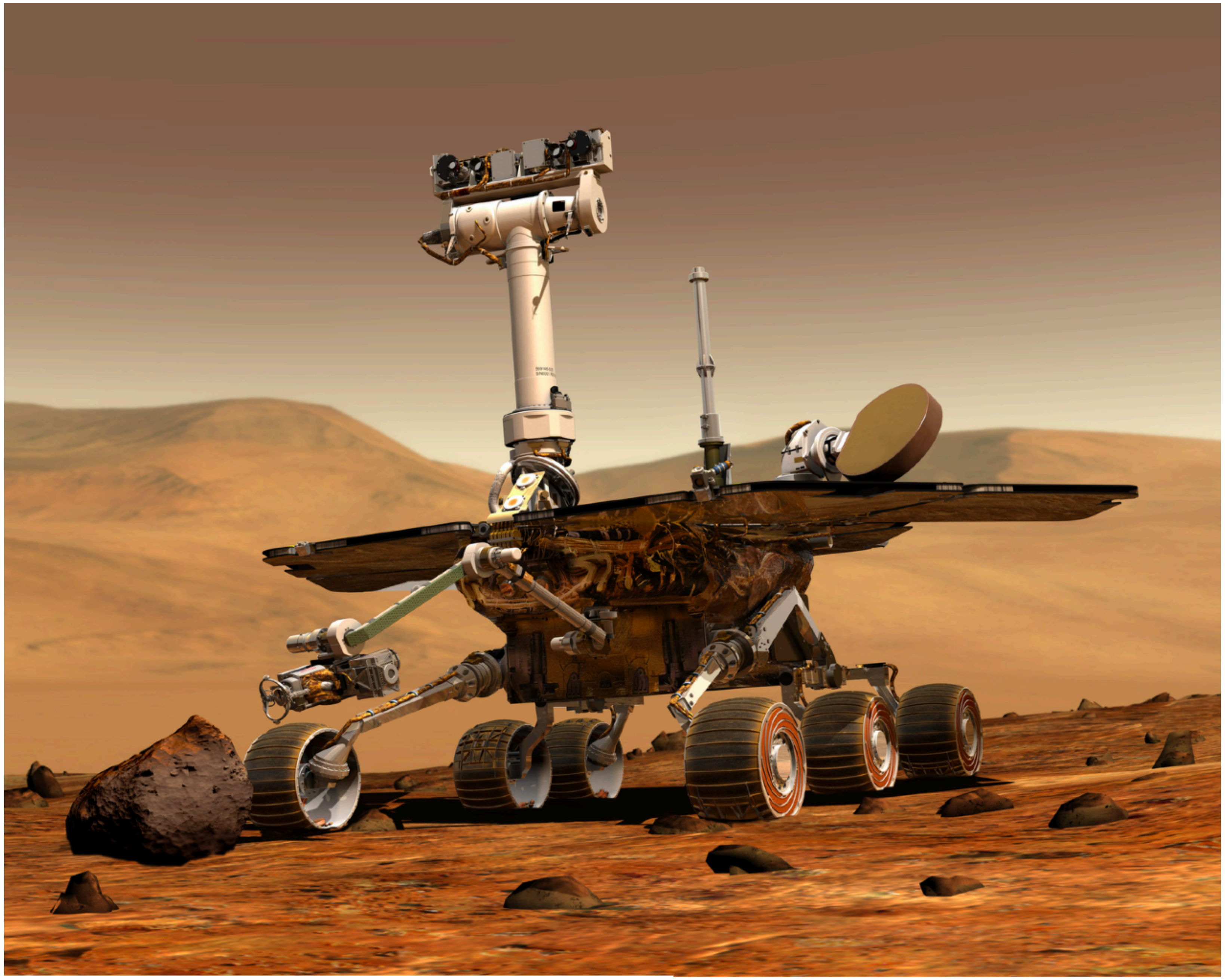


Source: public domain. Courtesy of NASA/JPL/Cornell University.

16.485: VNAV - Visual Navigation for Autonomous Vehicles

Luca Carlone

Lecture 20: Visual and Visual-Inertial Odometry

Where are we?

Week	Dates	Lecture topic	Lab
1	Sep 8, 10	Introduction	Lab 1: Linux, C++, Git
2	Sep 13, 15, 17	3D Geometry	Lab 2: ROS
3	Sep 20, 22, 24	Geometric Control	Lab 3: 3D trajectory following
4	Sep 27, 29	Trajectory Optimization	Lab 4: 3D trajectory optimization
5	Oct 1, 4, 6	2D Computer Vision	Lab 5: feature detection
6	Oct 8, 13, 15	2-view Geometry and Minimal Solvers	Lab 6: object localization
7	Oct 18, 20, 22	Non-minimal Solvers and Visual Odometry	Lab 7: GTSAM
8	Oct 25, 27, 29	Place Recognition	Lab 8: ML for robotics
9	Nov 1, 3, 5	SLAM and Visual-Inertial Navigation	Lab 9: SLAM
10	Nov 8, 10, 12	Advanced Topics: Open Problems in Robot Perception	Final project
11	Nov 15, 17, 19	Advanced Topics: Robustness	Final project
12	Nov 22, 24, 29, Dec 1	Advanced Topics: Metric-Semantic Understanding and Learning	Final project
13	Nov 25-26	Thanksgiving Break	
14	Dec 3, 6, 8	Guest Lectures and Students Presentations	Final project

Today

- VO: **V**isual **O**dometry
- VIO: **V**isual-**I**nertial **O**dometry
- (Beyond vision)

Visual Odometry

Part I: The First 30 Years and Fundamentals

Part II: Matching, Robustness, Optimization, and Applications

By Friedrich Fraundorfer and Davide Scaramuzza

On-Manifold Preintegration for Real-Time Visual-Inertial Odometry

Christian Forster, Luca Carlone, Frank Dellaert, Davide Scaramuzza

***Abstract*—Current approaches for visual-inertial odometry (VIO) are able to attain highly accurate state estimation via nonlinear optimization. However, real-time optimization quickly becomes infeasible as the trajectory grows over time; this problem is further emphasized by the fact that inertial measurements come at high rate, hence leading to fast growth of the number of variables in the optimization. In this paper, we address this issue by preintegrating inertial measurements between selected keyframes into single relative motion constraints. Our first contribution is a *preintegration theory* that properly addresses the manifold structure of the rotation group. We formally discuss the generative measurement model as well as the nature of the rotation noise and derive the expression for the *maximum a posteriori* state estimator. Our theoretical development enables the computation of all necessary Jacobians for the optimization and a-posteriori bias correction in analytic form. The second contribution is to show that the preintegrated IMU model can be seamlessly integrated into a visual-inertial pipeline under the unifying framework of factor graphs. This enables the application of incremental-smoothing algorithms and the use of a *structureless* model for visual measurements, which avoids optimizing over the 3D points, further accelerating the computation. We perform an extensive evaluation of our monocular VIO pipeline on real and simulated datasets. The results confirm that our modelling effort leads to accurate state estimation in real-time, outperforming state-of-the-art approaches.**

of monocular vision and gravity observable [1] and provides robust and accurate inter-frame motion estimates. Applications of VIO range from autonomous navigation in GPS-denied environments, to 3D reconstruction, and augmented reality.

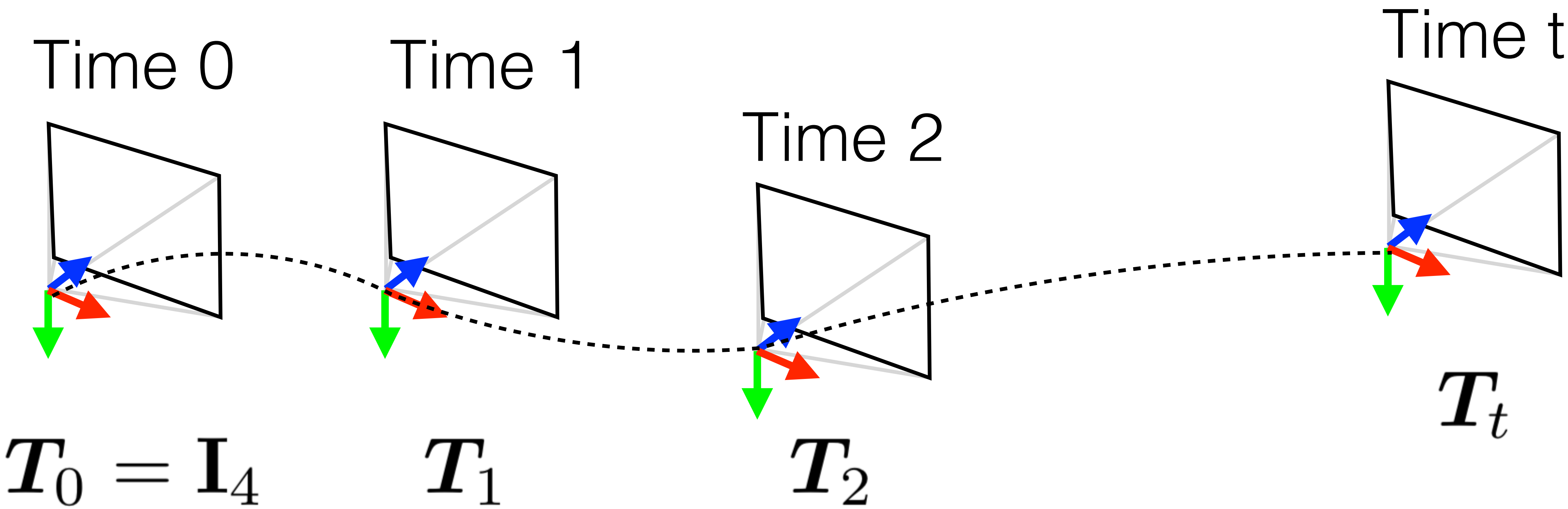
The existing literature on VIO imposes a trade-off between accuracy and computational efficiency (a detailed review is given in Section II). On the one hand, filtering approaches enable fast inference, but their accuracy is deteriorated by the accumulation of linearization errors. On the other hand, full smoothing approaches, based on nonlinear optimization, are accurate, but computationally demanding. Fixed-lag smoothing offers a compromise between accuracy for efficiency; however, it is not clear how to set the length of the estimation window so to guarantee a given level of performance.

In this work we show that it is possible to overcome this trade-off. We design a VIO system that enables fast incremental smoothing and computes the optimal *maximum a posteriori* (MAP) estimate in real time. An overview of our approach is given in Section IV.

The first step towards this goal is the development of a novel preintegration theory. The use of *preintegrated IMU measurements* was first proposed in [2] and consists of combining

Visual Odometry

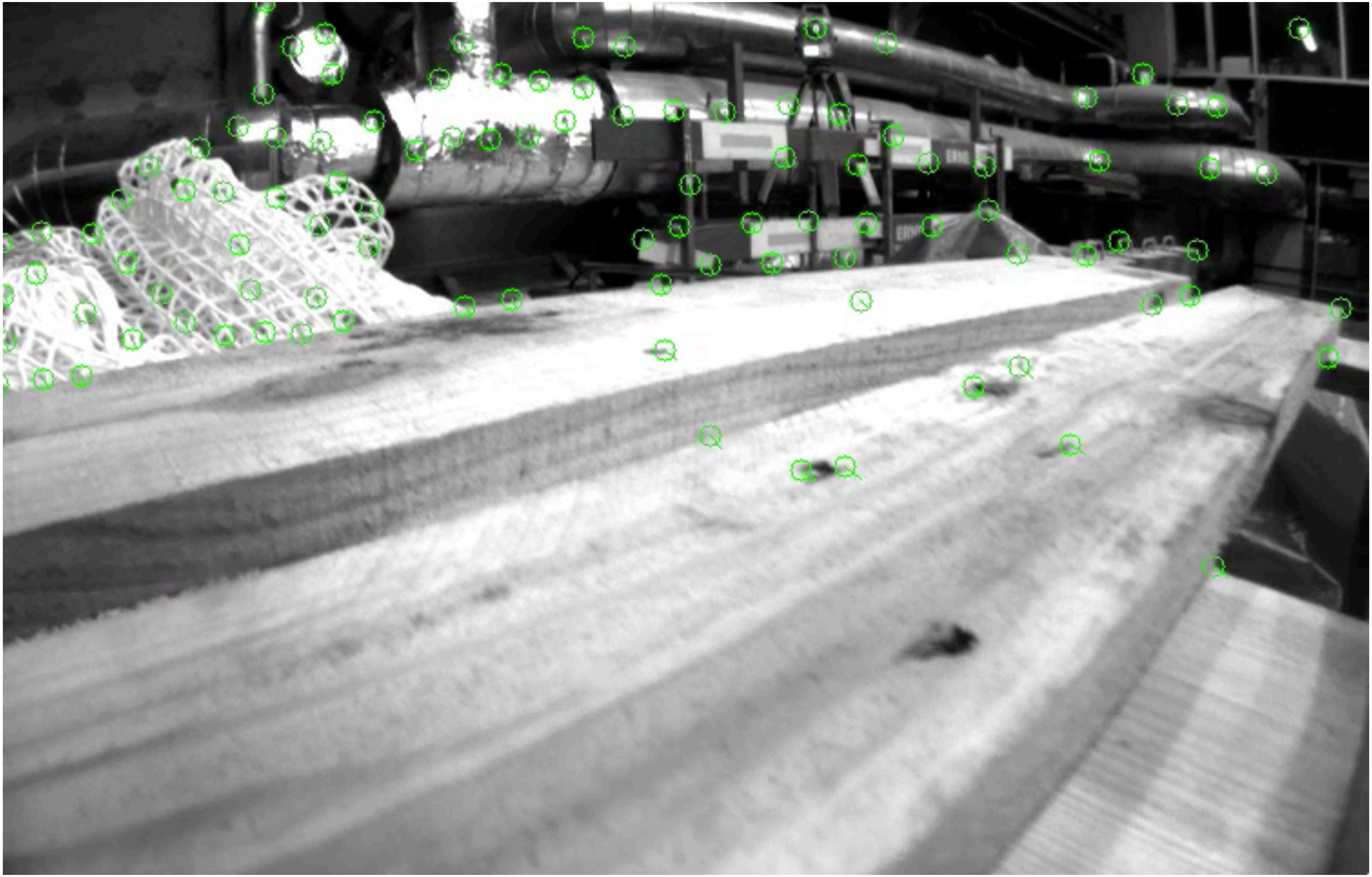
odometry: incremental motion estimation



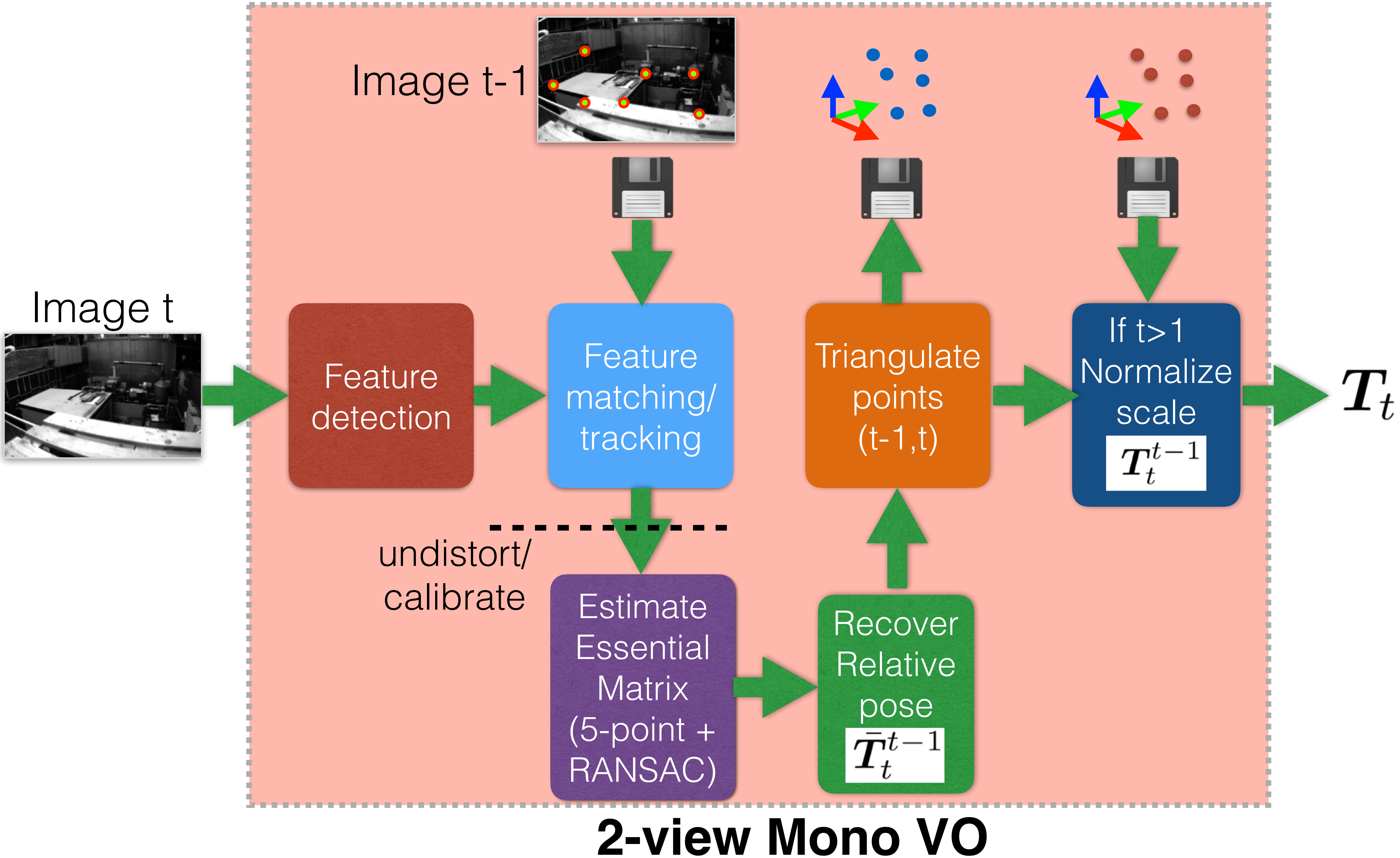
Visual odometry (VO): motion estimation estimation based on cameras (monocular, stereo, RGB-D, …)

others: wheel odometry, inertial, visual-inertial

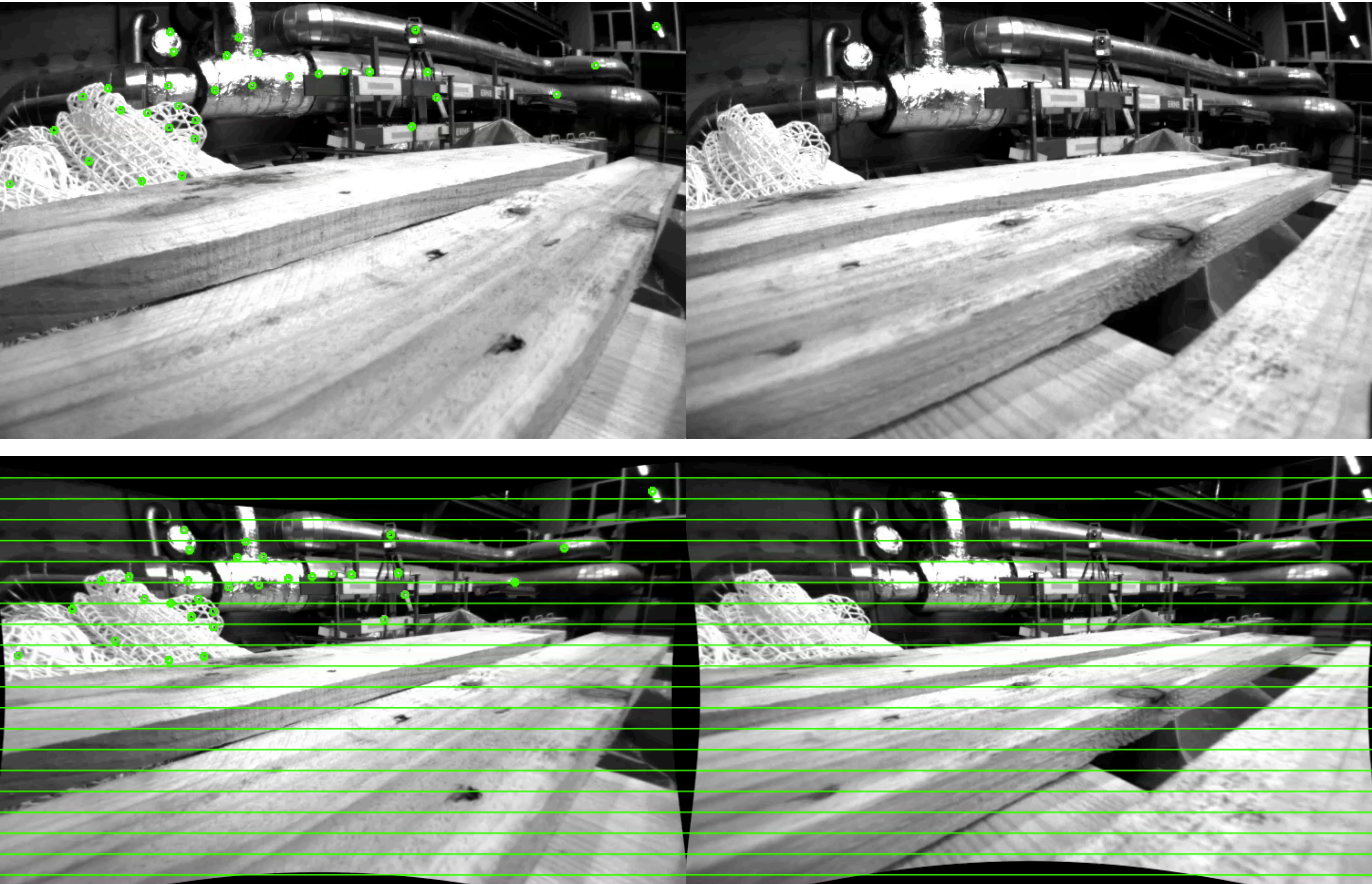
Feature Tracking



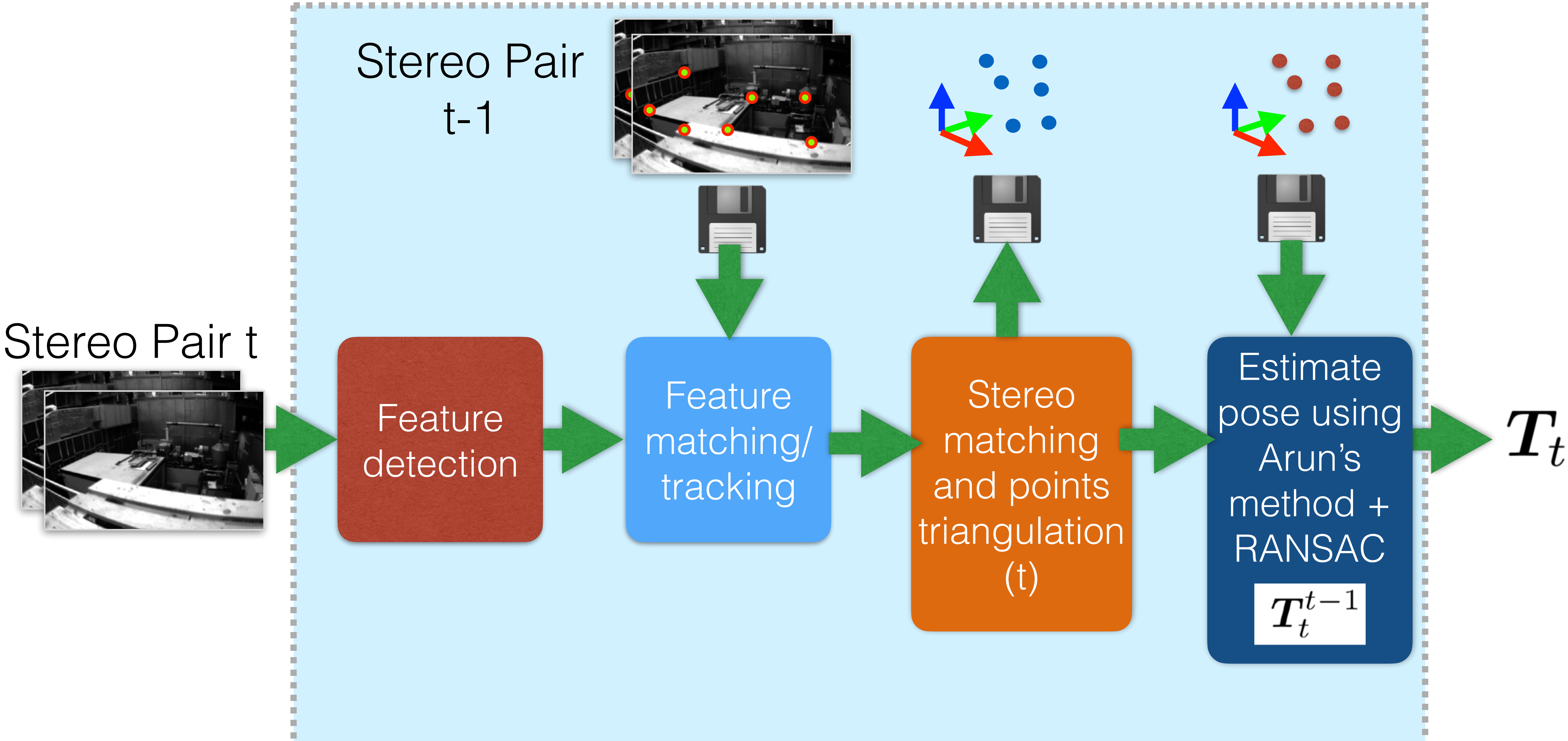
Monocular VO with 2D-2D Correspondences



Stereo Matching



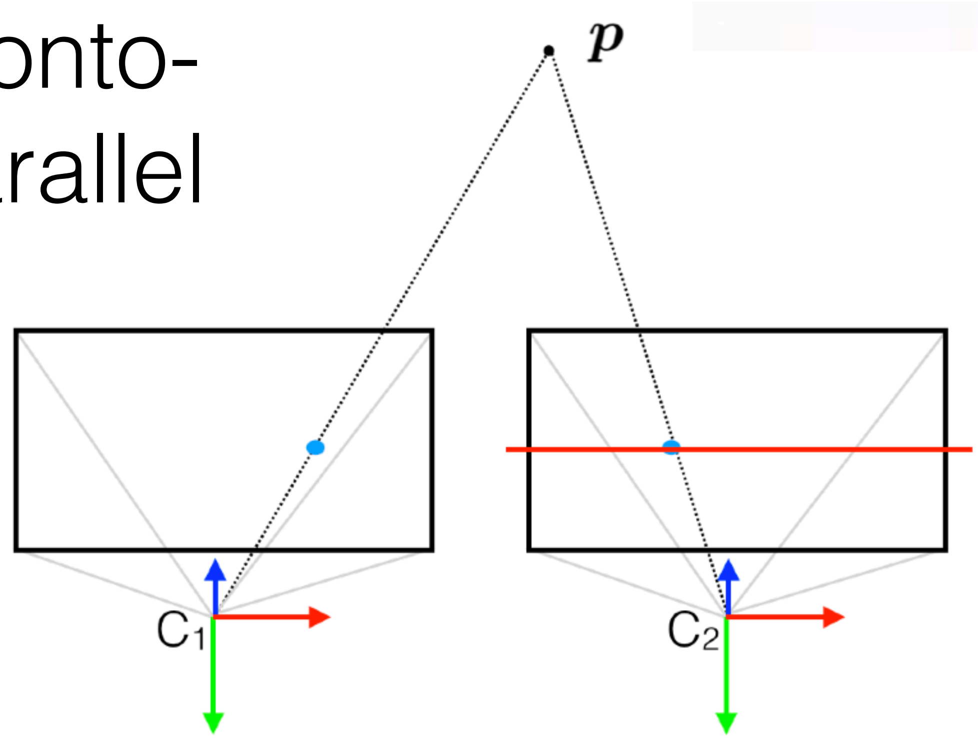
Stereo VO with **3D-3D** Correspondences

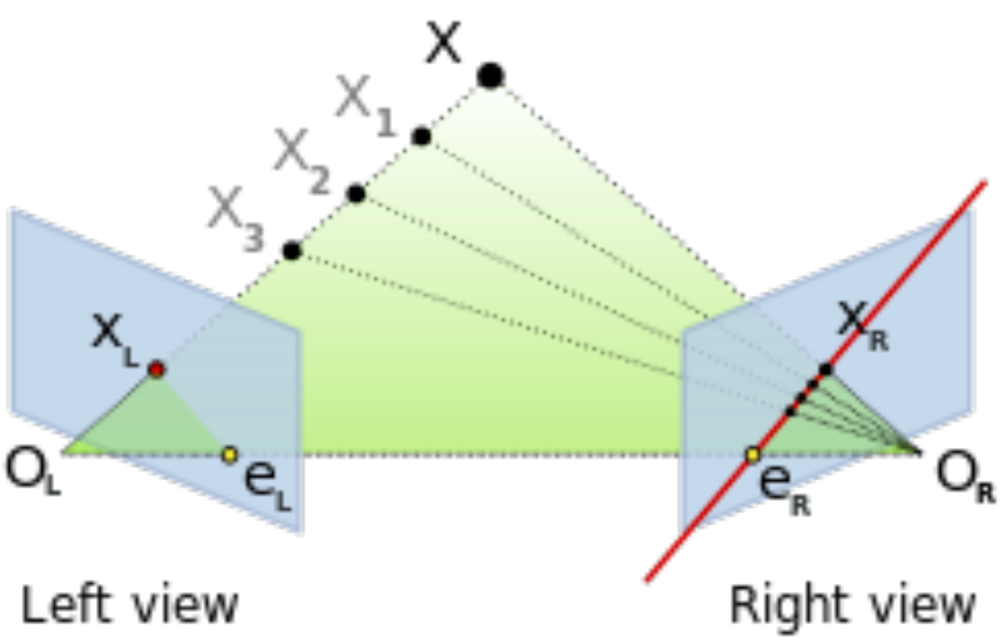


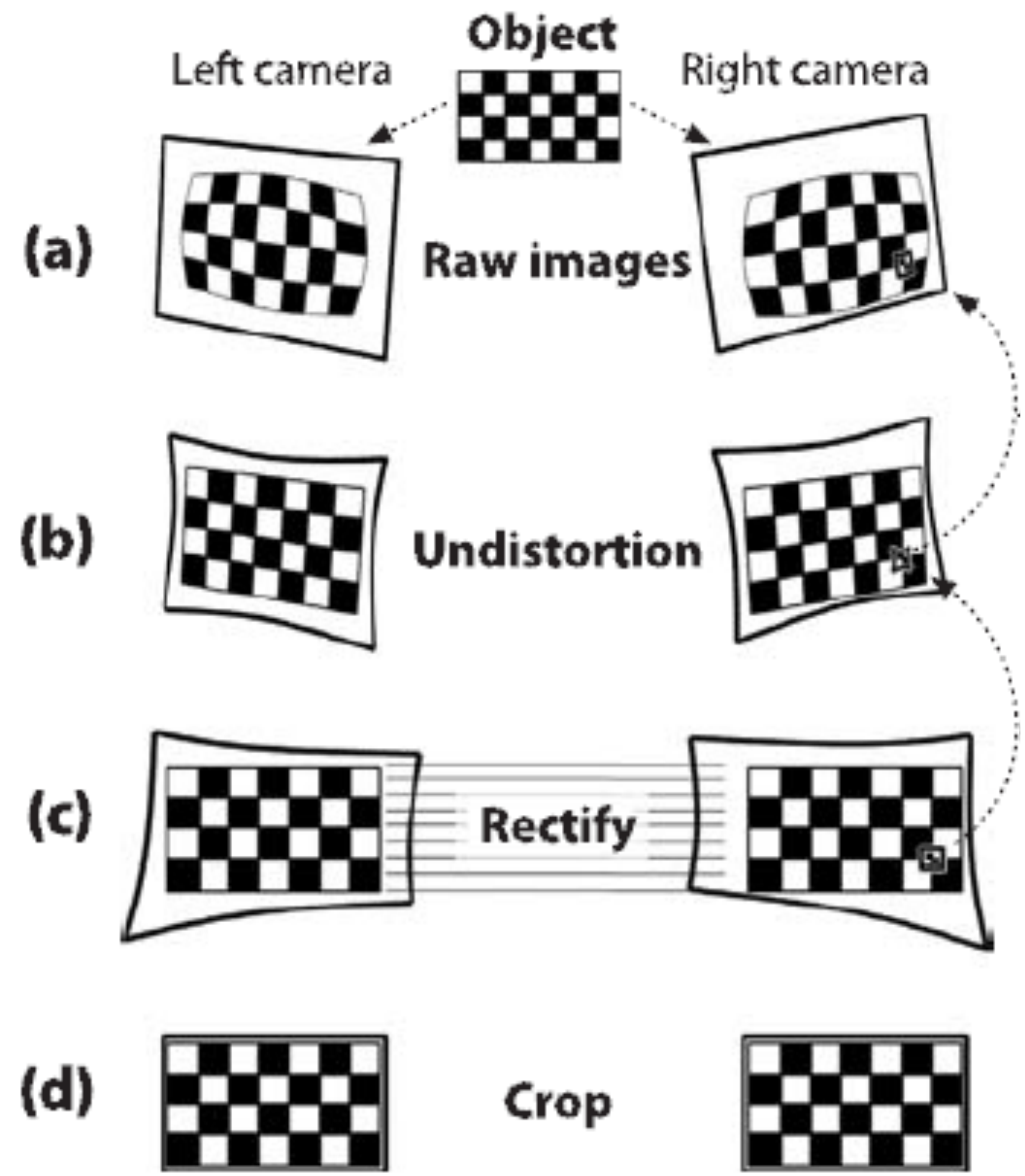
2-view Stereo VO

(Parenthesis on Stereo Matching)

Fronto-parallel

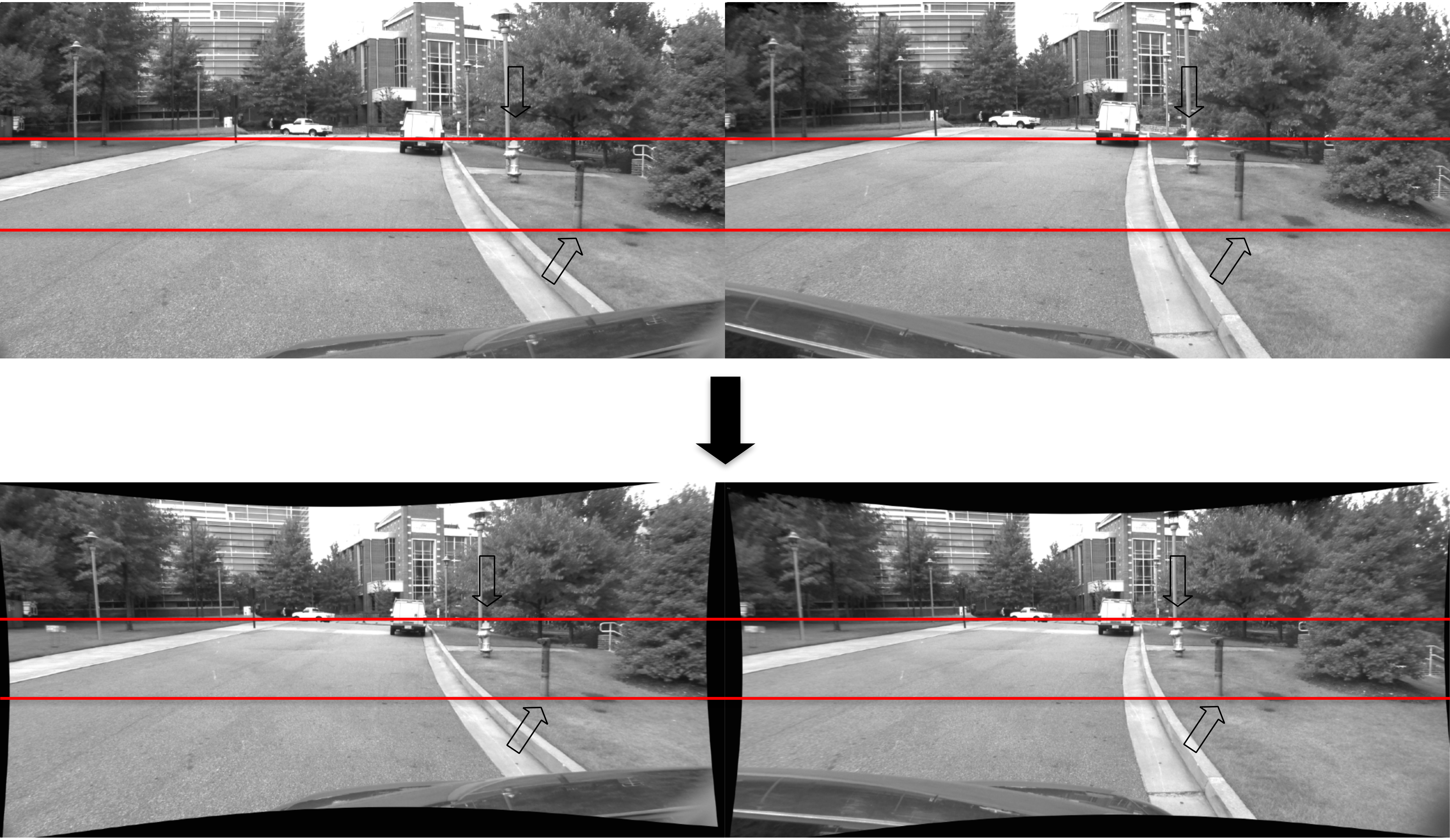






OpenCV: stereoRectify, initUndistortRectifyMap

(Parenthesis on Stereo Matching)

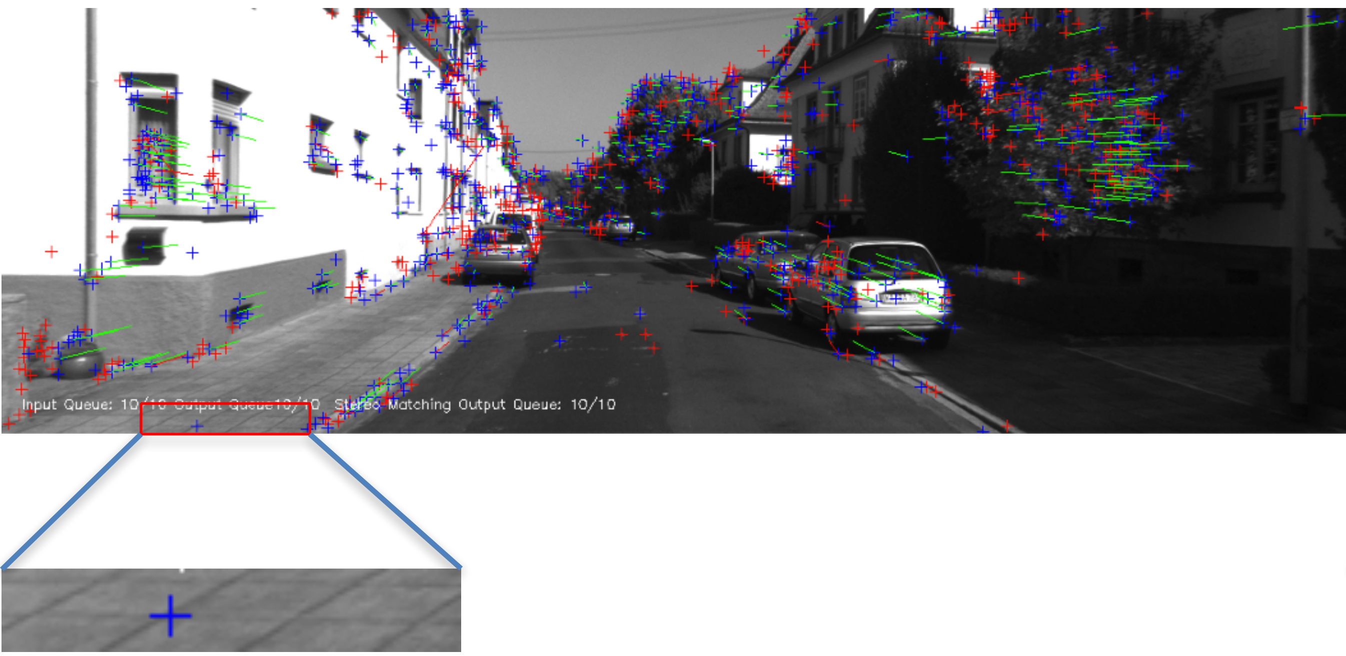


[courtesy of Frank Dellaert]

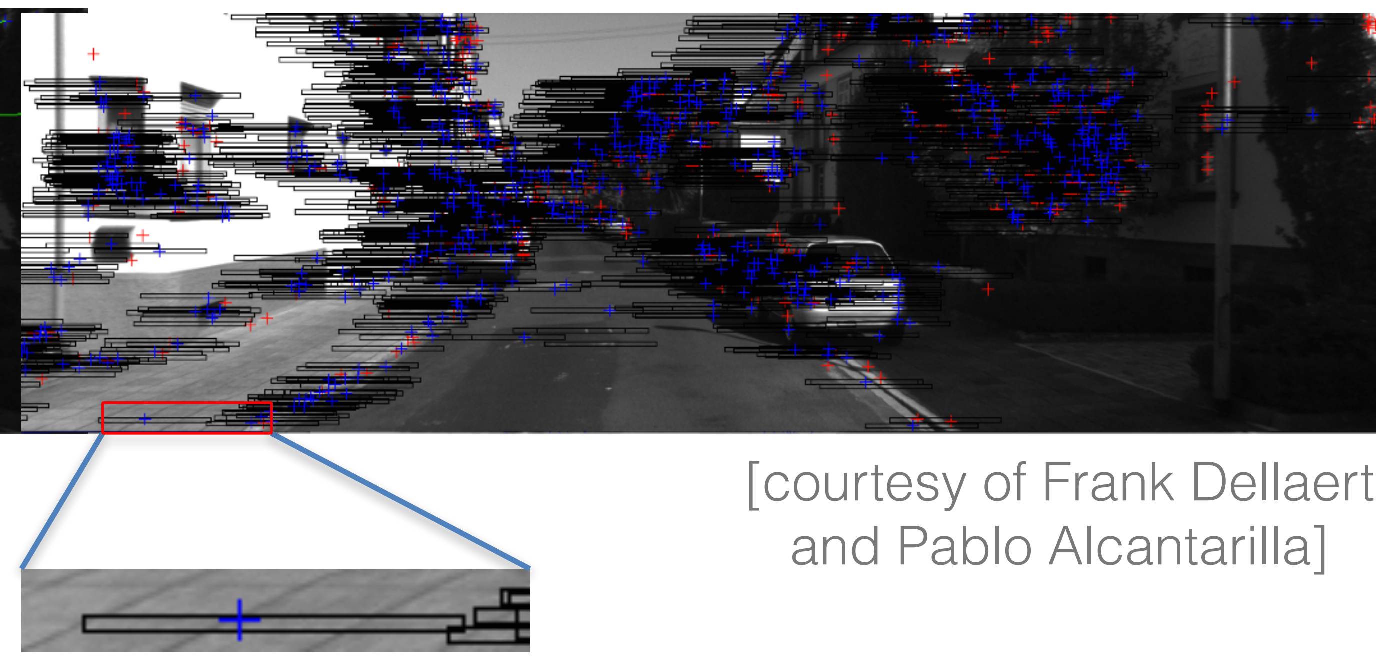
(Parenthesis on Stereo Matching)

After **rectification**, we can restrict search for left-right matches to horizontal lines

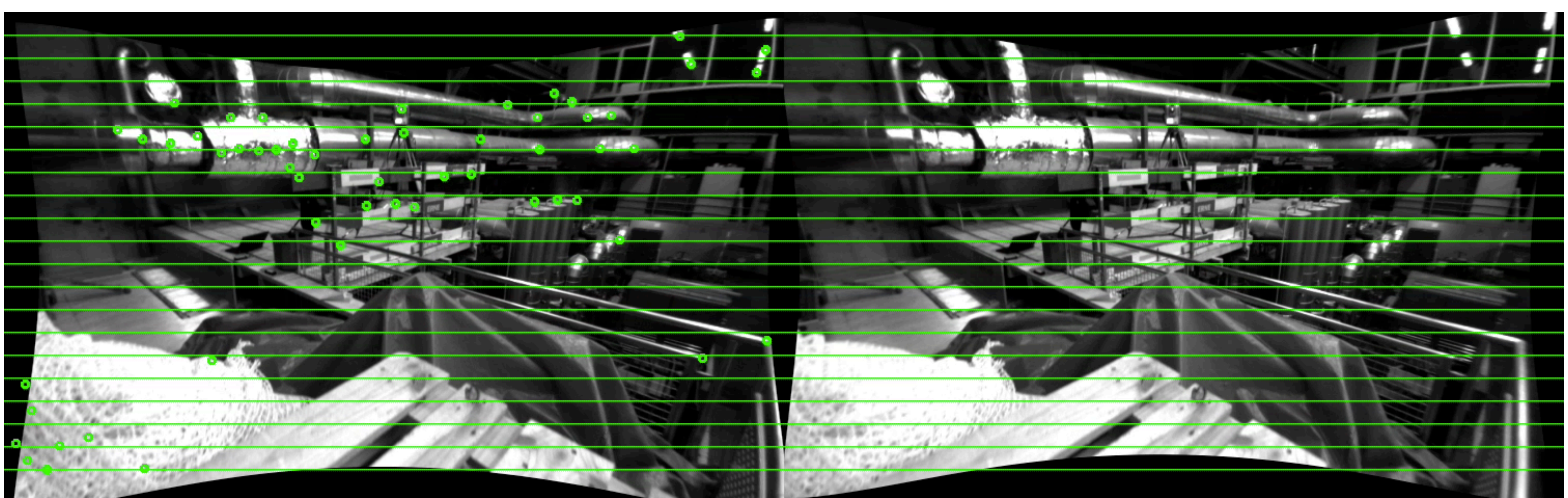
Left image



Right image



[courtesy of Frank Dellaert and Pablo Alcantarilla]



Comparing VO approaches

$$
\begin{aligned}
\boldsymbol{T}_0 &= \mathbf{I}_4 \\
\boldsymbol{T}_1 &= \boldsymbol{T}_0 \boldsymbol{T}_1^0 \\
\boldsymbol{T}_2 &= \boldsymbol{T}_1 \boldsymbol{T}_2^1 = \boldsymbol{T}_0 \boldsymbol{T}_1^0 \boldsymbol{T}_2^1 \\
&\vdots \\
\boldsymbol{T}_t &= \boldsymbol{T}_{t-1} \boldsymbol{T}_t^{t-1} = \boldsymbol{T}_0 \boldsymbol{T}_1^0 \boldsymbol{T}_2^1 \cdots \boldsymbol{T}_t^{t-1}
\end{aligned}
$$

Drift (error accumulation):

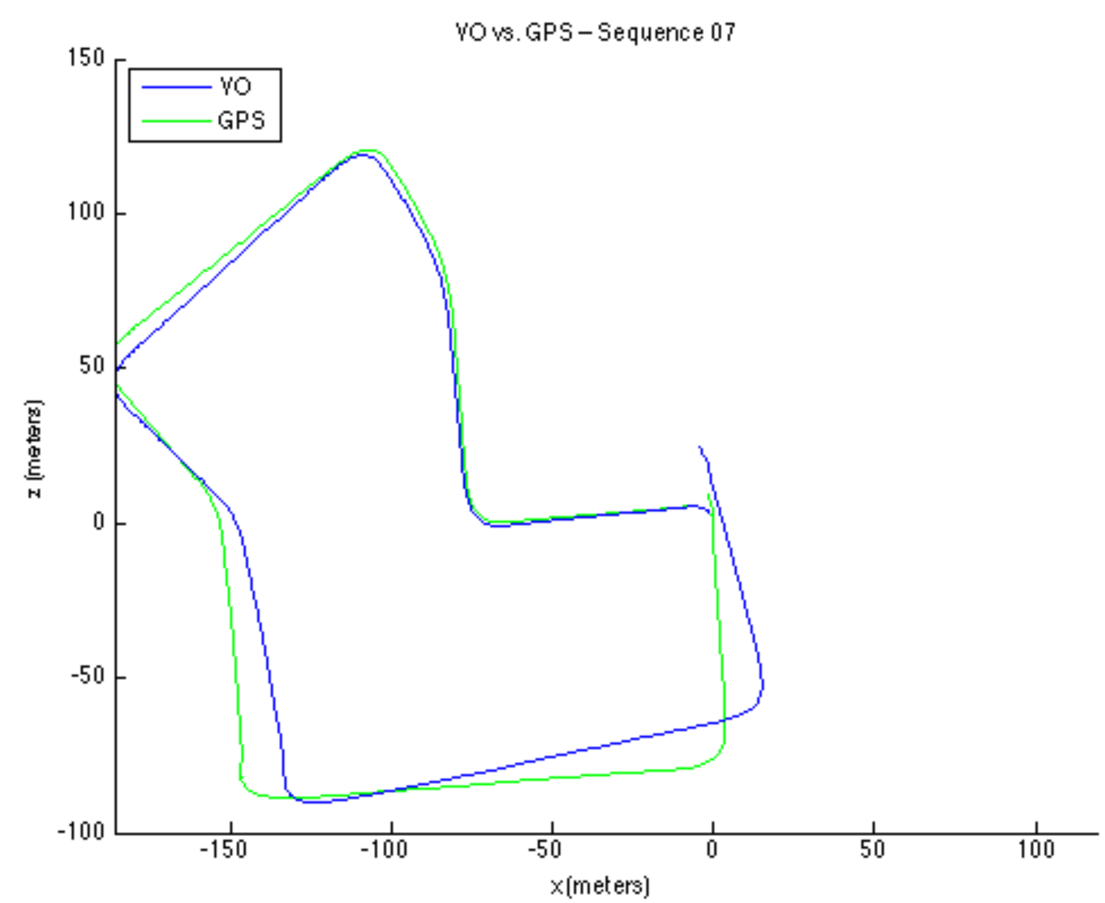
Mono VO:

- 5-point method accurate

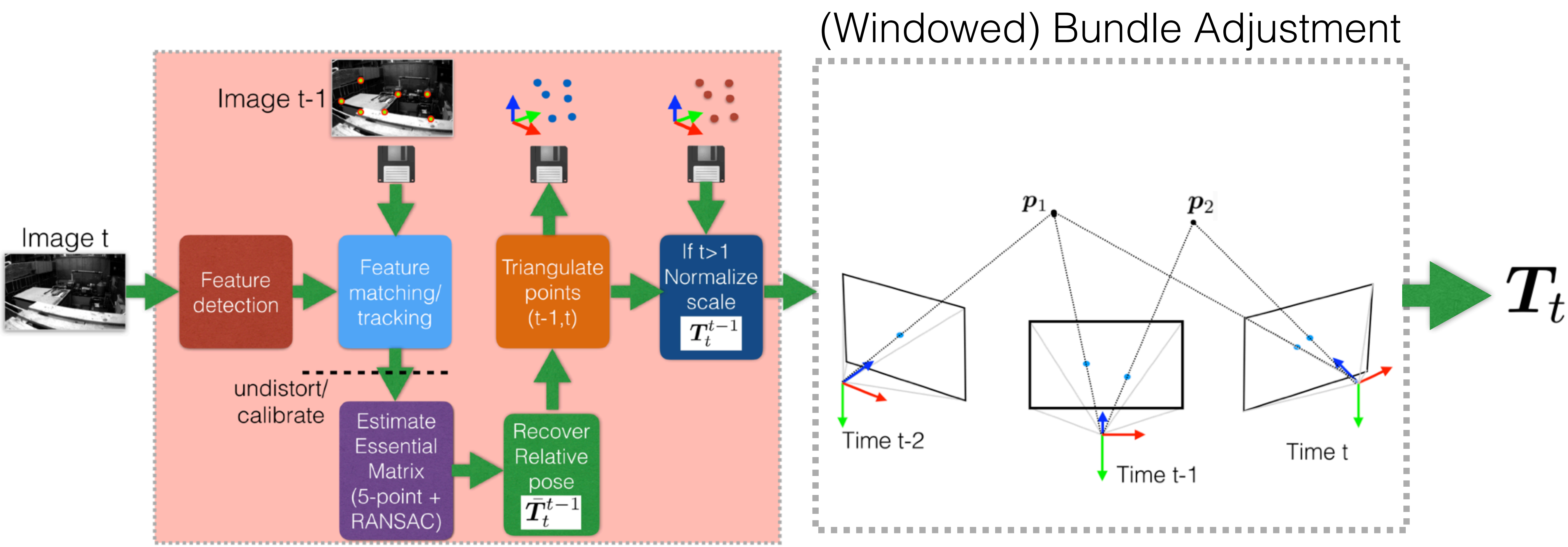
Stereo VO:

- scale

Can we do better?



Refinement: Bundle Adjustment

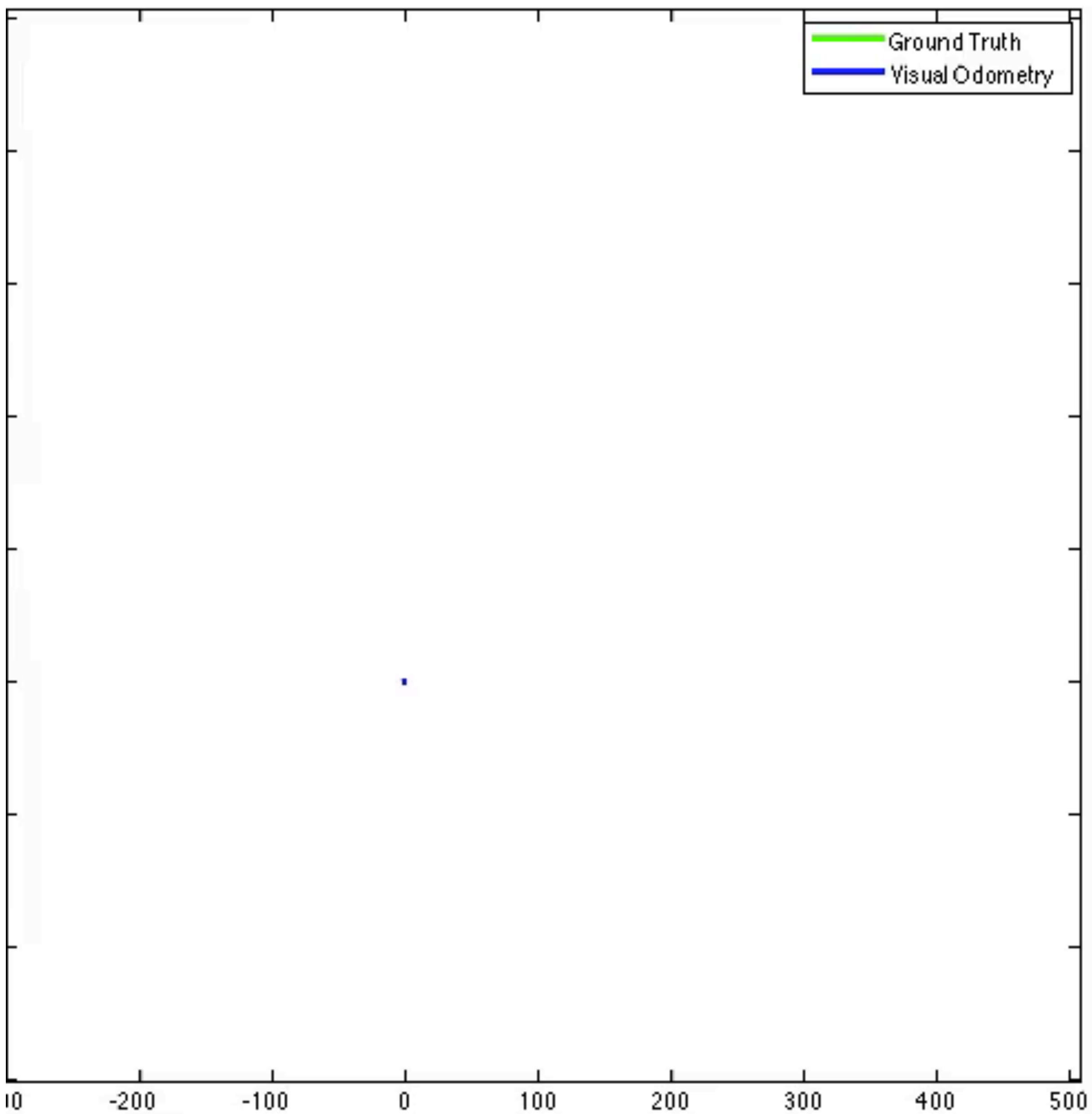


Windowed Bundle Adjustment: optimization of the most recent camera poses and points via non-linear least squares

$$\min_{\substack{\boldsymbol{T}_i, i=1,\dots,N_C \\ \boldsymbol{p}_k, k=1,\dots,N}} \sum_{k=1}^{N} \sum_{i \in \mathcal{C}_k} \|\boldsymbol{x}_{k,i} - \pi(\boldsymbol{T}_i, \boldsymbol{p}_k)\|^2$$

Can be applied to all the pipelines discussed today

Stereo VO example (2)



Typical drifts: 0.1% to 2% of trajectory travelled

[courtesy of Frank Dellaert]

Challenges for VO (1/3): Illumination and Features

Feature detection,
tracking,
matching …

Challenges for VO (2/3): Dynamic Scenes

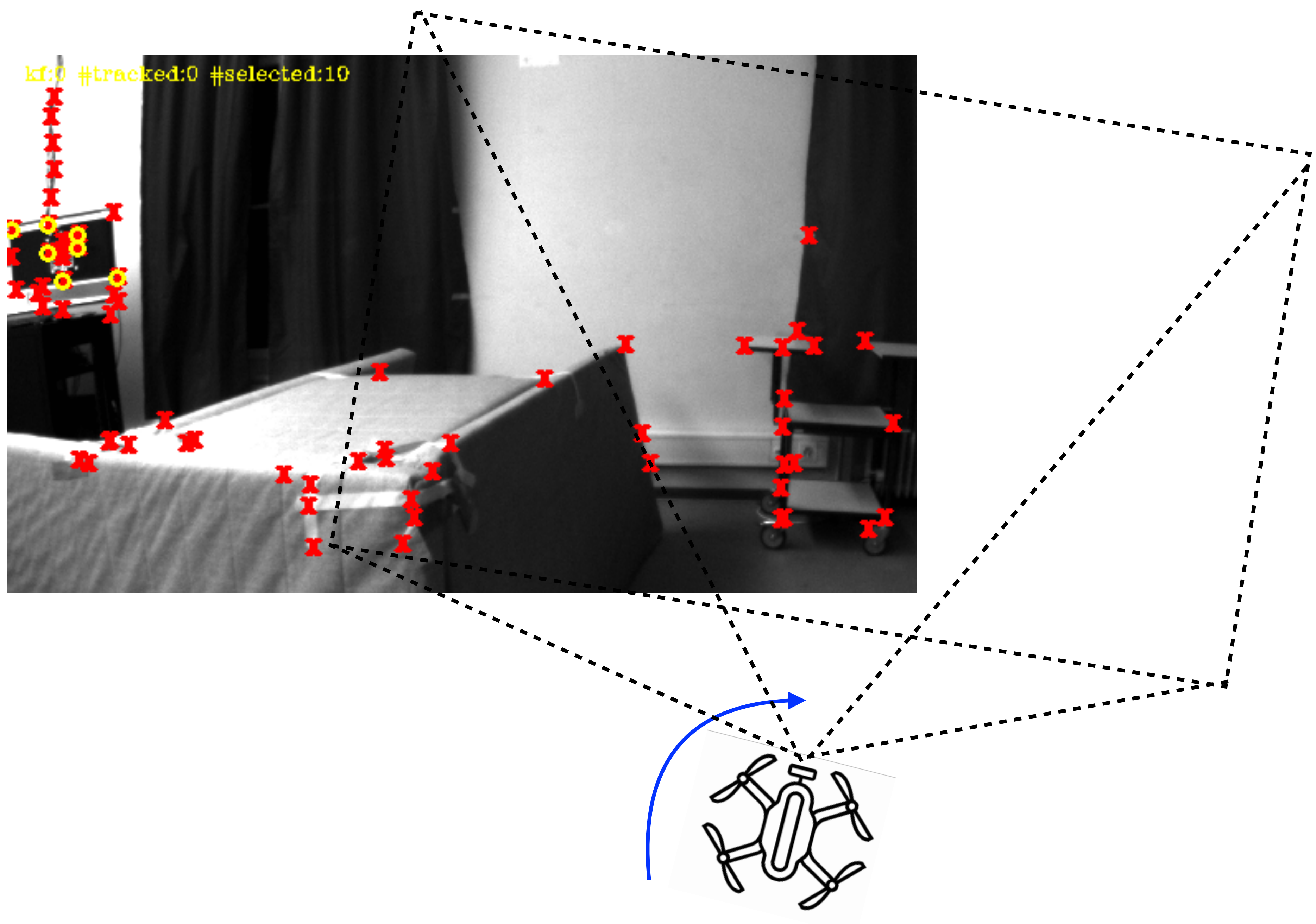
- Dynamic, crowded scenes present a real challenge
- Can't rely on RANSAC to always recover the correct inliers
- Example: Large van "steals" inlier set in passing

Inliers Outliers

[courtesy of Frank Dellaert]

Challenges for VO (3/3): Fast Motion

Need good overlap between consecutive images

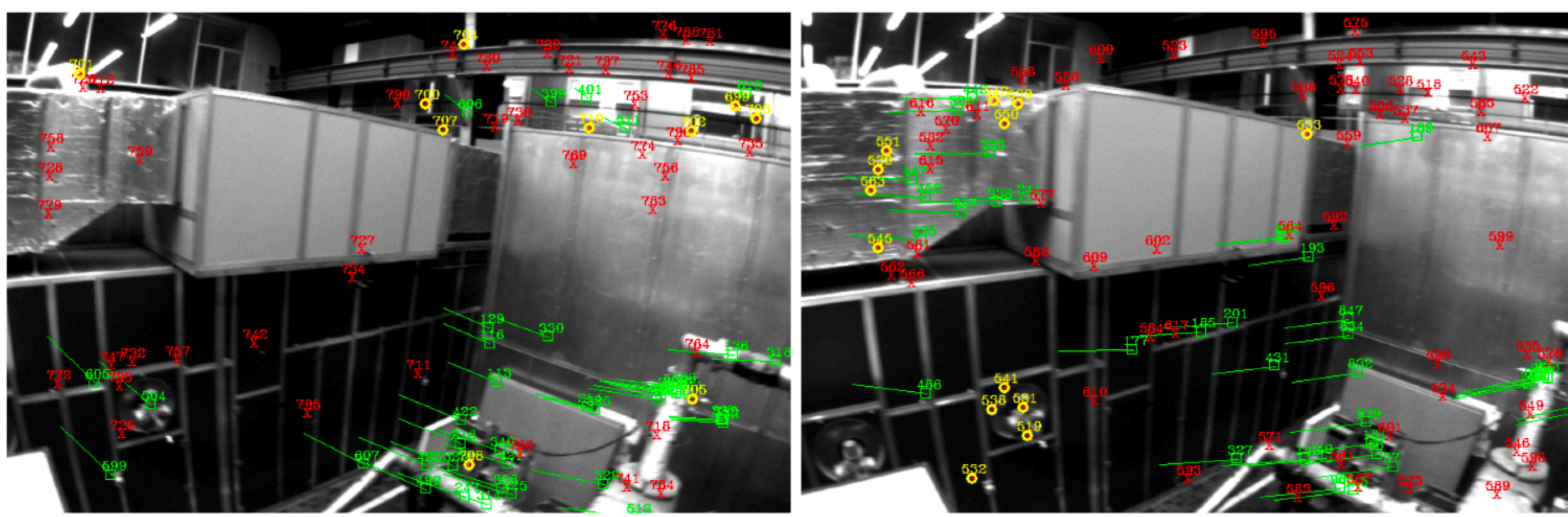


Robot speed, camera framerate, …

VO Tricks (1/2): Feature Distribution

Feature Binning:

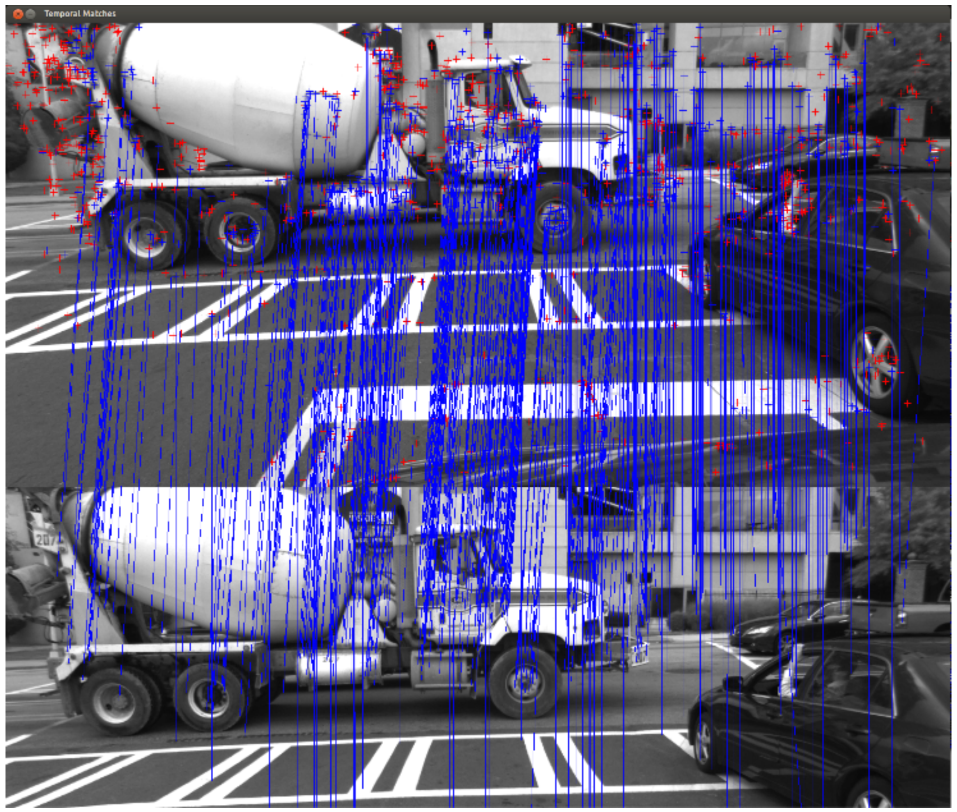
Attention & Anticipation:



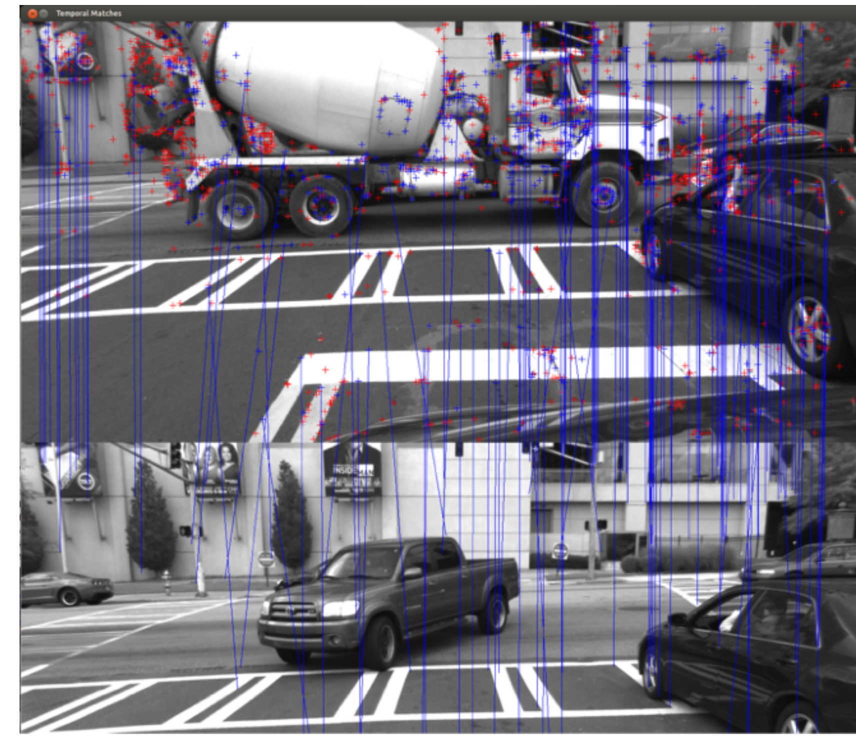
select features depending on motion of the robot

VO Tricks (2/2): Domain Knowledge and Keyframes

- Stereo VO Example: Cross-traffic while waiting to turn left at light

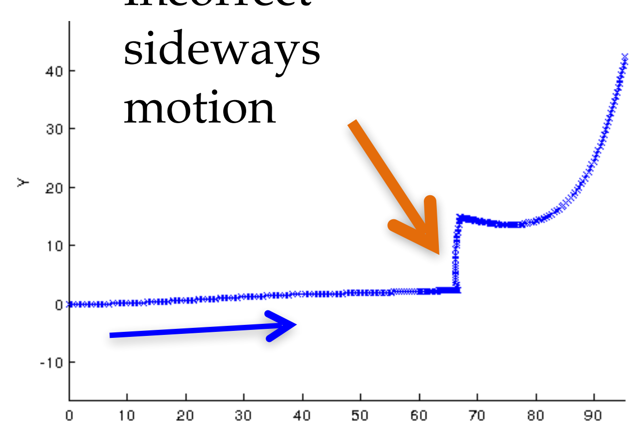


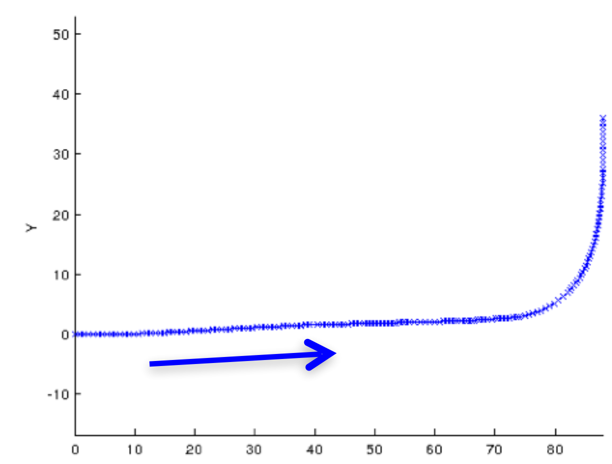
Only accept incremental pose if:
- Translation > 0.5m
- Dominant direction is forward



Without keyframing

Incorrect sideways motion

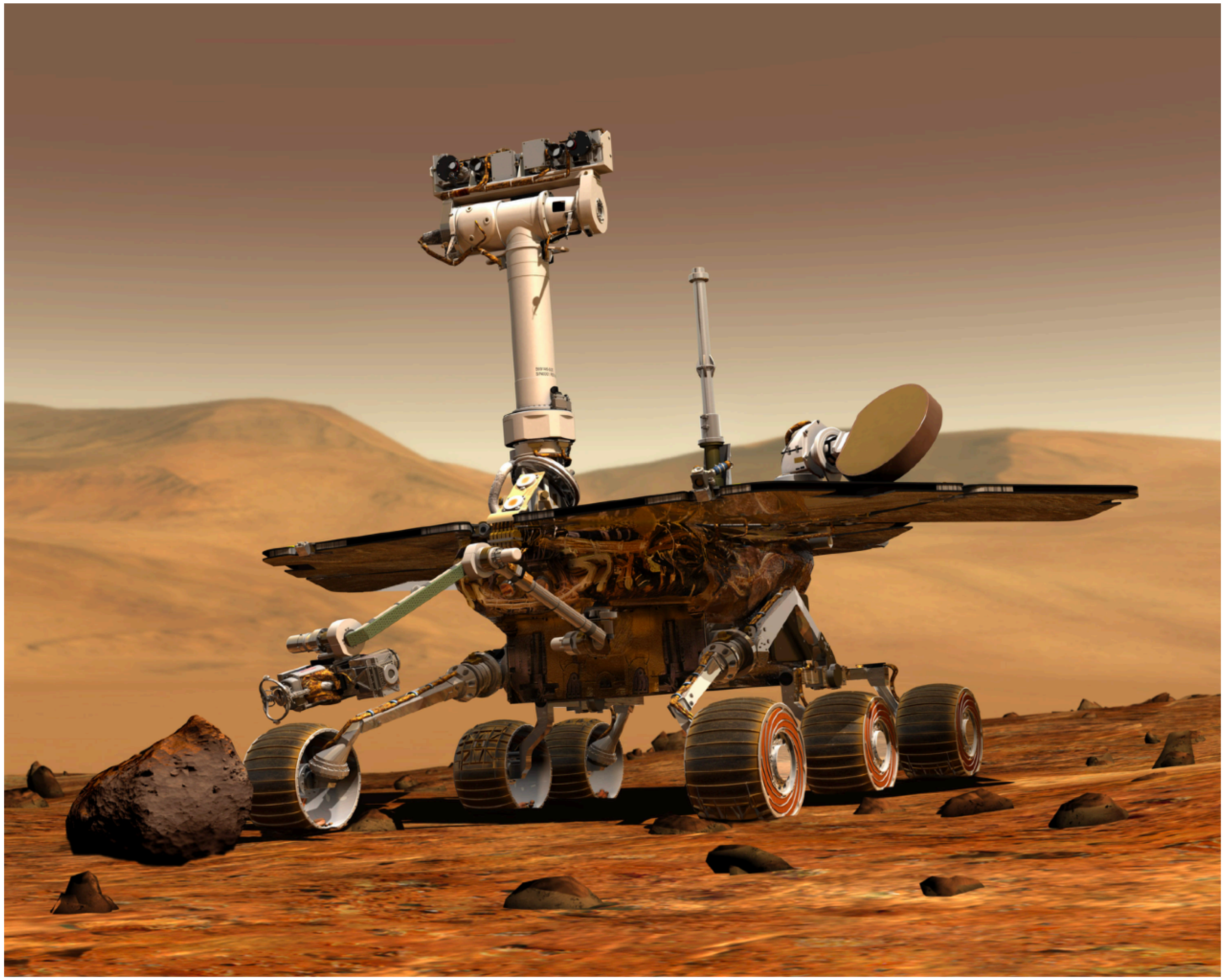




With keyframing

[courtesy of Frank Dellaert]

Stereo VO example (1)



Source: public domain. Courtesy of NASA/JPL/Cornell University.

Spirit and Opportunity Mars rovers:

- stereo VO
- 20-MHz CPU
- up to three minutes for 2-view VO
- Drift ~0.5% of trajectory travelled

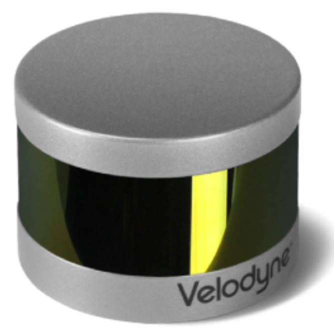
Earlier implementation: Moravec's PhD Thesis (1980)

Beyond VO

How to get scale and improve robustness?

add more sensors!

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

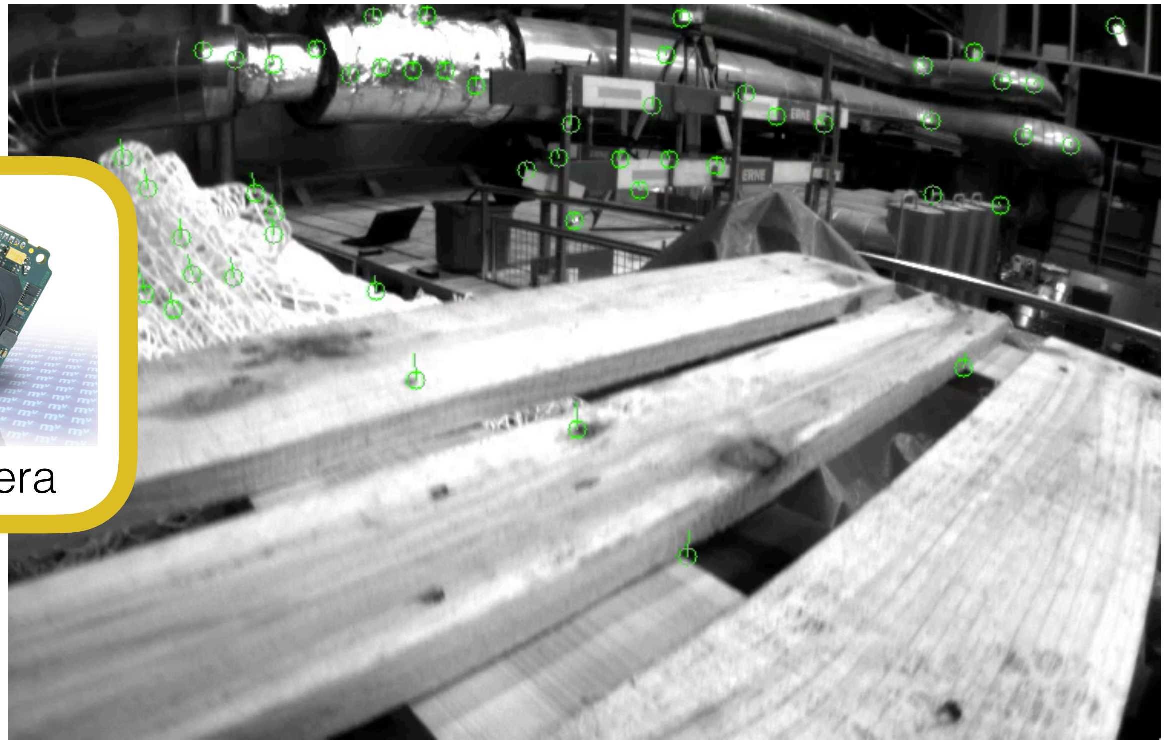


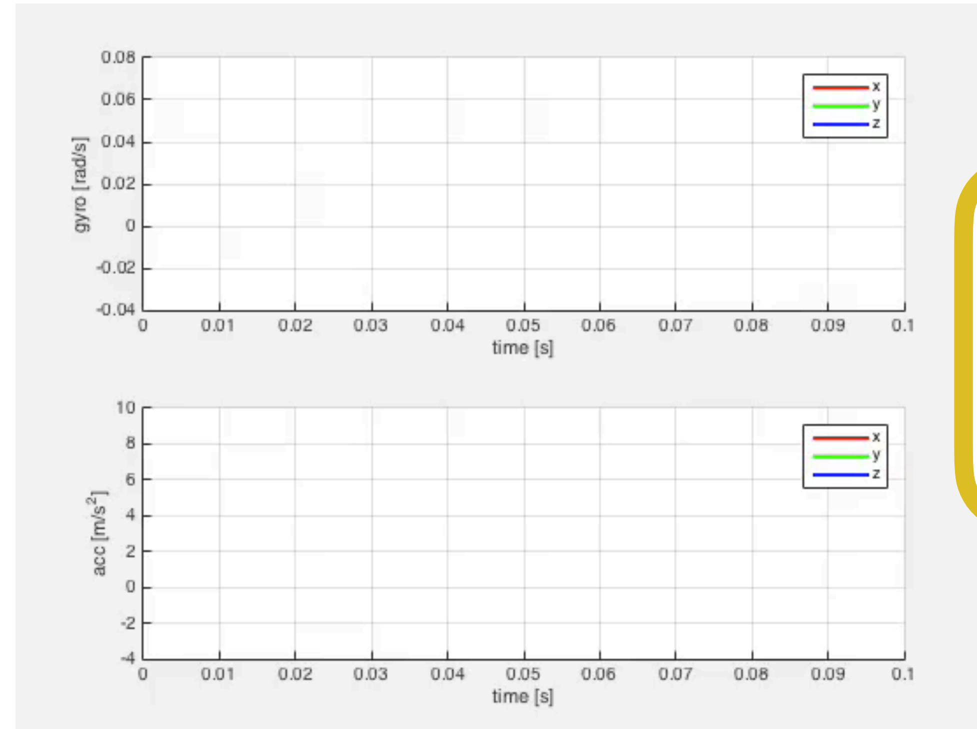


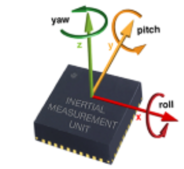
830g	160g	4g	3g
8 W	2.5 W	0.3W	~1 W

Visual-Inertial Navigation (VIN)

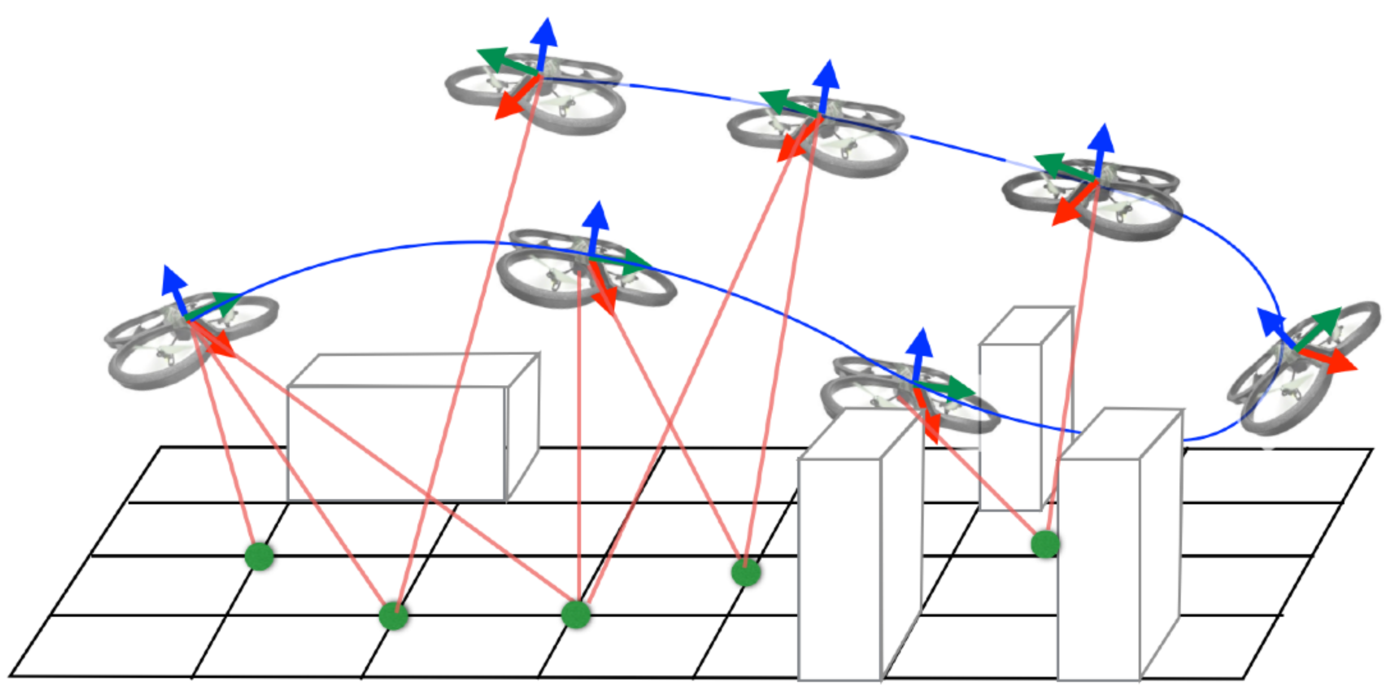
camera



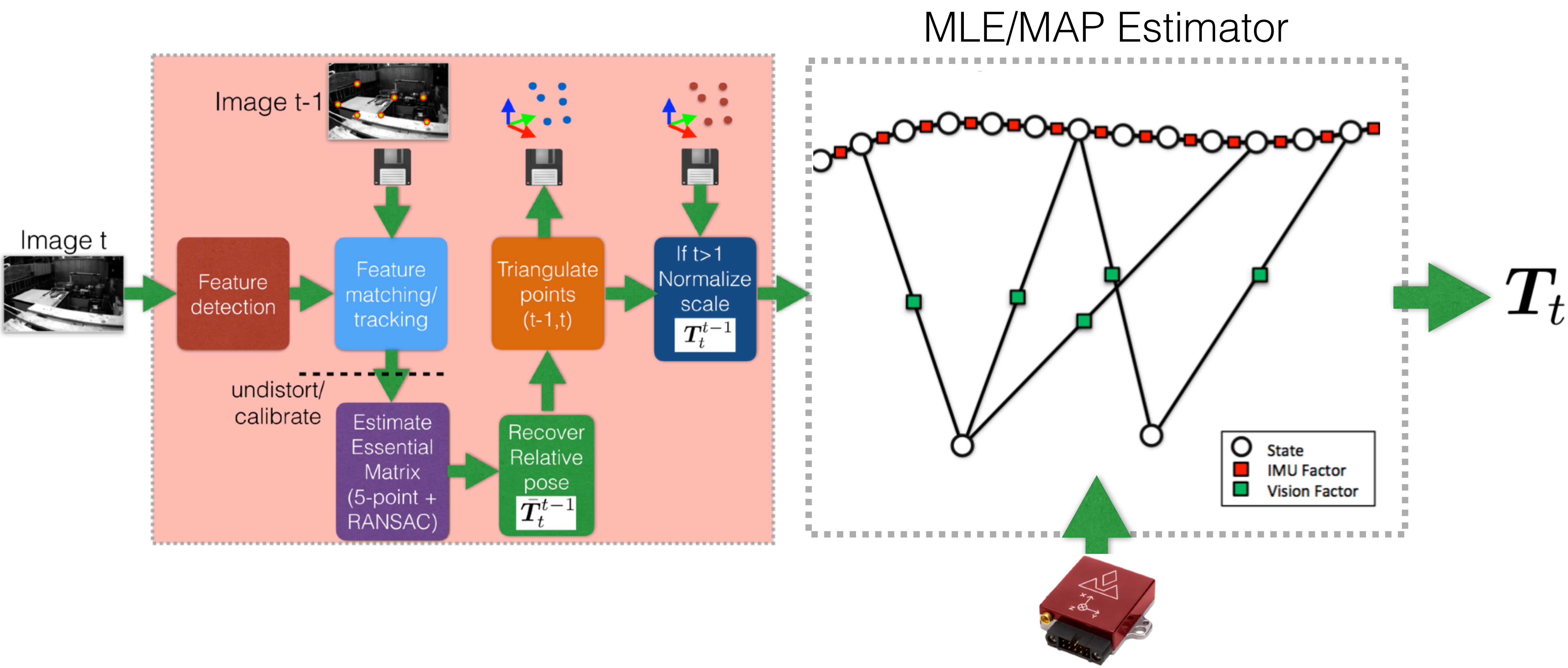




Inertial Measurement Unit (IMU)



Visual-Inertial Odometry



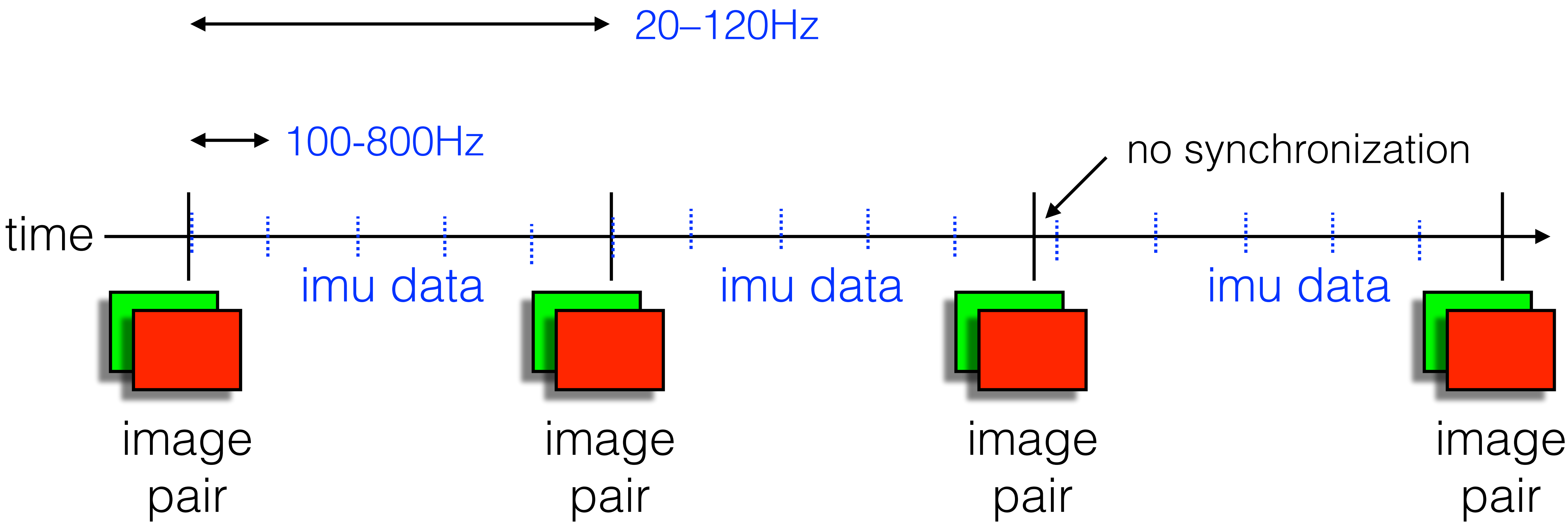
Camera factors

Imu factors

$$\min_{\substack{\boldsymbol{T}_i, i=1,\dots,N_C \\ \boldsymbol{p}_k, k=1,\dots,N}} \sum_{k=1}^{N} \sum_{i\in\mathcal{C}_k} \|\boldsymbol{x}_{k,i} - \pi(\boldsymbol{T}_i, \boldsymbol{p}_k)\|^2 + \sum_{i=1,\dots,N_C-1} \|r_{\mathrm{imu}}(\boldsymbol{T}_i, \boldsymbol{T}_{i+1}, \boldsymbol{v}_i, \boldsymbol{v}_{i+1}, \boldsymbol{b}_i, \boldsymbol{b}_{i+1})\|^2$$

Need to include velocities and IMU biases in the state …

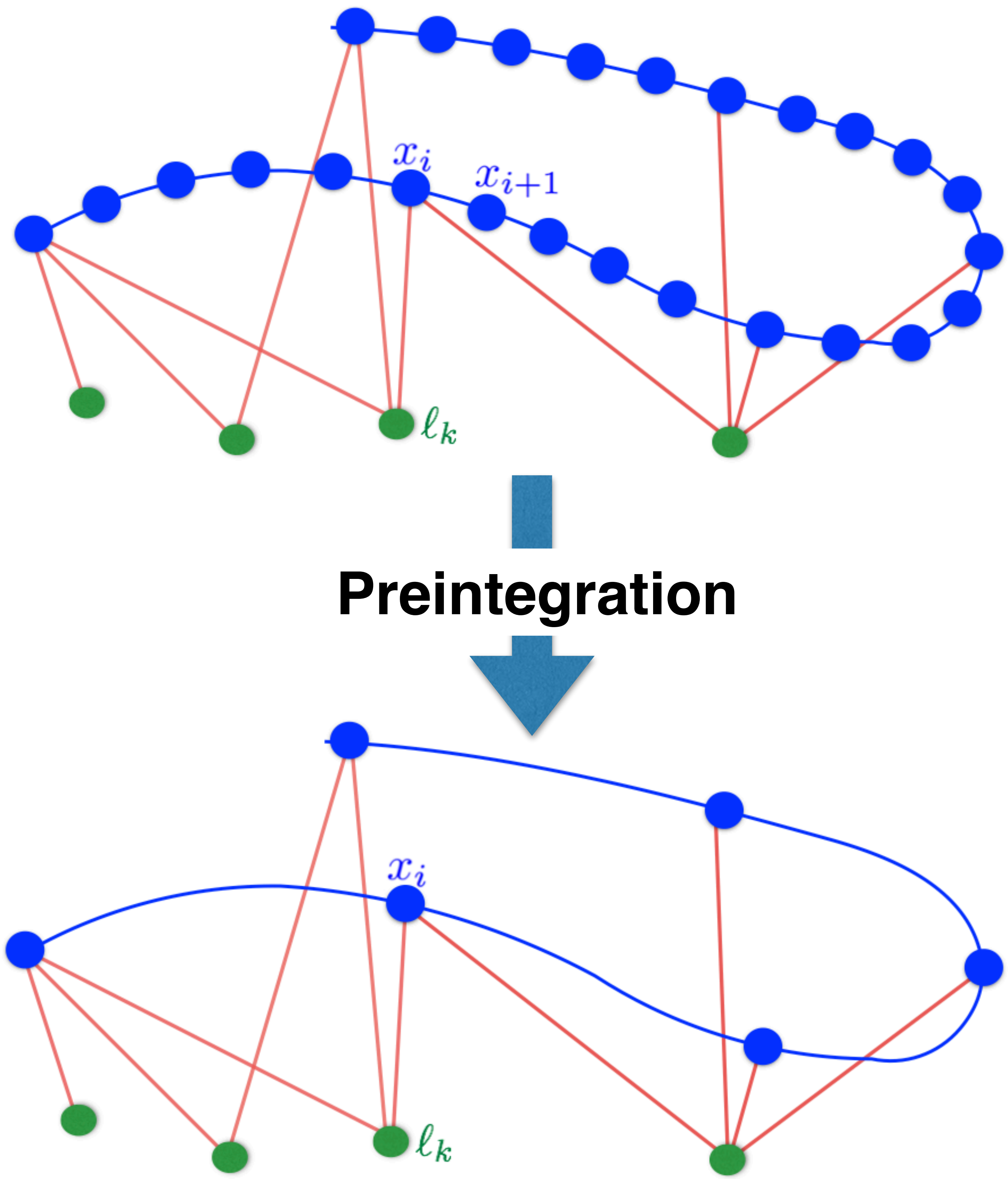
Visual-Inertial Odometry



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

Pre-integration

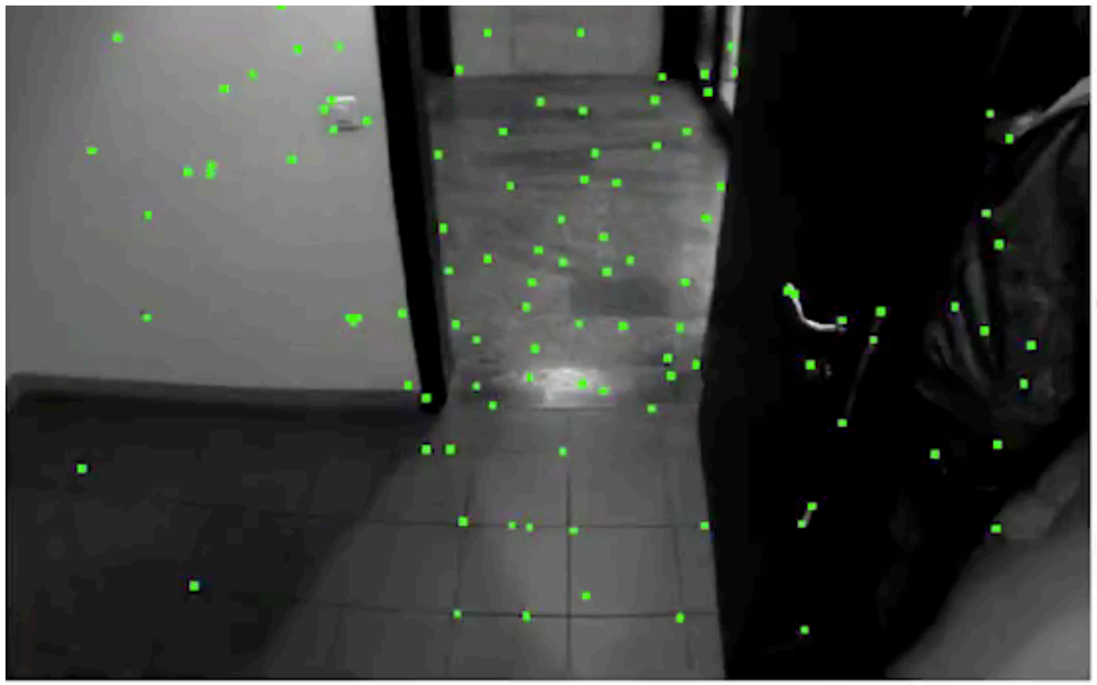


After 10 seconds, original problem has ~10^4 states

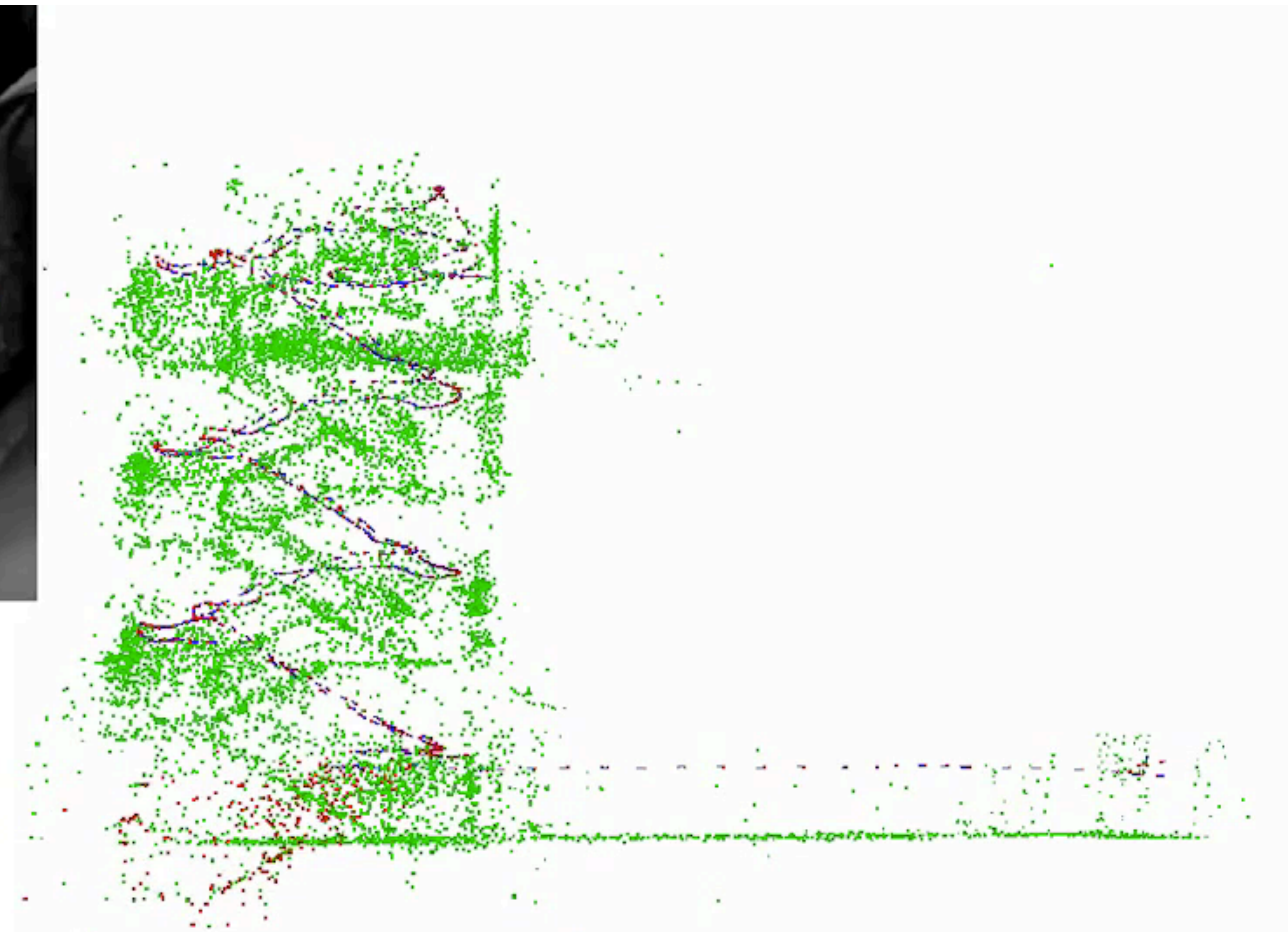
After 10 seconds, preintegrated problem has ~10^2 states

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

Visual-Inertial Odometry



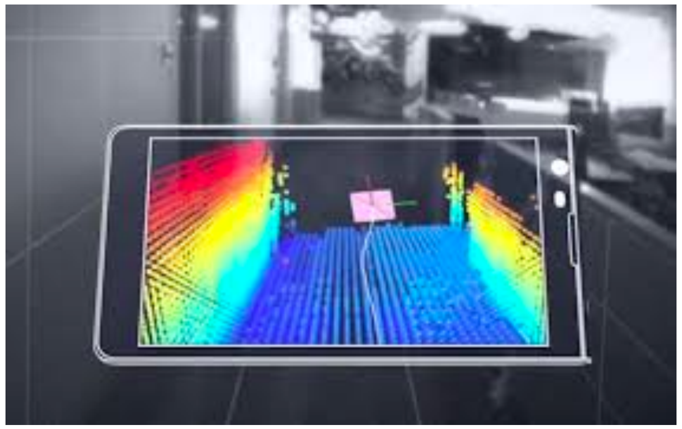
Hand-held sensor



Implemented in GTSAM (ImuFactor)

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

Recent Implementations / Products



Project Tango

2014

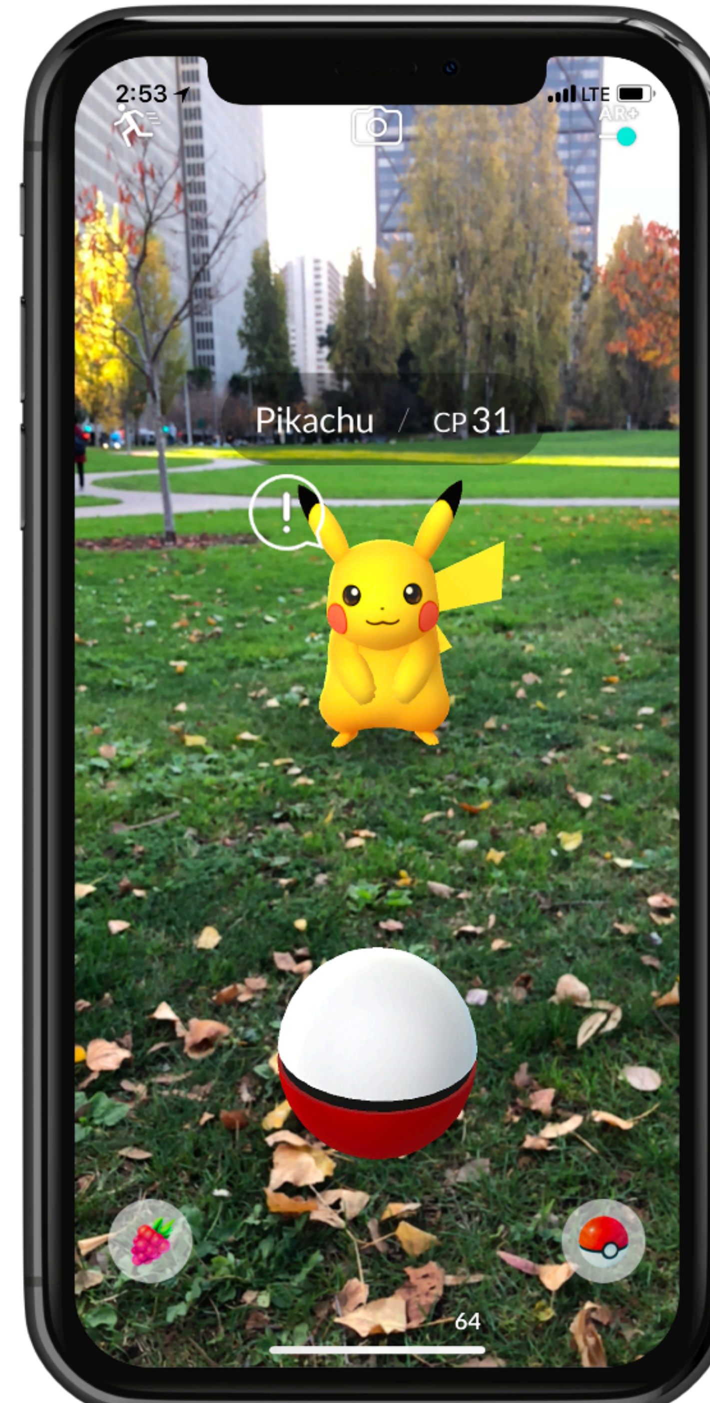
Reinvented as ARCore in 2017

Oculus Rift

Announced in 2012. Acquired by Facebook in 2014

Navion Chip
2017
(http://navion.mit.edu/)

Pokemon Go

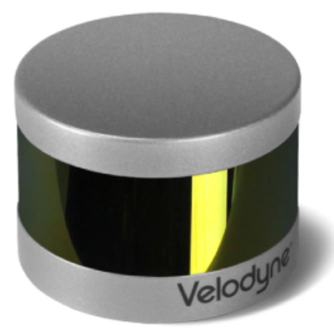


Beyond VO

How to get scale and improve robustness?

add more sensors!

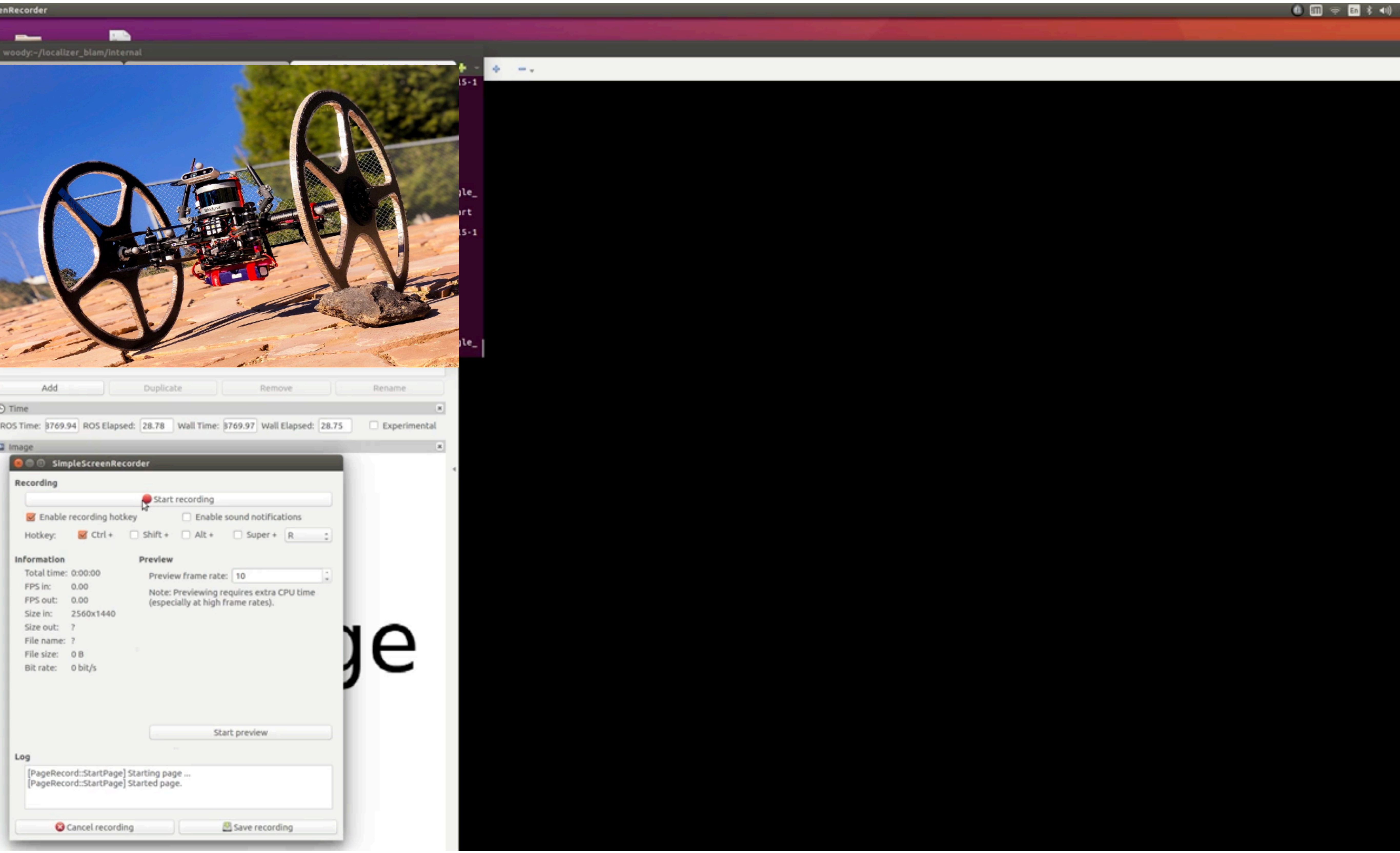
- wheel odometry
- GPS
- Lidar
- Inertial Measurement Unit (IMU)





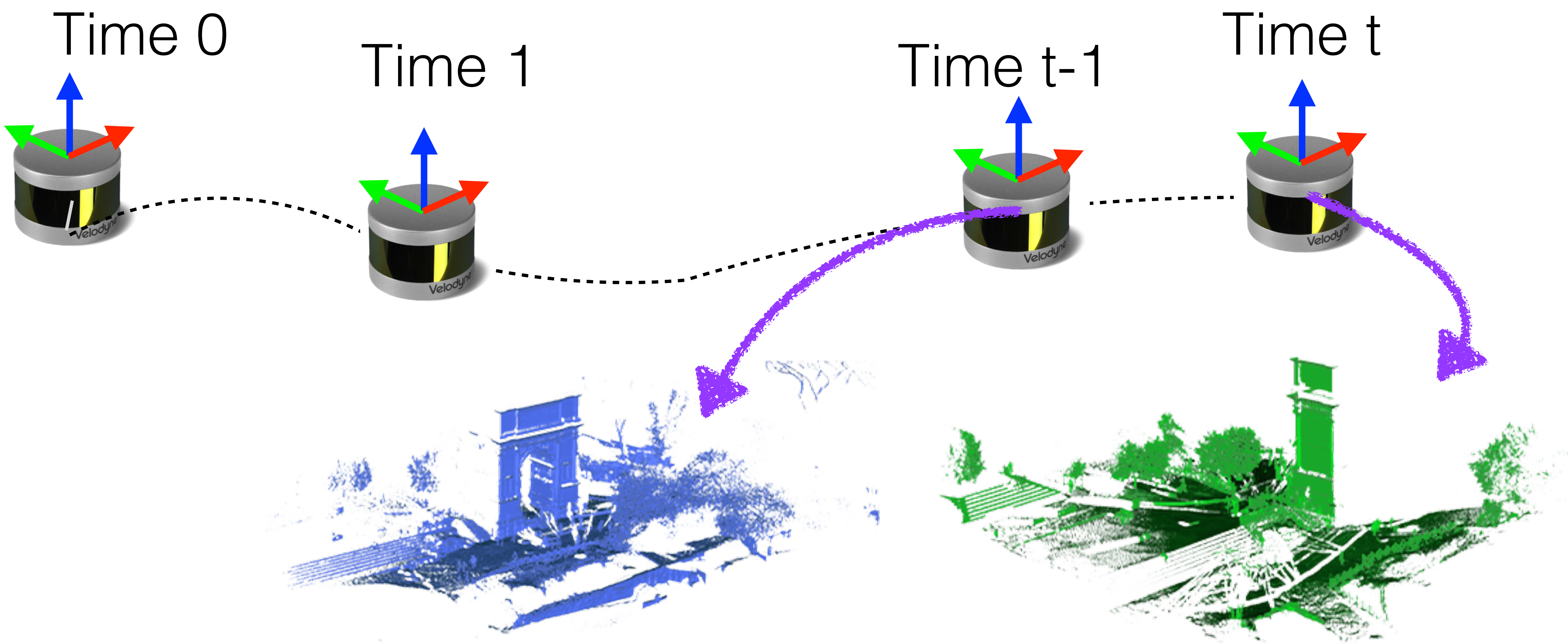
830g	160g	4g	3g
8 W	2.5 W	0.3W	~1 W

Lidar Odometry & Lidar SLAM

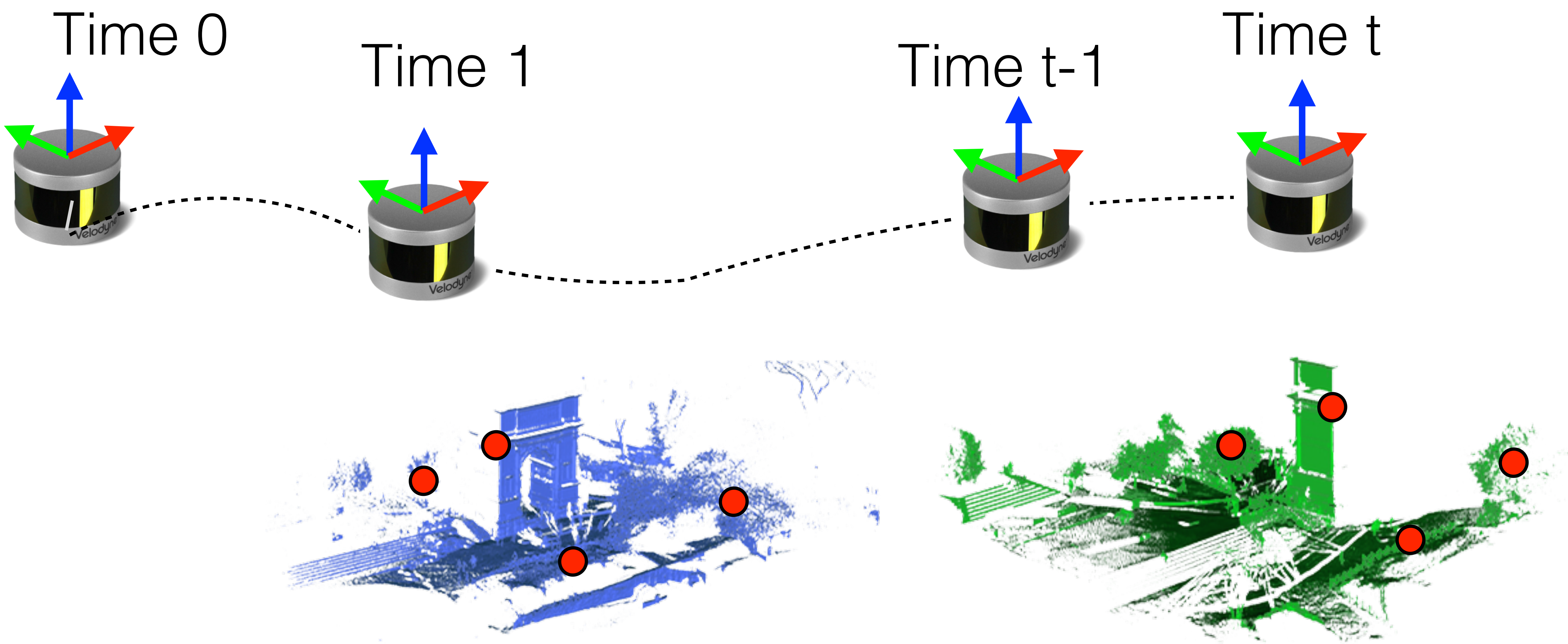


DARPA Subterranean Challenge, in collaboration with JPL

Feature-based Lidar Odometry



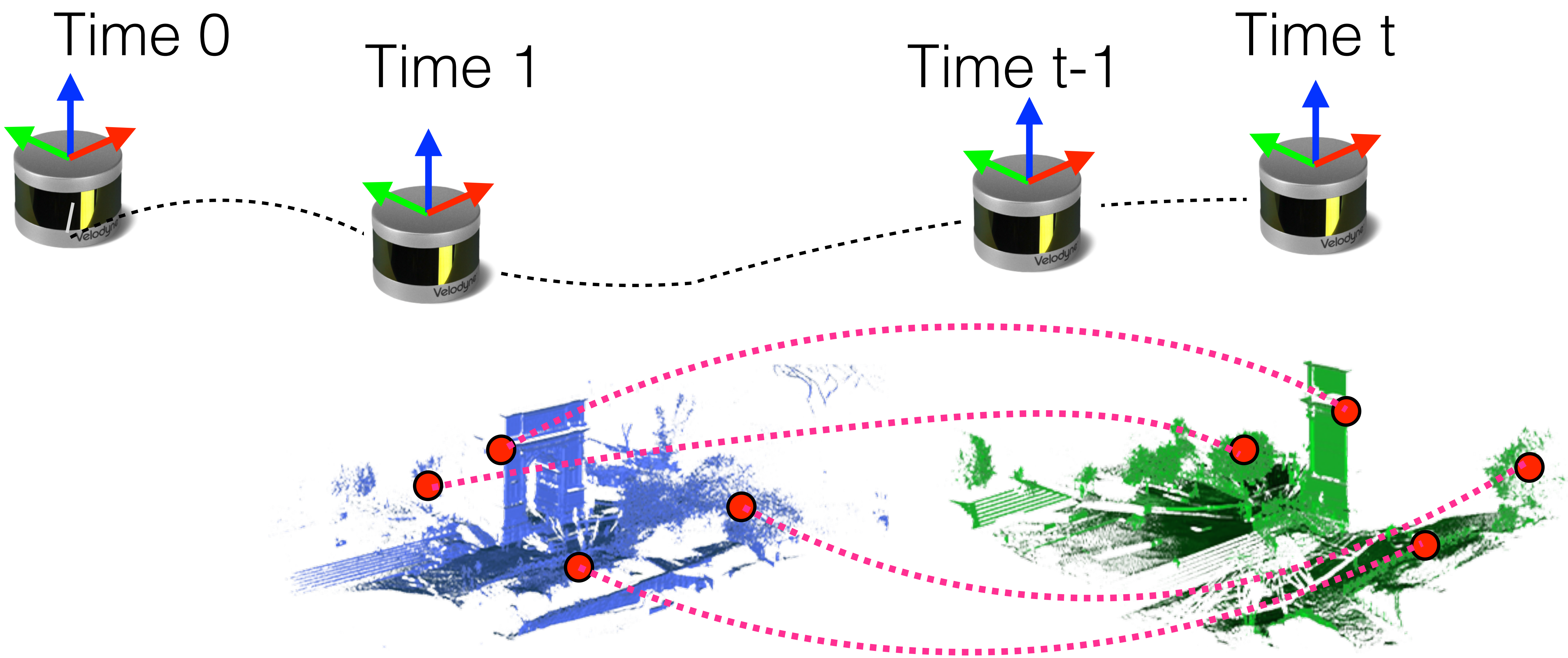
Feature-based Lidar Odometry



Registration: compute relative pose between scans:

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

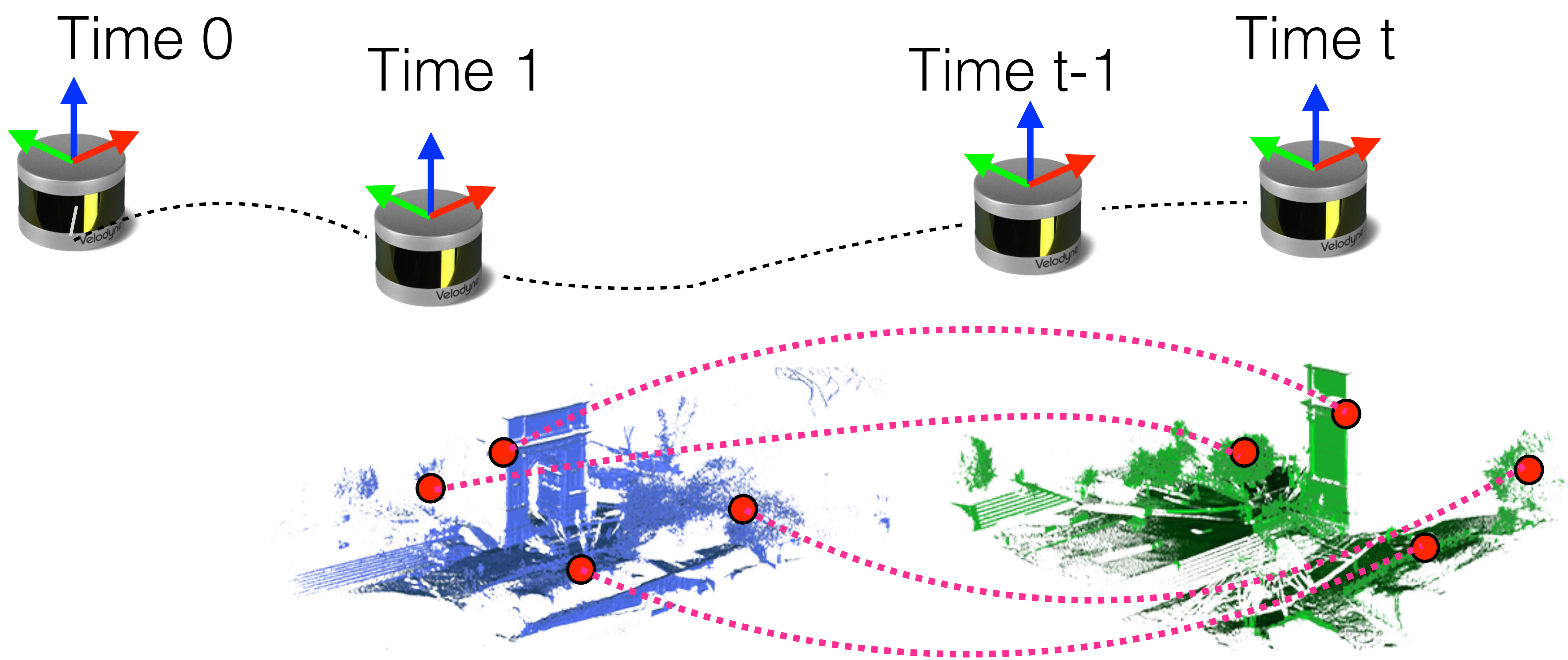
Feature-based Lidar Odometry

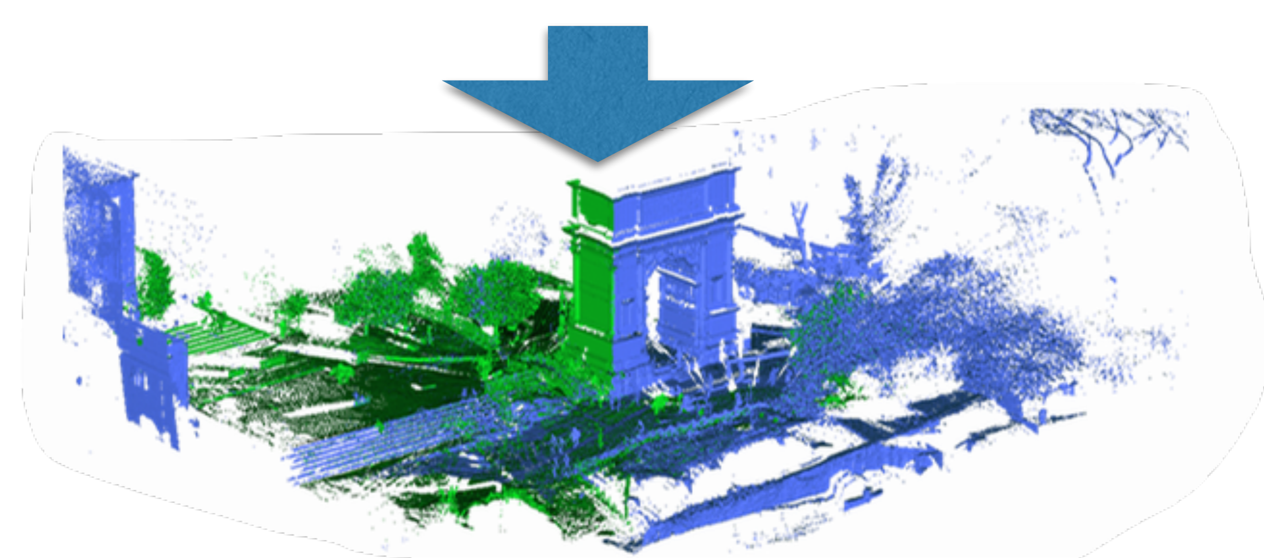


Registration: compute relative pose between scans:

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

Feature-based Lidar Odometry

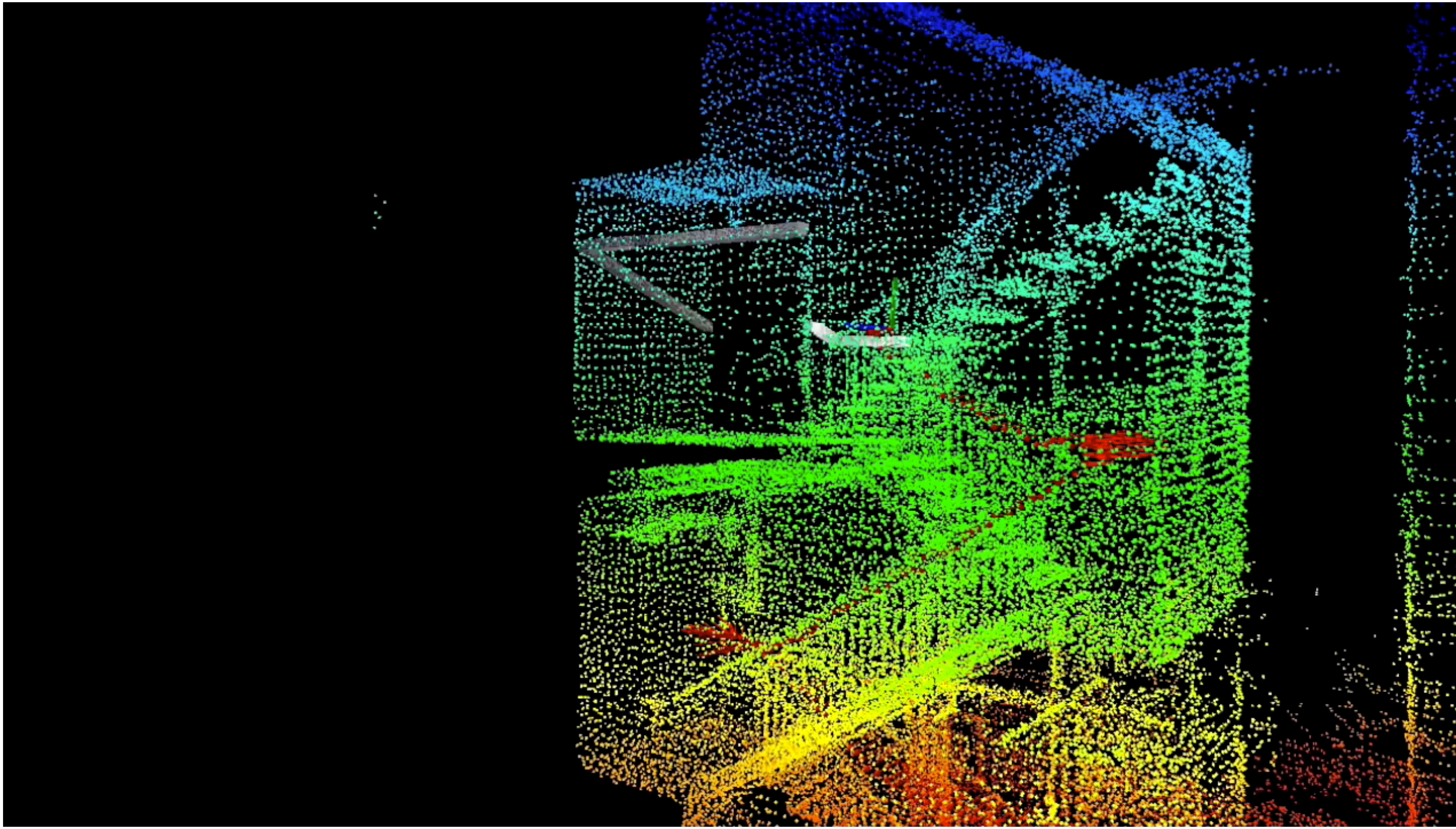




Registration: compute relative pose between scans:

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

Feature-based Lidar Odometry

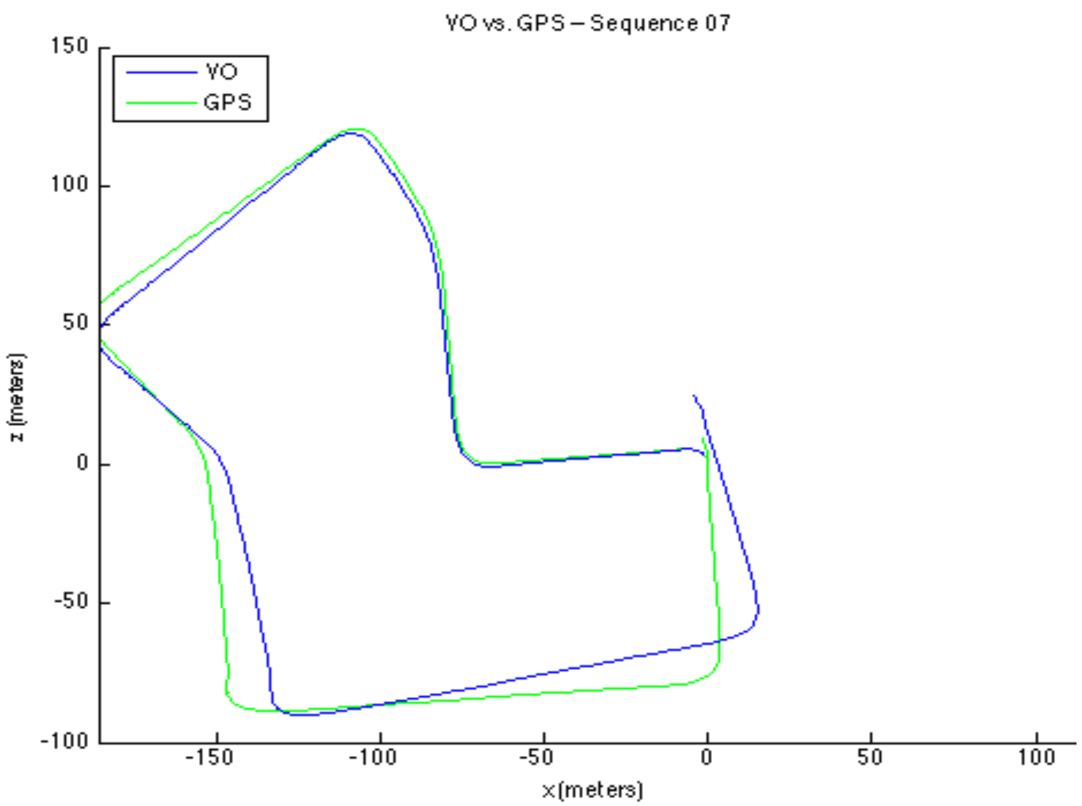


[Zhang and Singh: LOAM: Lidar Odometry and Mapping in Real-time, 2014]

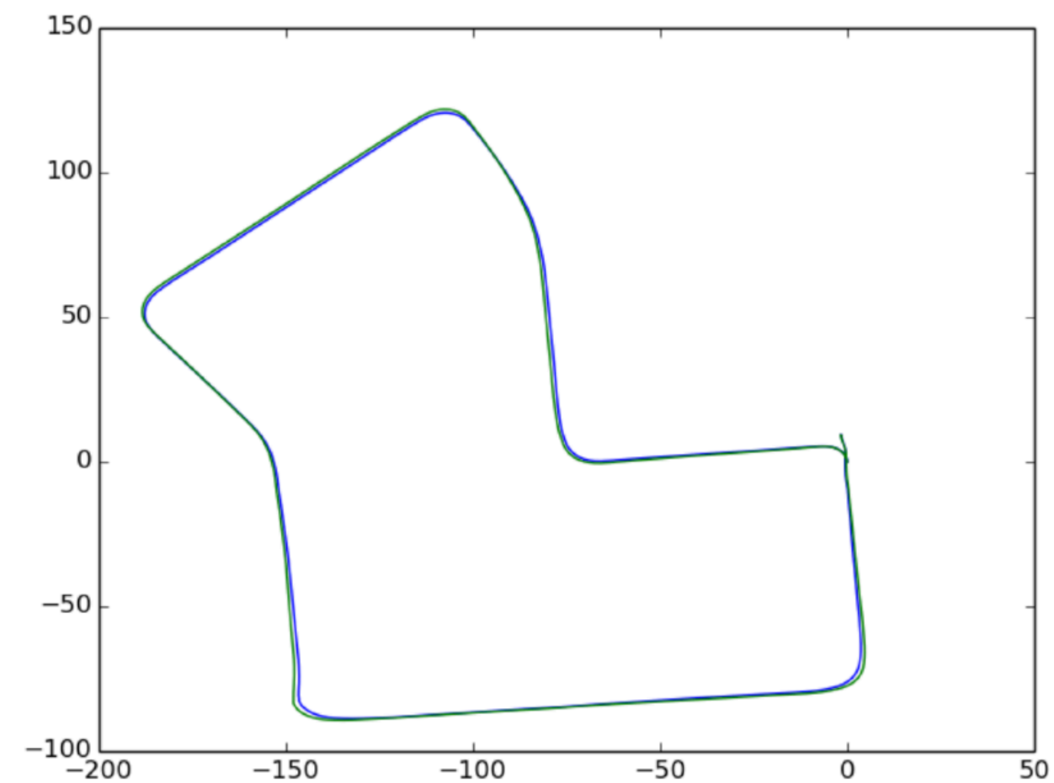
Other approaches: based on Iterative Closest Point (ICP)

Removing Drift via Loop Closure

Visual(-inertial) odometry



SLAM



SLAM requires:

- place recognition (loop closure detection)
- Re-detecting landmarks (e.g., objects)

Next lecture!

Need for loop closure

ORB-SLAM

Raúl Mur-Artal, J. M. M. Montiel and Juan D. Tardós

{raulmur, josemari, tardos} @unizar.es

Instituto Universitario de Investigación en Ingeniería de Aragón Universidad Zaragoza

Universidad Zaragoza 1542