Maplab: An Open Framework for Research in Visual-Inertial Mapping and Localization

Thomas Schneider , Marcin Dymczyk , Marius Fehr , Kevin Egger , Simon Lynen , Igor Gilitschenski , and Roland Siegwart

Autonomous Systems Lab

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

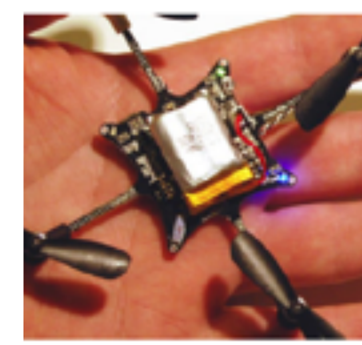
RA-L'18

Background

Visual Inertial Mapping and Localization has a wide range of application scenarios.

- Low-cost robotics (e.g. a mobile robot with a cheap camera)
- Agile robotics (e.g. drones)
- Smartphones
- Wearables
- AR/VR: inside-out tracking, gaming

Availability of precise, drift-free pose estimates system both outdoors and indoors has become a vital requirement.



Background

Requirements:

- collect data using **heterogeneous agents**
- build maps of **larger scale**
- cover **various visual appearance** conditions
- **maintain maps over a long time** horizon

However

Most openly available frameworks for VO and VIO focus on single-session case.

There's no system able to:

- be easily deployed with **multi-session mapping** supported
- provide **ready-to-use algorithm** and **state-of-the-art performance**
- offer **high flexibility** necessary for research and development
- **manipulate/merge/reuse** previous maps

maplab An open visual-inertial mapping framework

maplab is a research-oriented visual-inertial mapping and localization framework

- Create and localize from visual inertial maps
- Provide map maintenance and processing capabilities

Offers a set of tools:

- Multi-session map merging
- Map sparsification
- Loop closing
- Dense reconstruction
- Visualization
- …

maplab

Maplab An open visual-inertial mapping framework

Components:

ROVIOLI

frontend for online map building and localization

maplab

console for offline map processing

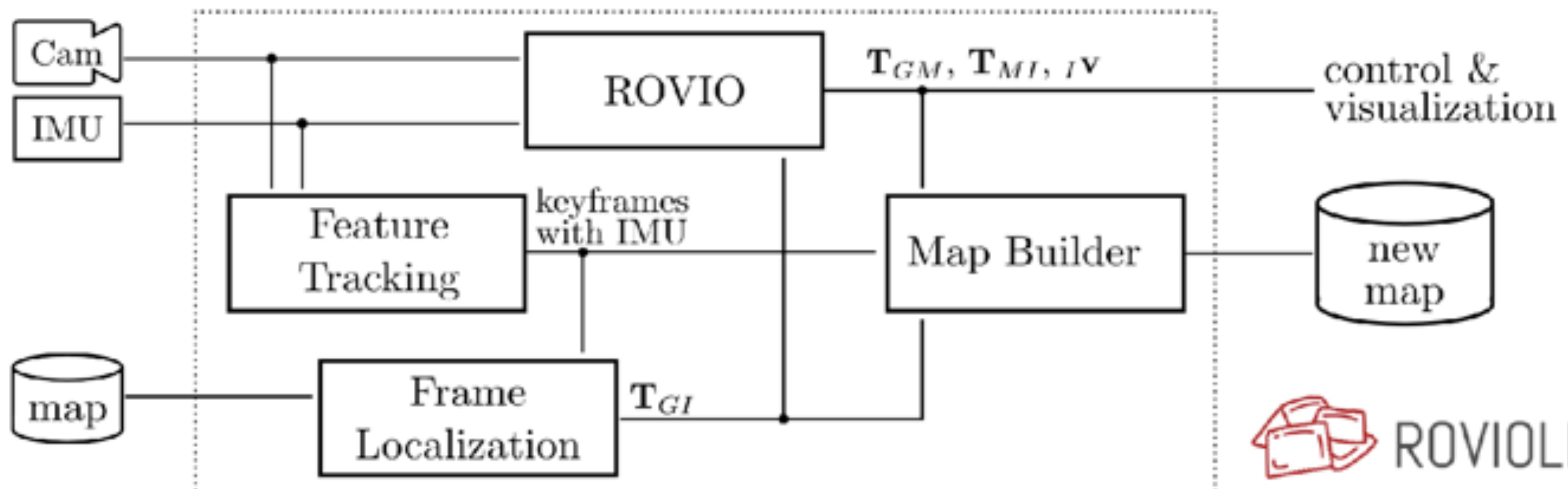
Features:

- Build with ROS build system catkin
- Written in C++ 11
- Eigen/Ceres
- ROS interface to input and output
- RViz as 3d visualization

maplab

ROVIOLI ROBust Visual Inertial Odometry with Localization Integration

ROVIOLI is a visual-inertial mapping front-end for maplab



ROVIO: Estimates the current global transformation, pose and velocity

Feature Tracking: Feature correspondences between frames are established by matching descriptors from frame to frame.

Frame Localization: Process feature points and descriptors to establish 2d-3d matches against the provided map
Map Builder: Synchronizes outputs of all modules to construct a VI-map

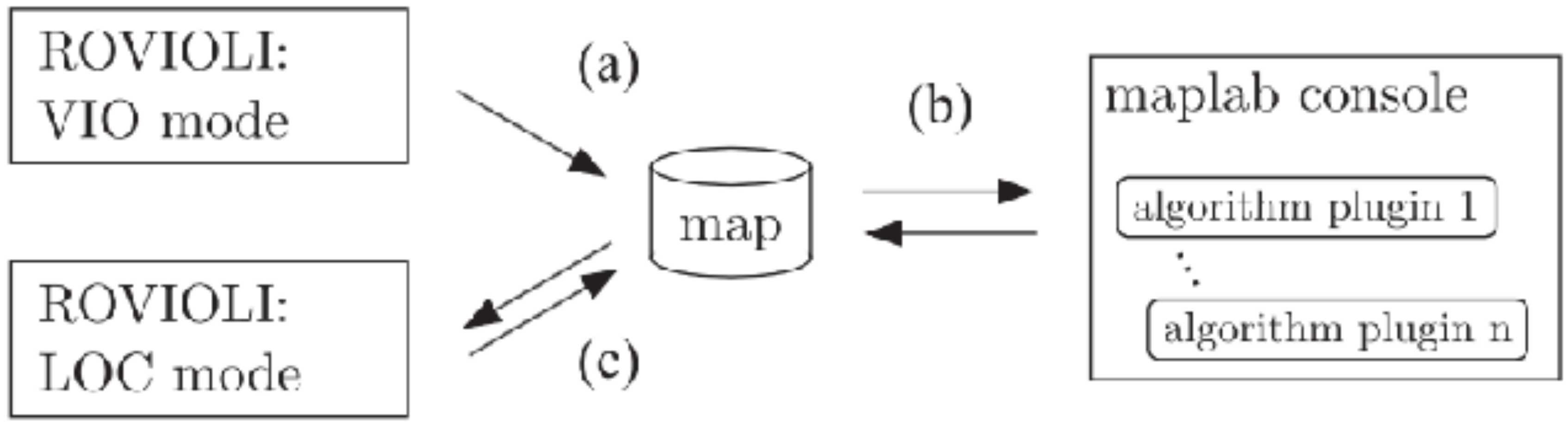
Console   The offline user interface

Console Refines the map, merges multiple maps together.

Capabilities:

- Map manager to access the maps
- A **plug-in architecture** to be easily extended
- Visual-inertial **least-squares optimization**
- Robust **pose-graph relaxation**
- BRISK/FREAK-based **loop closure**
- **Map summarization** for lifelong mapping
- Dense reconstruction
- Parameter specification at runtime

maplab workflow



(a) **VIO mode**: ROVIOLI estimates the pose of an agent w.r.t. a (drifting) local frame; additionally a map is built based on these estimates

(b) Resulting maps can be loaded in the maplab-console and all of the available algorithms can be applied (map merging, optimization, loop closure)

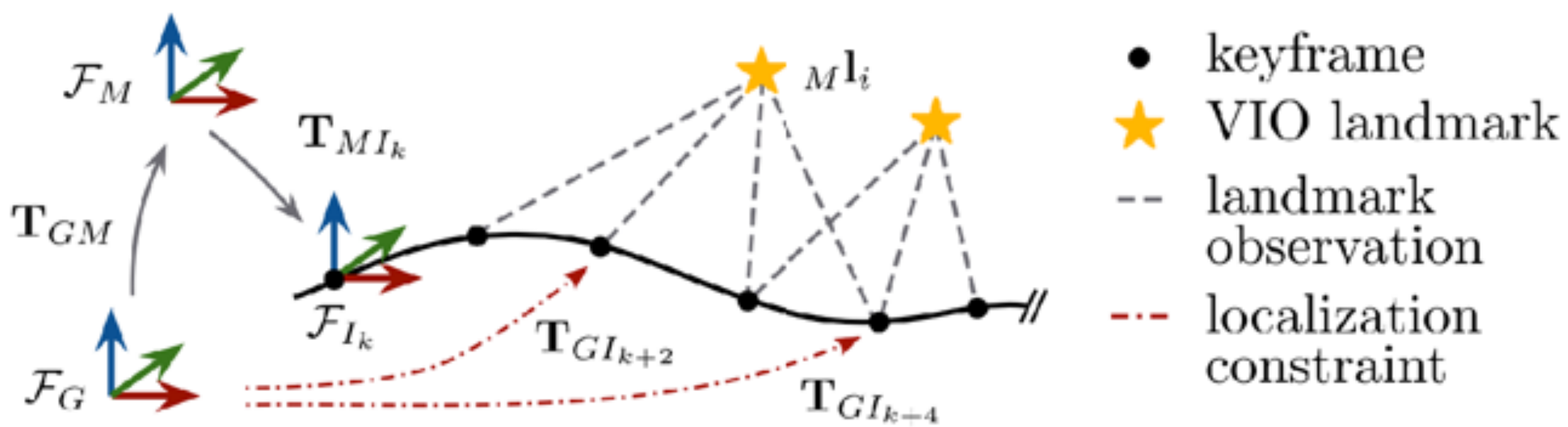
(c) **LOC mode**: ROVIOLI can load the updated map to track a global (drift-free) pose online.

maplab

Map Structure: VI-Map

VI-Map for visual-inertial mapping data, contains the **raw measurements** of all sensors and a **sparse reconstruction.**

The map can be represented as a graph contains vertices and edges



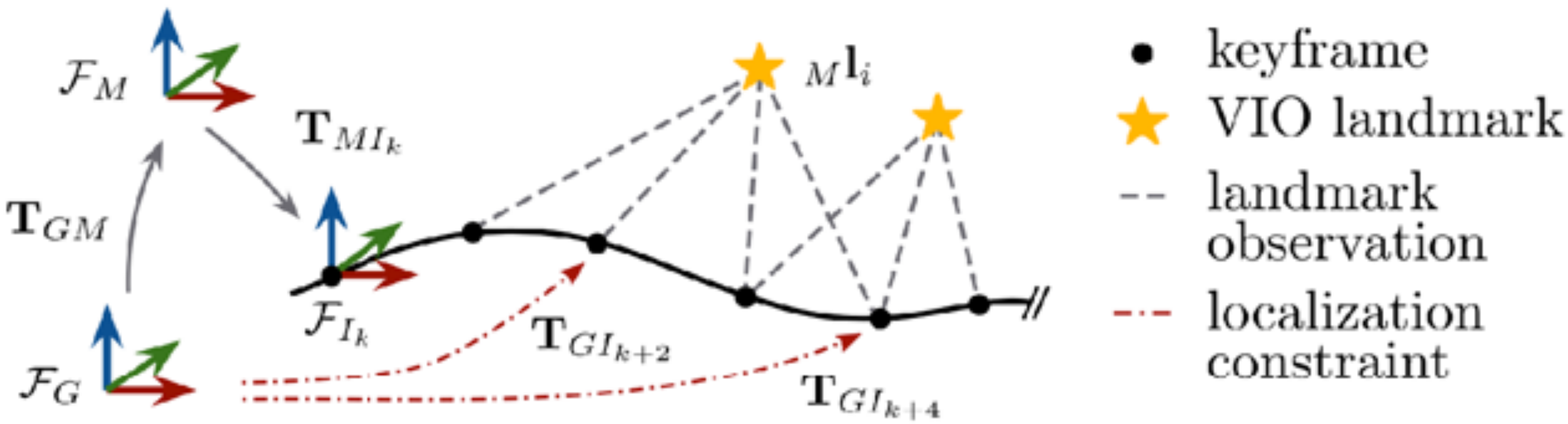
Vertex: A state captured at a certain time (pose/IMU biases/velocity/visual information)

Edge: Inertial measurements recorded between two neighboring vertices

VIO landmark: Visual observations tracked by multiple vertices are triangulated as 3D landmarks. (Stored in the vertex that first observed it)

Localization constraint: Constraints introduced by loop closure

Coordinate Frame



$\mathcal{F}_G$: global, gravity-aligned map frame

$\mathcal{F}_M$: gravity-aligned map frame that represents the origin of submap (the same as VO)

$\mathcal{F}_{I_k}$: IMU frame at timestamp k (body frame)

The pose and landmark position is based on submap frame $\mathcal{F}_M$ so that it suffices to manipulate the transformation $\mathbf{T}_{GM}$ to merge the map without updating any vertex pose or landmark position

Map Resources

Resource Management System

- The data-intensive objects (images/dense reconstructions) can be attached to the maps
- Resources are linked to either a vertex or a set of missions or simply a timestamp, and are stored separately from the main mapping data
- Large objects can be loaded on demand
- Map can be serialized by Protobuf for transmission and storage

Code Package

- **VIWLS**: visual-inertial weighted least-squares optimization used to refine map after initialization or loop closure.
- **Loop closure/localization**: a complete loop closure and localization system based on binary descriptors. The search backend uses an inverted multi-index[1] for efficient nearest neighbor retrieval on projected binary descriptors
- **ROVIOLI**: online visual-inertial mapping and localization frontend
- **Pose-graph relaxation**: pose-graph optimization using edges introduced by the loop closure system
- **aslam_cv2**: a collection of computer vision data structures and algorithms. It includes various camera and distortion models as well as algorithms for feature detection, extraction, tracking and geometric vision
- **Map sparsification**: algorithms to select the best landmarks for localization and keyframe selection to sparsify the pose graph
- **Dense reconstruction**: a collection of dense reconstruction, depth fusion and surface reconstruction algorithms

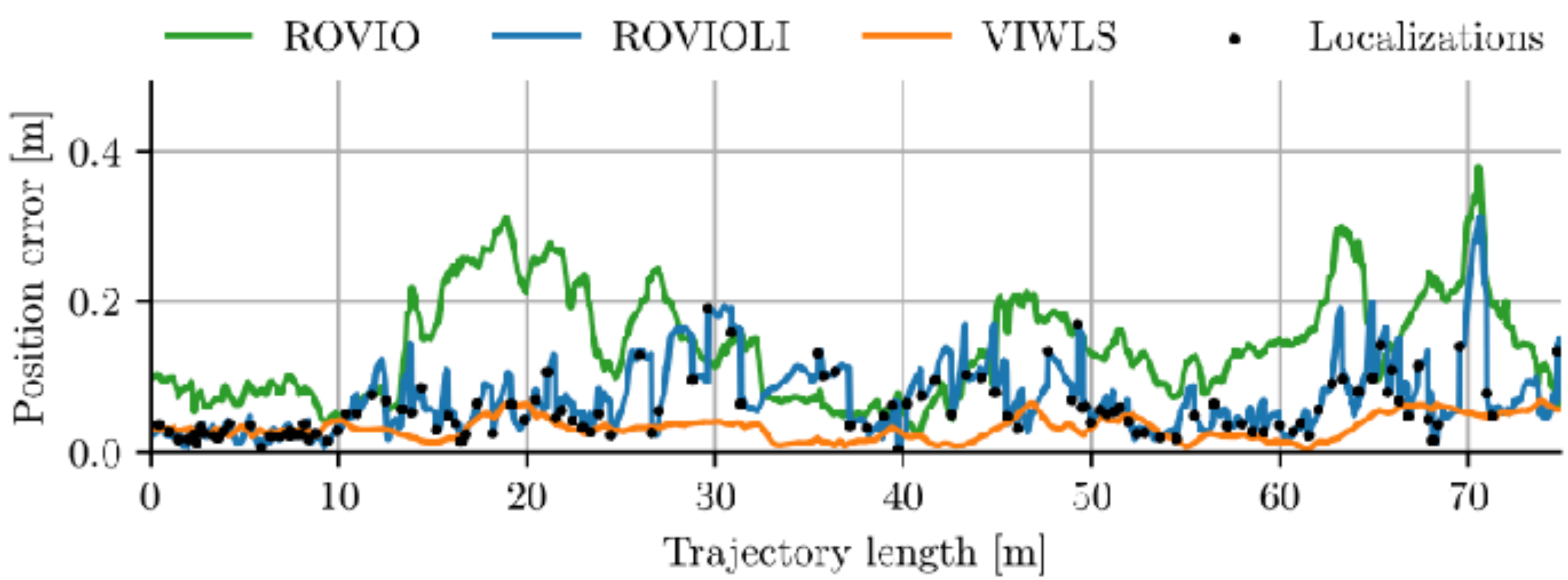
[1] Lynen, Simon, et al. "Get out of my lab: Large-scale, real-time visual-inertial localization." *Robotics: Science and Systems*. Vol. 1. 2015.

Use Cases #1 Online Mapping and Localization With ROVIOLI

Pipeline:

1. Use ROVIOLI to create an initial VI-map of the desired area
2. Upon completion, VI-map is automatically loop closed, optimized
3. Localize with the optimized map

Evaluation:



Use Cases #1 Online Mapping and Localization With ROVIOLI

Result

	MH1 *LOC: MH2		V2-easy *LOC: V2-medium	
	position	orientation	position	orientation
ROVIO	0.178 m	1.49 deg	0.064 m	0.90 deg
ROVIOLI*	0.082 m	1.43 deg	0.057 m	1.57 deg
ROVIO+ VIWLS	0.036 m	1.29 deg	0.027 m	1.06 deg
ORB-SLAM2* (batch mode)	0.084 m	0.78 deg	0.121 m	1.14 deg
ORB-SLAM2* (real-time)	0.464 m	13.34 deg	X	X

(a)		
	Frame update	CPU load
ROVIO	23 ms	56% ± 7.7%
ROVIOLI*	44 ms	105% ± 14.8%
ORB-SLAM2* (batch mode)	63 ms	162% ± 10.9%
(b)		
ROVIO update	22.7 ms	
Feature tracking	20.6 ms	
Localization	20.4 ms	
Map building	3.2 ms	
Total	**44.2 ms**	

Use Cases #2 Multi-Session Mapping

This use case demonstrates the process of creating a map of a university building

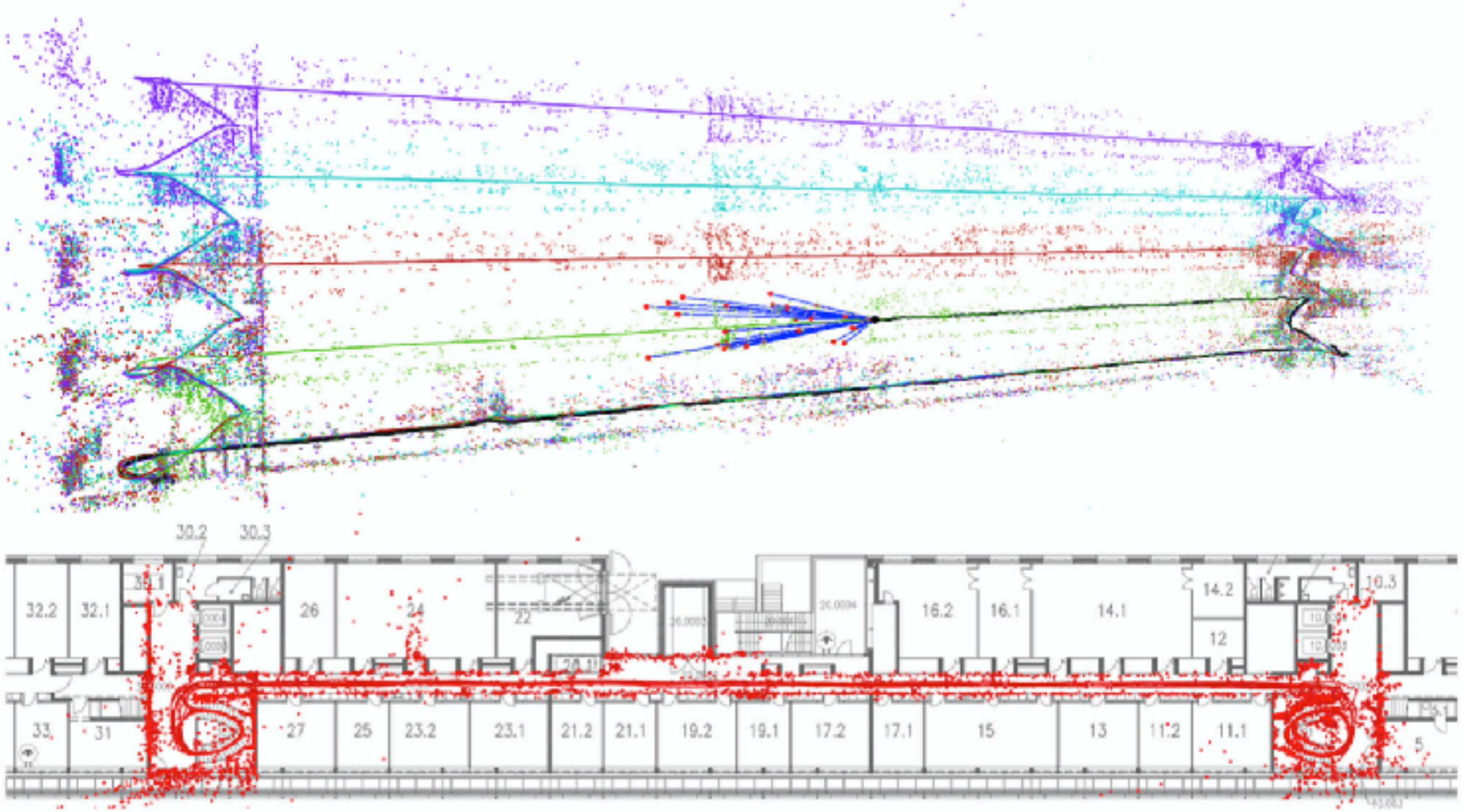
- Four individual trajectories
- Cover over 1,000 meters and contain about 463,000 landmarks
- Employ a keyframe pruning scheme using heuristics based on vertex distance, orientation, and landmark covisibility

Commands

```
# Load multiple single session maps from ROVIOLI.
load_merge_all_maps --maps_folder YOUR_MAPS_FOLDER
# Keyframing and initial optimization.
kfh
optvi
# Set one mission as base, anchor the others.
set_mission_baseframe_to_known
anchor_all_missions
# Pose-graph relaxation, loop-closure, optimization.
relax
lc
optvi
```

Use Cases #2 Multi-Session Mapping

Map



The result is a compact, geometrically-consistent localization map of 8.2 MB ready to be used

Use Cases #3 Map Maintenance

Setup:

- Experiment on the previous university building map
- Each session covers about 90 meters
- Each session contains 20,000 landmarks, 5000 are considered reliable.
- A fifth dataset is used as a query

Sparsification Scheme:

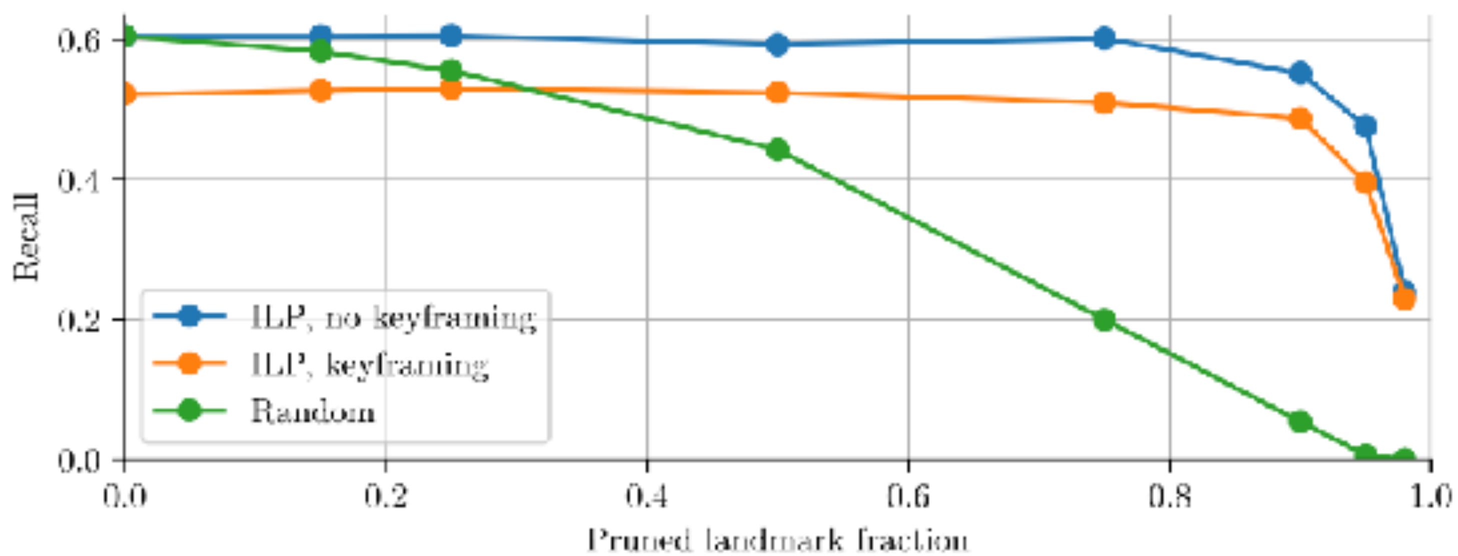
- Integer-based optimization to select landmark
- Attempts to remove the least commonly seen landmarks but at the same time maintain a balanced coverage of the environment
- Keyframe pruning algorithm removes redundant vertices and only keep the ones necessary

Use Cases #3 Map Maintenance

Commands:

```
# Keyframe the map and sparsify landmarks to 10,000.
kfh
landmark_sparsify --num_landmarks_to_keep=10000
```

Result:



pruning fraction	0	0.5	0.75	0.9	0.95	0.98
# landmarks	18,316 (16,088)	8,824 (9,148)	4,259 (4,570)	1,818 (1,822)	899 (906)	349 (358)
map size [MB]	34.559 (3.740)	29.217 (3.209)	24.028 (2.602)	17.619 (1.837)	12.203 (1.214)	6.707 (0.712)

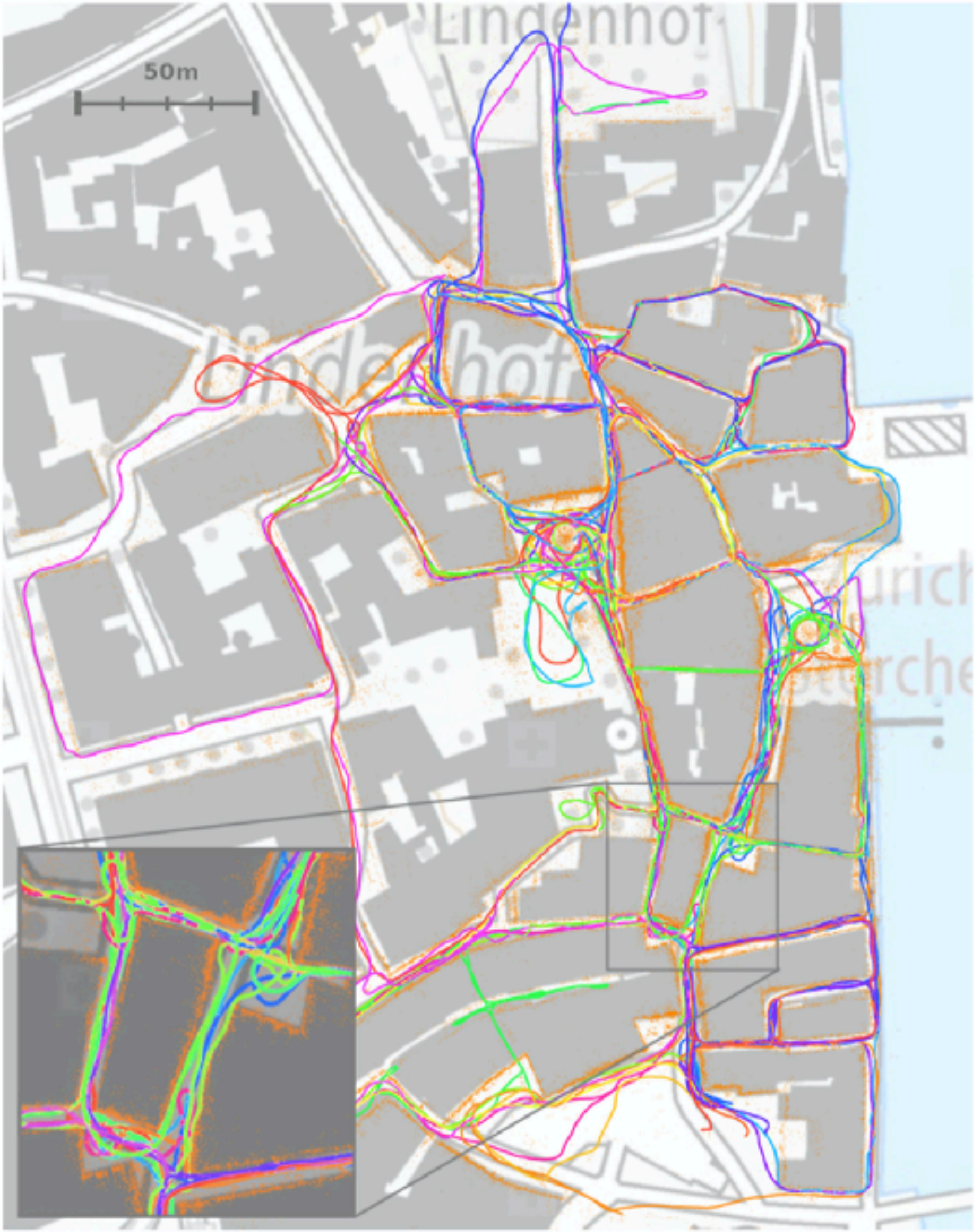
Localization performance and map size after ILP landmark summarization and keyframe summarization (in brackets)

Use Cases #4 Large-Scale Mapping

- Google Tango tablets
- Large-scale, multi-session map of the old town of Zurich
- 45 sessions
- 231 min data over two days
- 435k landmarks
- 7.3M observations
- 480 MB map size
- Map processing on a 32 GB RAM desktop computer overnight

Use Cas

- Google T
- Large-sc
- 45 sessio
- 231 min
- 435k lan
- 7.3M obs
- 480 MB r
- Map proc ght

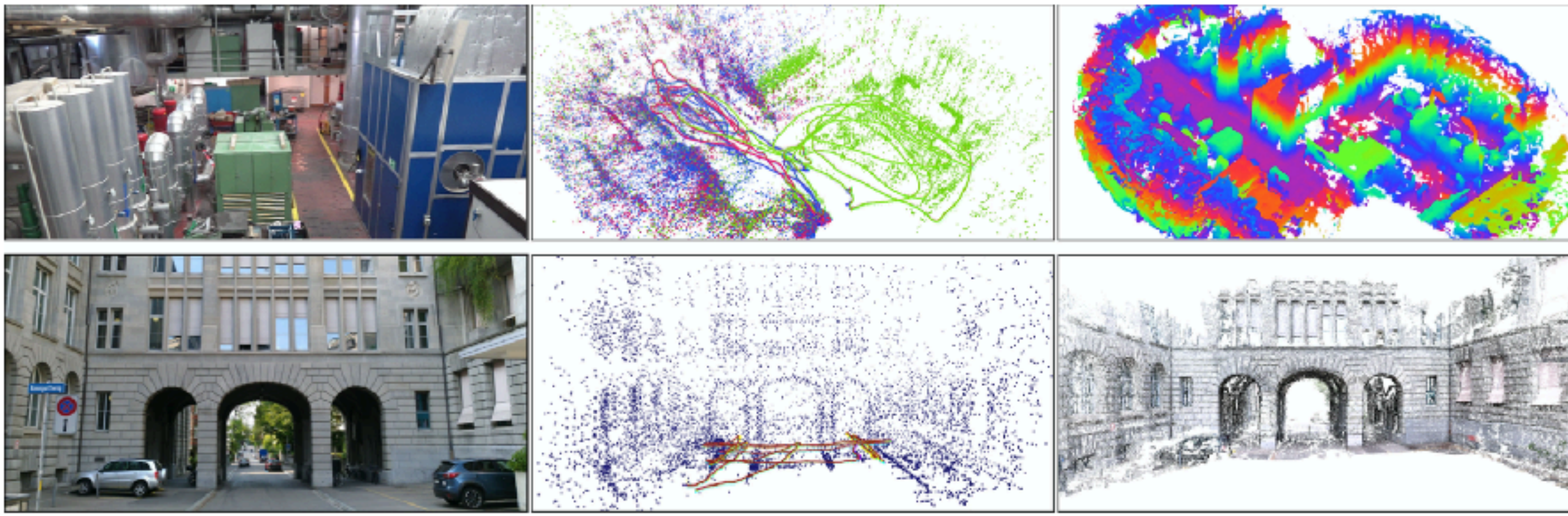


Use Cases #5 Dense Reconstruction

Maplab offers several dense reconstruction tools, which use the optimized vertex poses of the sparse map to compute dense depth information based on camera images attached to the VI-map.

- **Stereo Dense Reconstruction**: this tool first identifies stereo cameras that are suitable for planar rectification. It then utilizes a (semi-global) block matcher to compute depth maps for every stereo pair along the trajectory
- **TSDF-Based Depth Fusion**: maplab uses depth information to get map optimized utilized to create an equally consistent global 3d reconstruction. maplab employs *voxblox* for TSDF-based depth fusion and surface reconstruction
- **Export to CMVS/PMVS2**: maplab offers an export command to convert the sparse VI-map and images to the input data format for the open-source multi-view-stereo pipeline CMVS/PMVS2

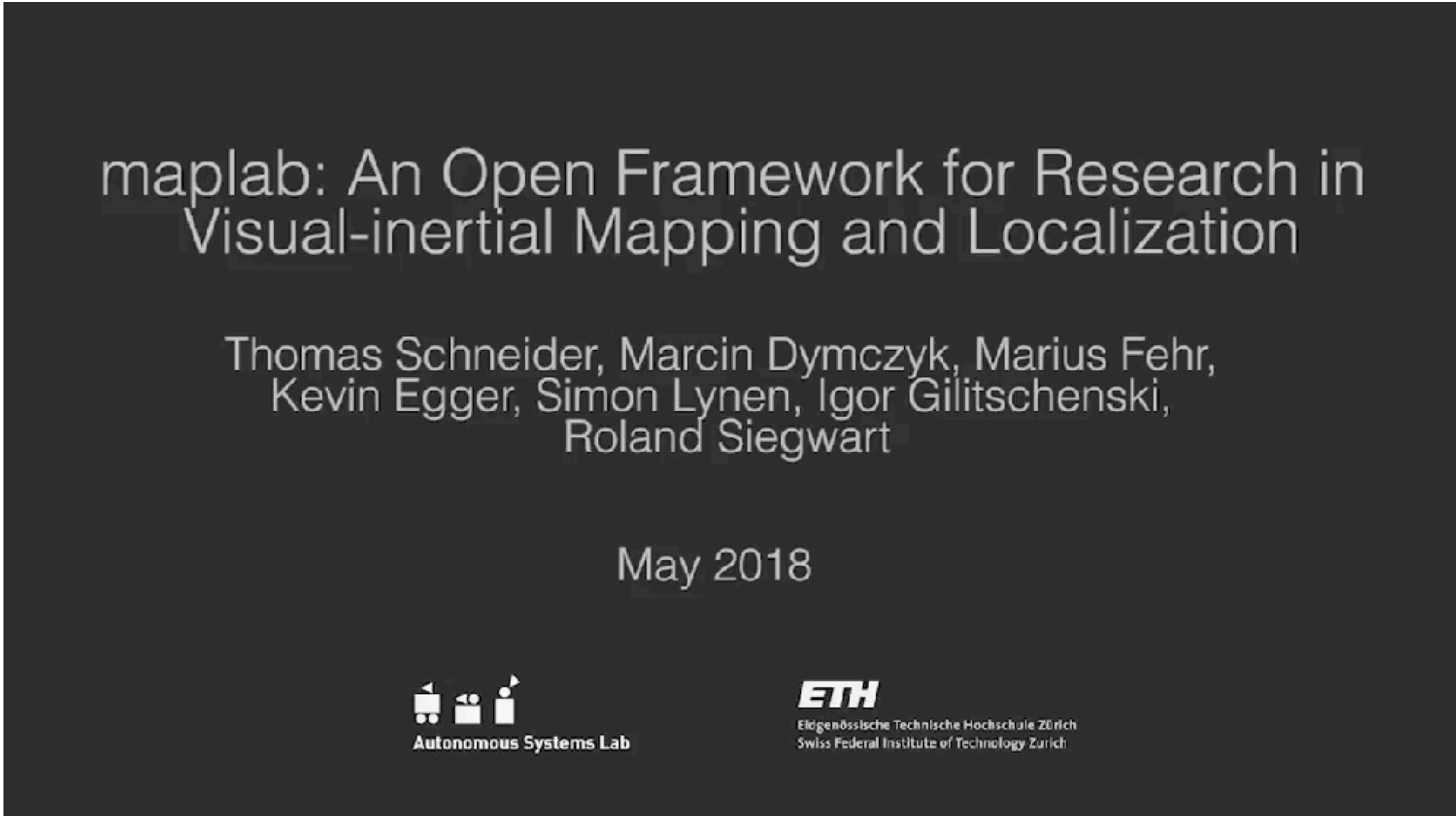
Use Cases #5 Dense Reconstruction



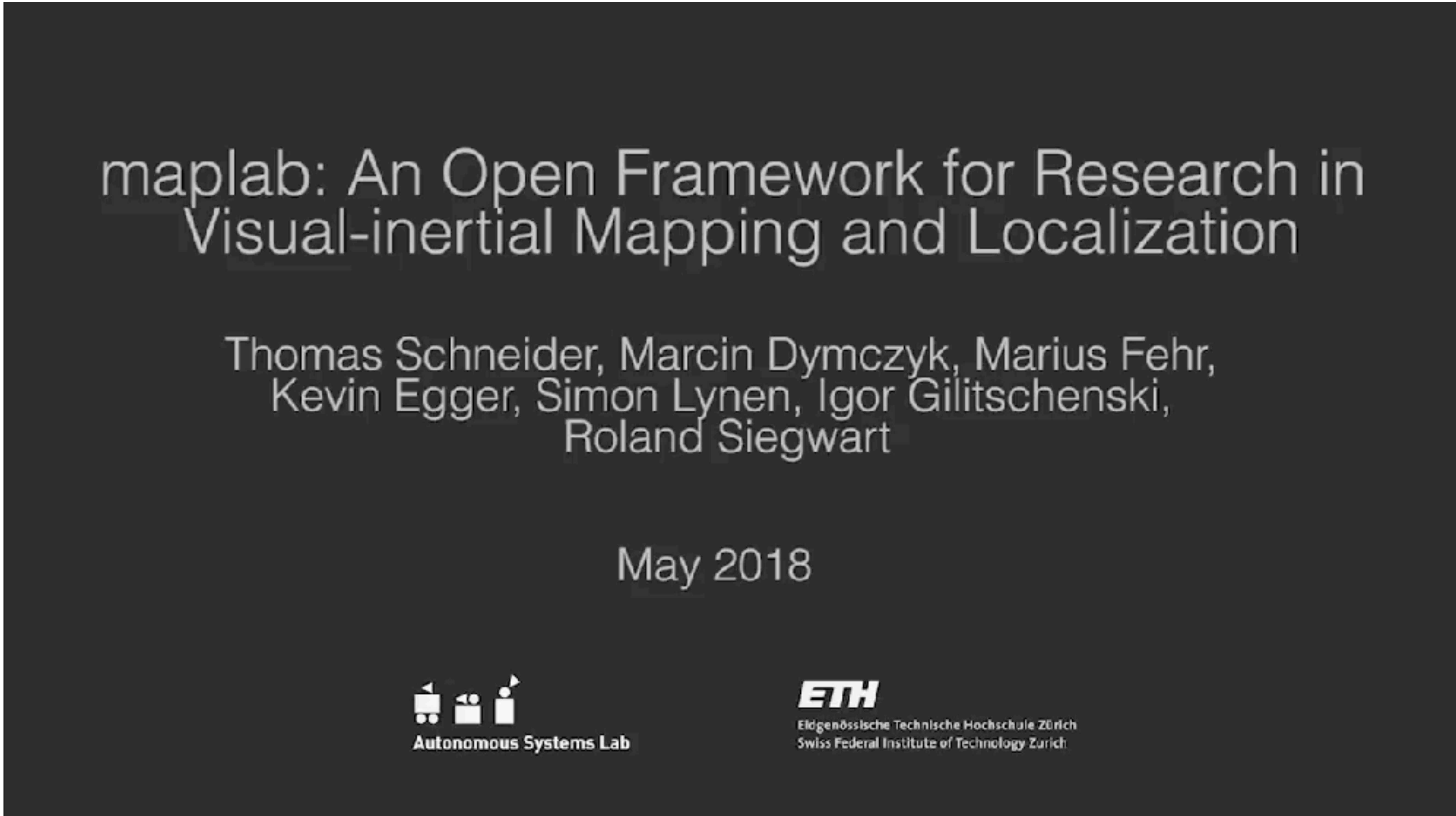
Top: stereo dense reconstruction is used to compute depth maps based on grayscale images and optimized camera poses. They are then fused in *voxblox* to create a surface mesh.

Bottom: CMVS/PMVS2 reconstruction results using a multi-camera system with a RGB camera.

Demo


maplab: An Open Framework for Research in Visual-inertial Mapping and Localization
Thomas Schneider, Marcin Dymczyk, Marius Fehr, Kevin Egger, Simon Lynen, Igor Gilitschenski, Roland Siegwart
May 2018
Autonomous Systems Lab
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Demo


maplab: An Open Framework for Research in Visual-inertial Mapping and Localization
Thomas Schneider, Marcin Dymczyk, Marius Fehr, Kevin Egger, Simon Lynen, Igor Gilitschenski, Roland Siegwart
May 2018
Autonomous Systems Lab
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Conclusion

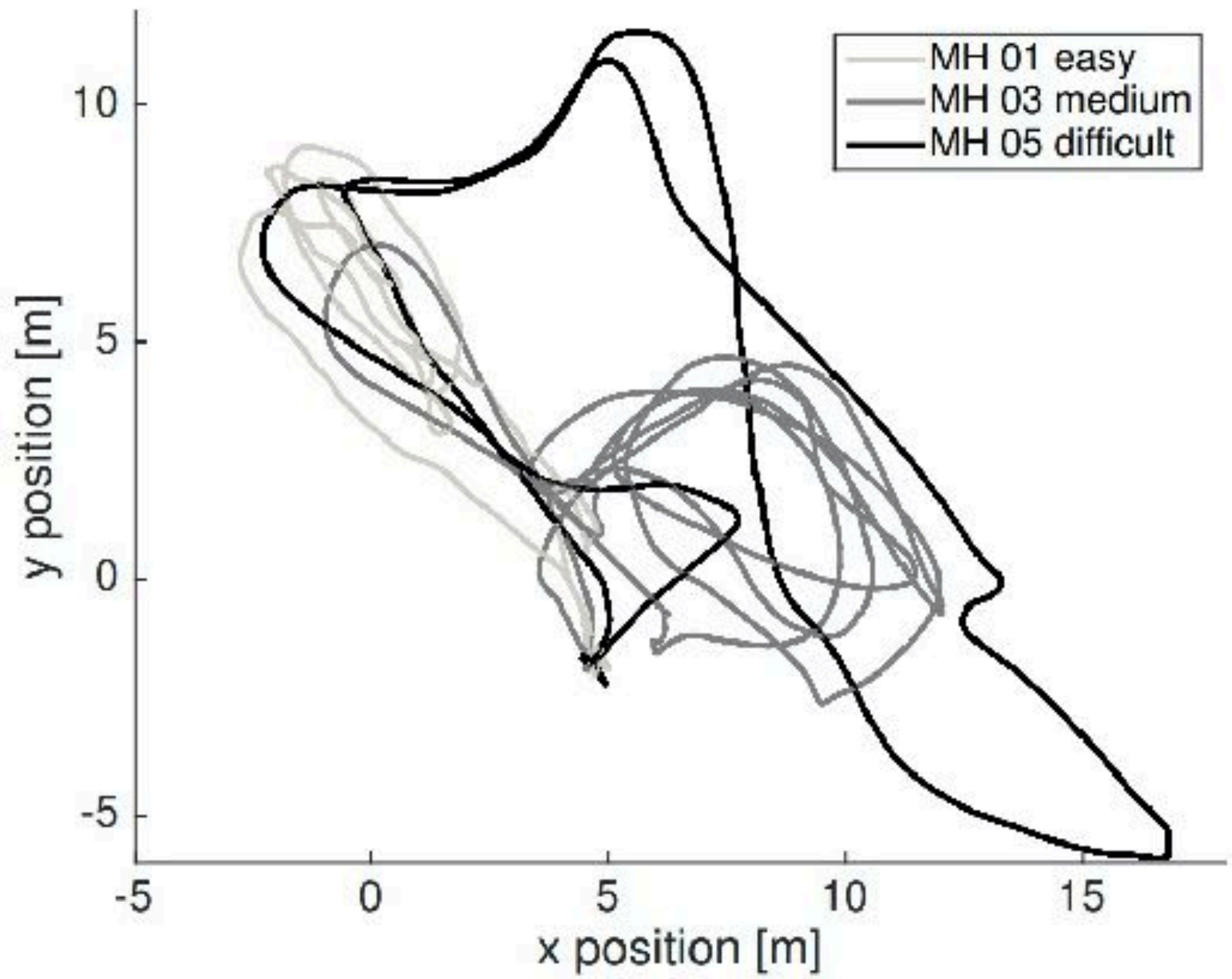
Highlights:

- Making research more efficient by providing a collection of basic algorithms
- All components are written in a flexible and extensible way
- Provides an implementation of the most important tools such as optimization, loop closure, multi-session map merging, pose-graph relaxation and extensive introspection and visualization tools
- Present an online visual-inertial mapping and localization frontend ROVIOLI
- Open source / Uses google-c++ code style / Unit-tests / Detailed wiki

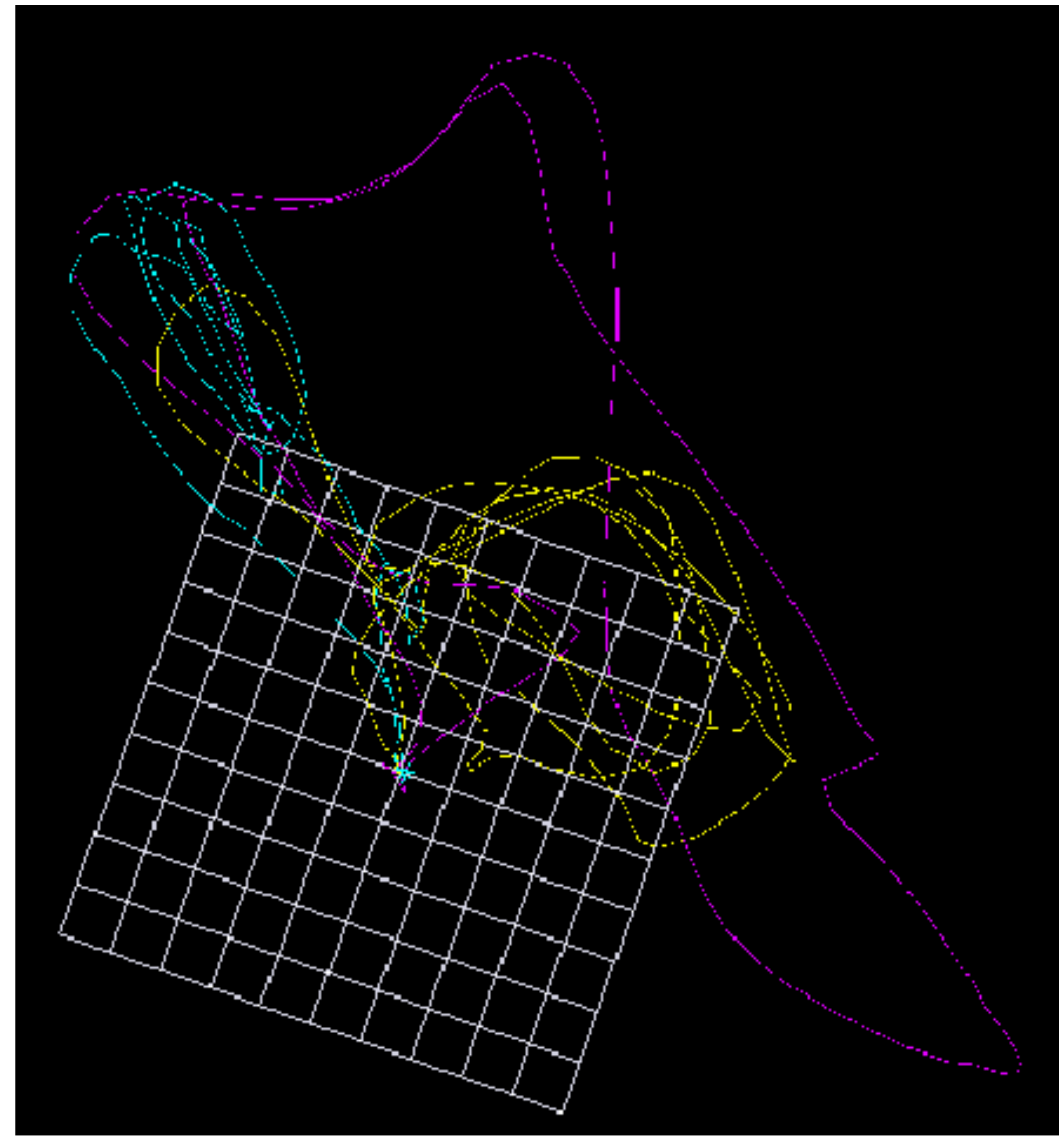
Limitations:

- The frontend is tightly coupled and there's no interface to replace frontend to ORB except rewriting the whole part
- The system is unable to accomplish simultaneously localization and mapping
- Why not just integrate a SLAM system instead of using ROVIOLI

Map Merging on EuRoC MH Dataset



Groundtruth



Result

```
maplab <vi_app_test>:/$ ls
VIMaps saved in the storage (2 total):
    my_empty_map -> No map folder defined!
  * vi_app_test -> /home/eggerk/catkin_ws/src/maplab/tools/maplab_test_data/tes
t_maps/vi_app_test
```

Reflection

学习的角度:

- 作为地图融合、地图压缩、多地图管理等很好的学习材料
- 提供了可视化、地图压缩、调节参数、验证算法的工具

贡献的角度:

- 在 Large Scale EdgeSLAM 和之前的项目开发过程中，会积累很多分析、实验工具（Relocalizer/MapOptimizer/MapStatistics/MapManager/LoopCandidatesLookup）。可以作为论文附带的 tool set 开源，也可以继续优化、封装，作为基于ORB SLAM的或者通用的SLAM开发辅助工具发布。

Thank you