

# **SLAM and Localization Research at Nagoya University**

Naoki Akai

Designated Associated Professor, Nagoya University

CEO, LOCT Co., Ltd.

# Background

## □ How can robots generate appropriate actions in complex environments?

- Generating appropriate actions in environments where it is difficult to define safe rules remains a challenging problem in robotics.



## □ Map-based navigation vs. Physical AI

### ➤ Map-based navigation

- Safe rules can be defined on a pre-built map.
- ✗ The mapping process is always time-consuming.

### ➤ Physical AI

- Appropriate actions can be generated using AI.
- ✗ The reliability of AI is uncertain.

Map-based approaches are popular and we focus on **applications of SLAM** beyond autonomous driving.

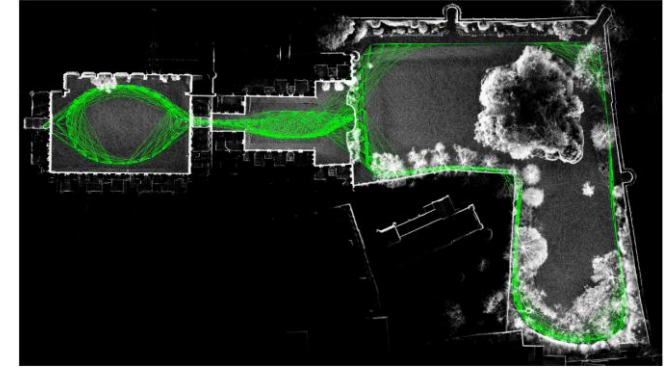
## □ Research keywords

- Bayesian Network, Factor Graph Optimization, SLAM, LiDAR
- Particle Filter, Kalman Filter, Gauss-Newton method

# SLAM and Localization Research at Nagoya University

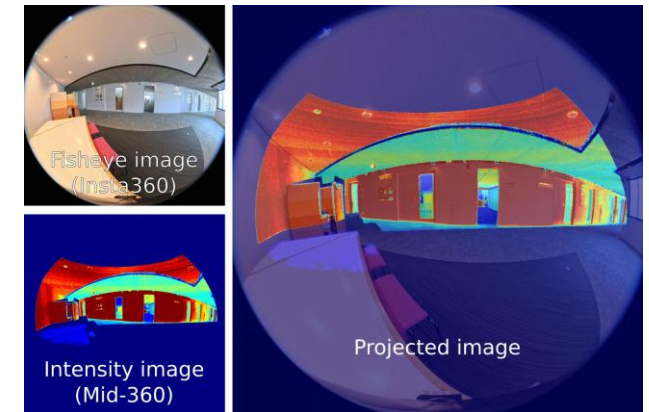
## □ LiDAR-inertial SLAM

- plain\_slam\_ros2 [https://github.com/NaokiAkai/plain\\_slam\\_ros2](https://github.com/NaokiAkai/plain_slam_ros2)
- LiDAR-inertial odometry (LIO), SLAM, and localization are integrated into a single package.



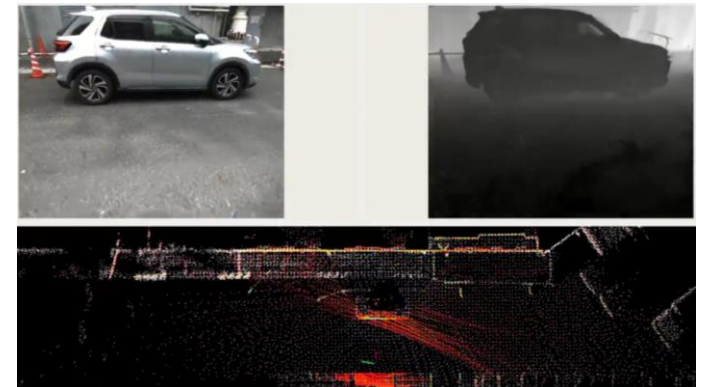
## □ LiDAR-camera calibration

- Intrinsic and extrinsic parameters are simultaneously estimated.
- Cameras that are not supported by OpenCV can be calibrated.



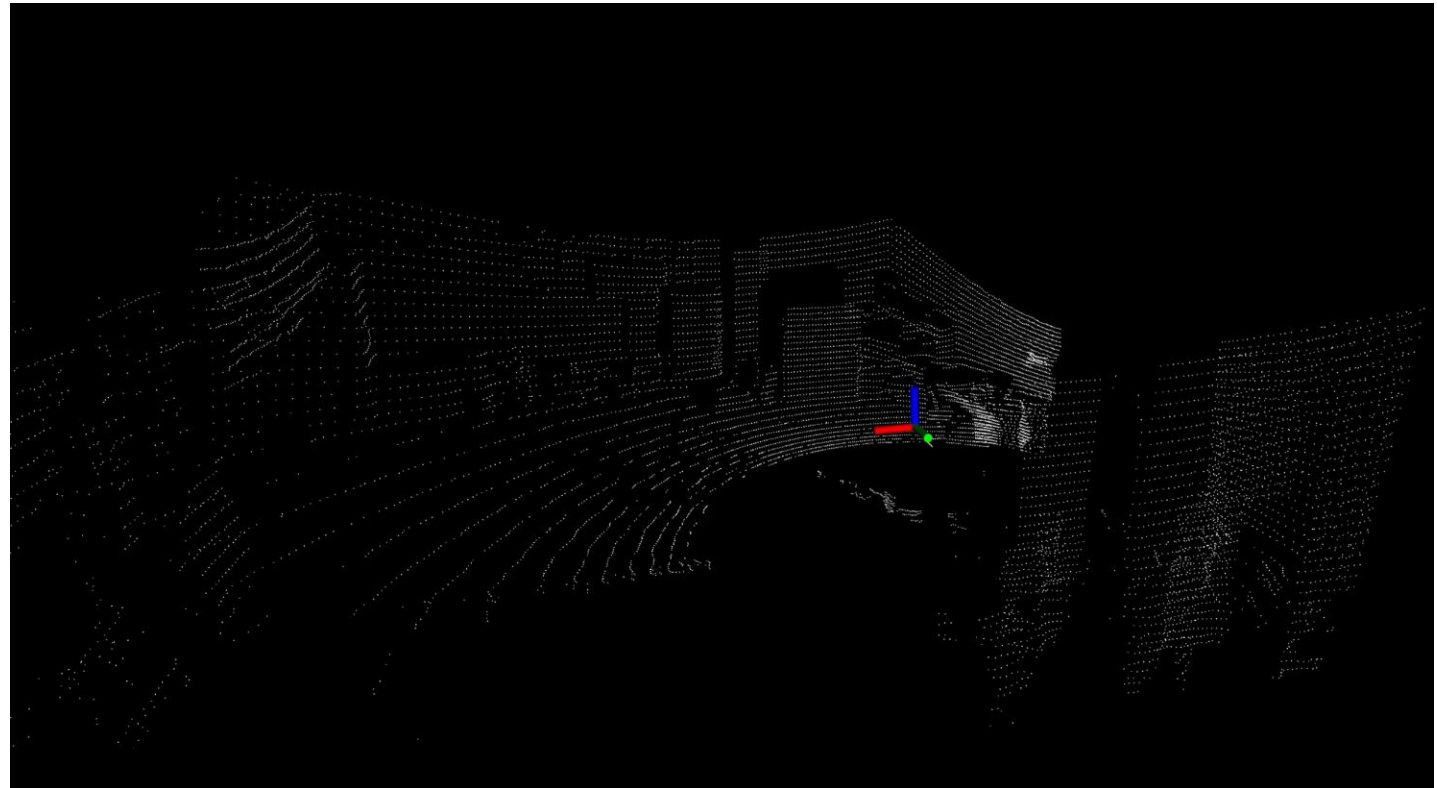
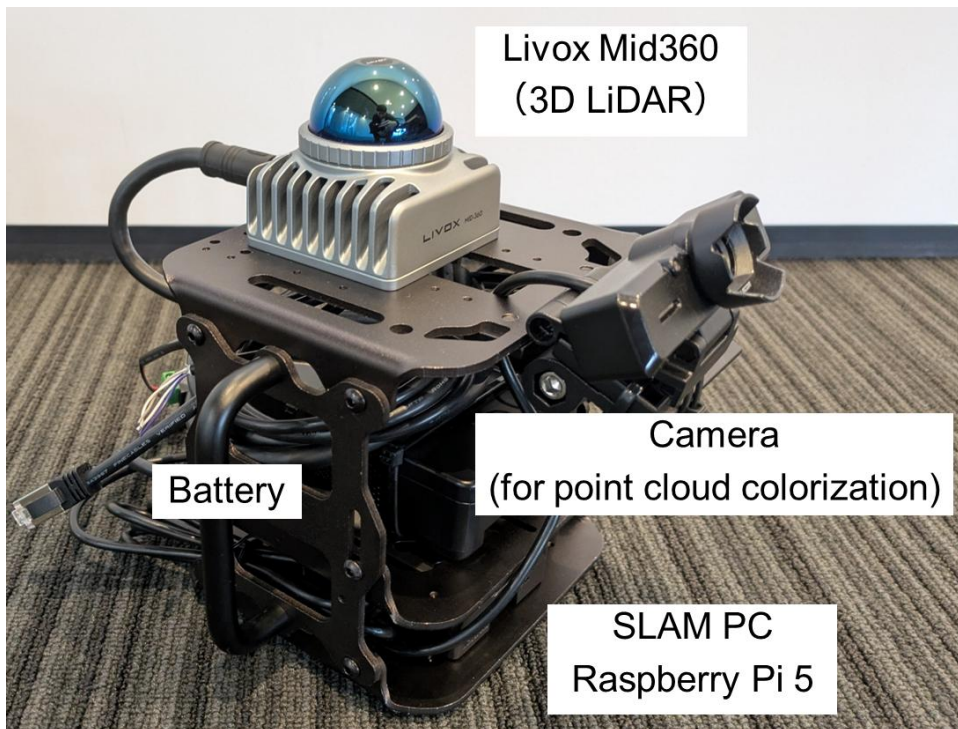
## □ Monocular metric localization using Gaussian splatting (GS)

- GS models are constructed based on LiDAR SLAM results.
- Photorealistic renderings generated by GS are used to localize a monocular camera.



# Portable SLAM Device and Robust LiDAR-Inertial Odometry

- All the devices needed for SLAM are integrated into a single portable device.
  - Our LIO can track aggressive sensor motions, with angular velocities exceeding 180 deg/s.
  - Estimating gyroscope and accelerometer biases and correcting motion distortion

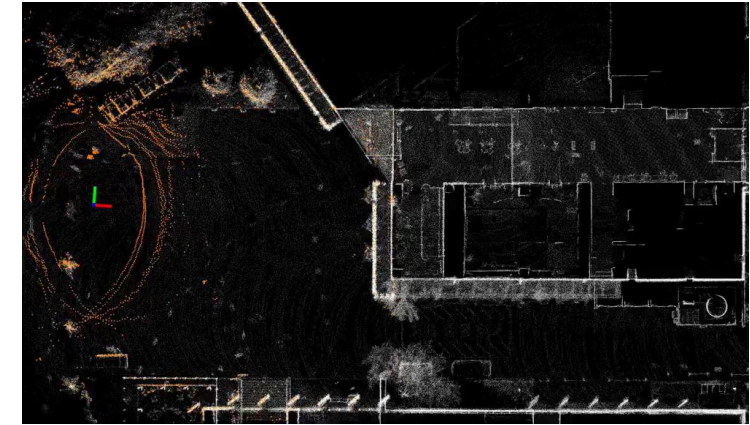


# LiDAR-Inertial SLAM (plain\_slam\_ros2)

□ Simple and concise implementation including LIO, SLAM, and localization.

➤ Core C++ source code contains fewer than 1,900 lines.

- ✓ Both loosely and tightly coupled approaches for LIO
- ✓ Loop detection based on GICP
- ✓ Pose graph optimization for global consistency
- ✓ Point-cloud-map-based localization using the same approaches as LIO



□ Mathematical details are presented in the GitHub repo.

- The use of Lie algebra and Lie groups is detailed.
- Educational Materials for SLAM

## 2.5 Lie Group and Lie Algebra

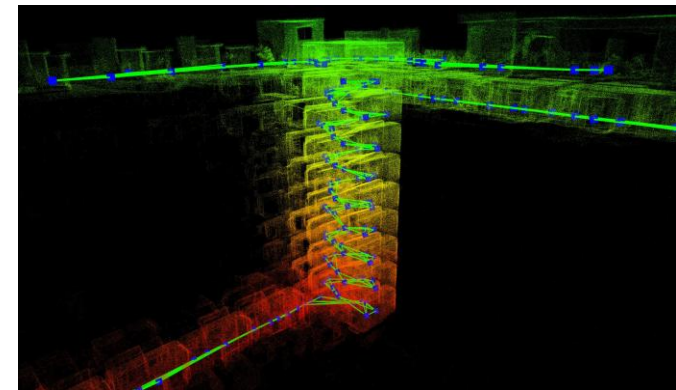
In this book, we frequently make use of **Lie groups** and **Lie algebras**. While a rigorous explanation of Lie groups and Lie algebras is beyond the scope of this text, throughout this book the term Lie group will specifically refer to **rotation matrices** and **rigid transformation matrices**.

A rotation matrix is defined as follows.

$$\{R \in \mathbb{R}^{3 \times 3} | R^T R = I, \det(R) = 1\}. \quad (2.24)$$

A rotation matrix, also referred to as the Special Orthogonal group in three dimensions ( $SO(3)$ ), represents rotations in three-dimensional space. A rigid transformation matrix is defined as follows.

$$\left\{ \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \in \mathbb{R}^{4 \times 4} | R \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\}. \quad (2.25)$$



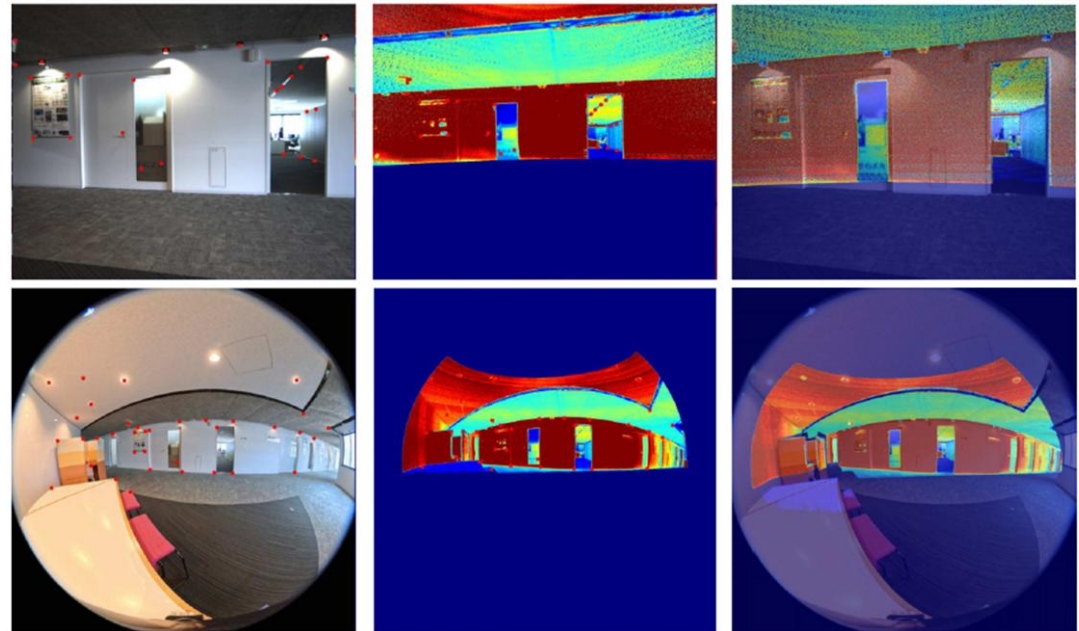
The SLAM result using the legged robot

# LiDAR-Camera Calibration

- A typical process for LiDAR-camera calibration consists of:
  - Estimating intrinsic parameters
  - Estimating extrinsic parameters based on given correspondences and intrinsic parameters

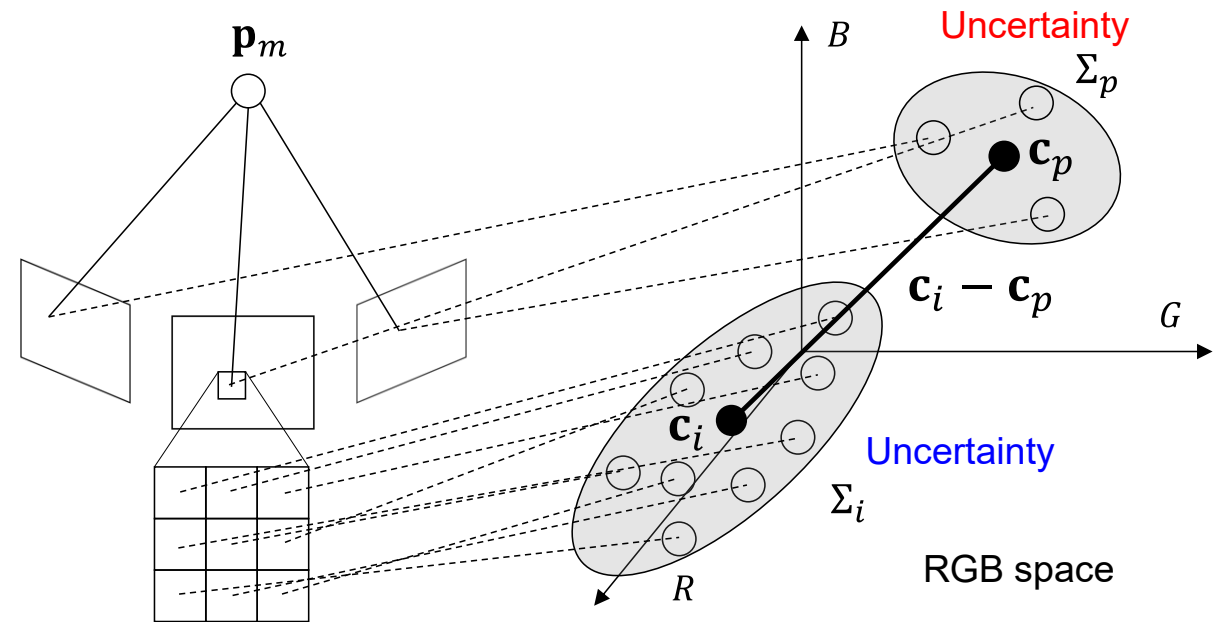
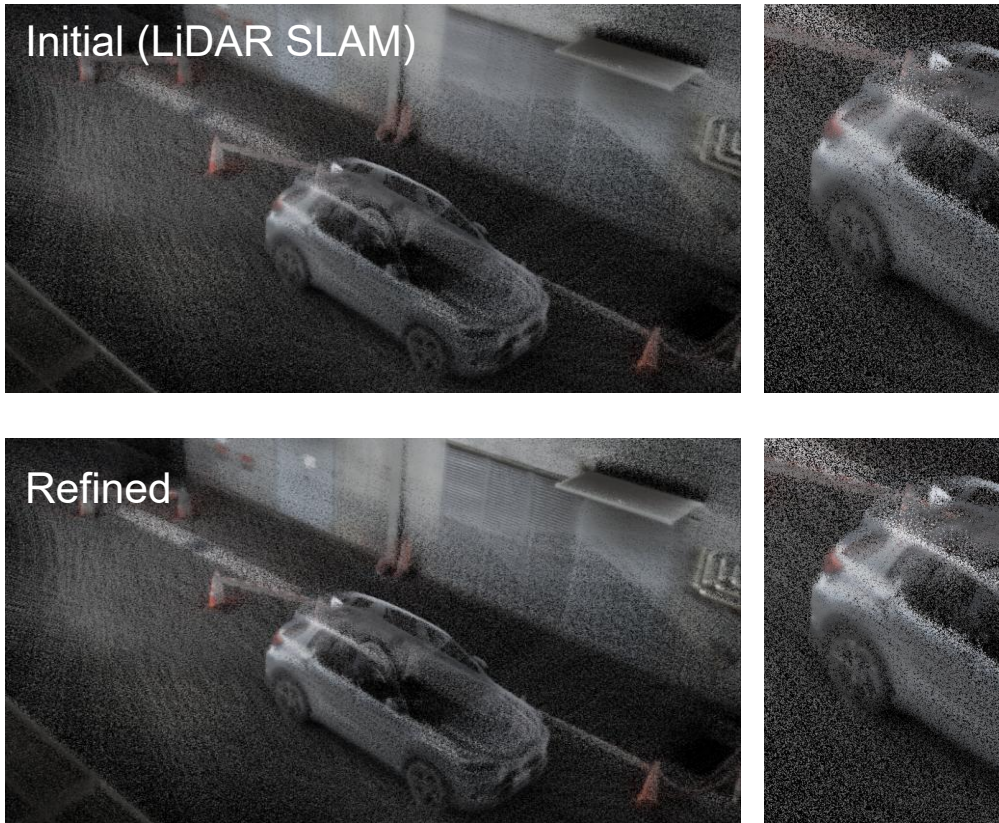
Intrinsic parameters must be precisely estimated before extrinsic calibration.

- Our method can calibrate LiDAR-camera systems without estimating intrinsic parameters.
  - Both intrinsic and extrinsic parameters are simultaneously optimized with factor graph optimization.
  - Cameras that are not supported by OpenCV, e.g., Insta360 (bottom), can be calibrated.



# Colored Point Cloud Generation

- ❑ Colored point cloud can be generated using the calibrated LiDAR-camera system.
  - However, there are **spatial and temporal synchronization errors** in the estimated camera trajectory.
  - These synchronization errors are compensated for using color-uncertainty-aware camera pose refinement.

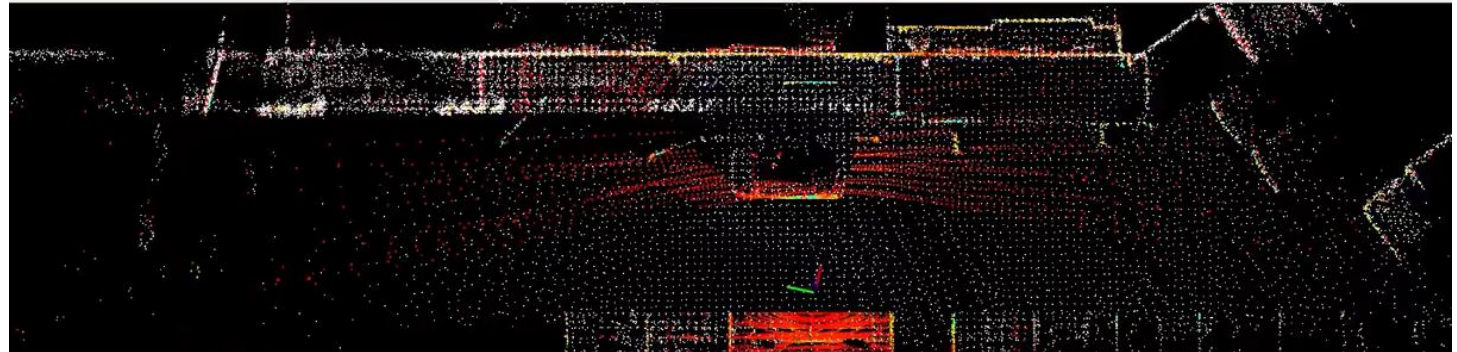
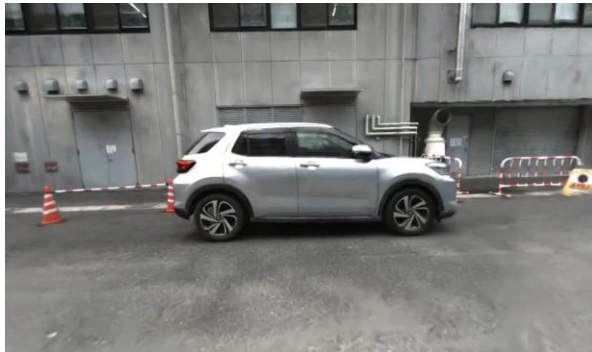


# Colored Point Cloud to Gaussian Splatting Models

- ❑ Realistic environment modeling using Gaussian splatting (GS)
  - GS models with an accurate metric scale are generated based on LiDAR SLAM results.
  - GS rendering can be performed in metric space.

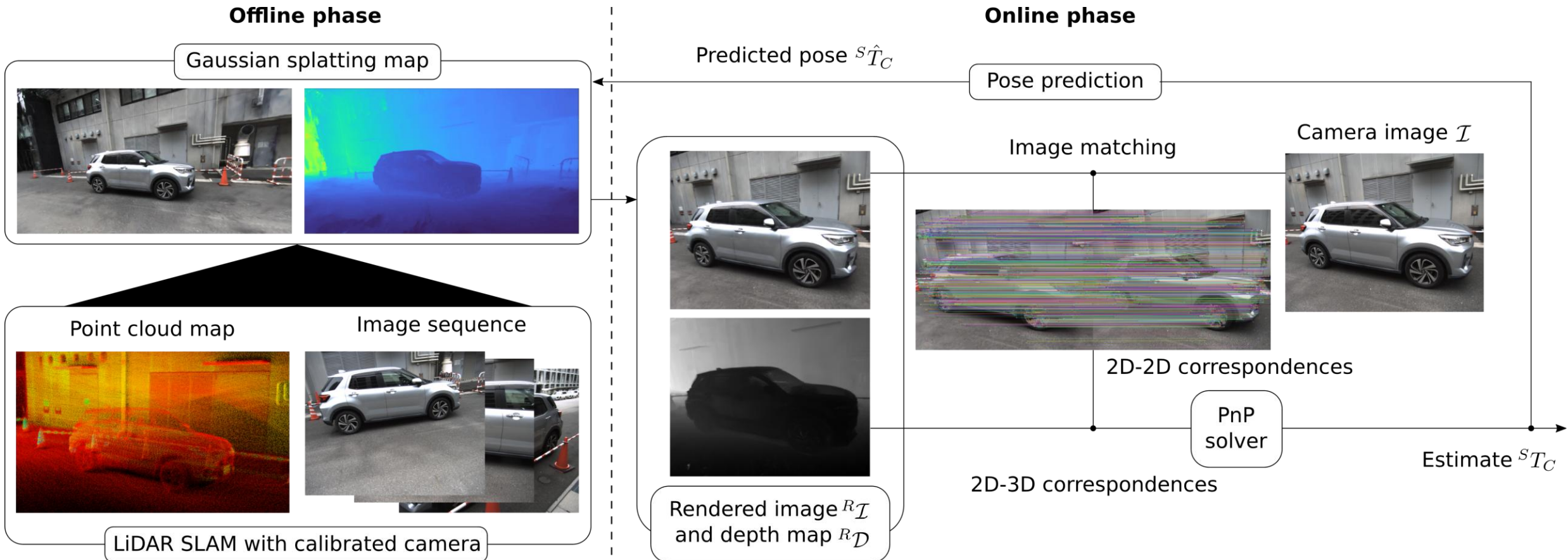


↓ GS optimization



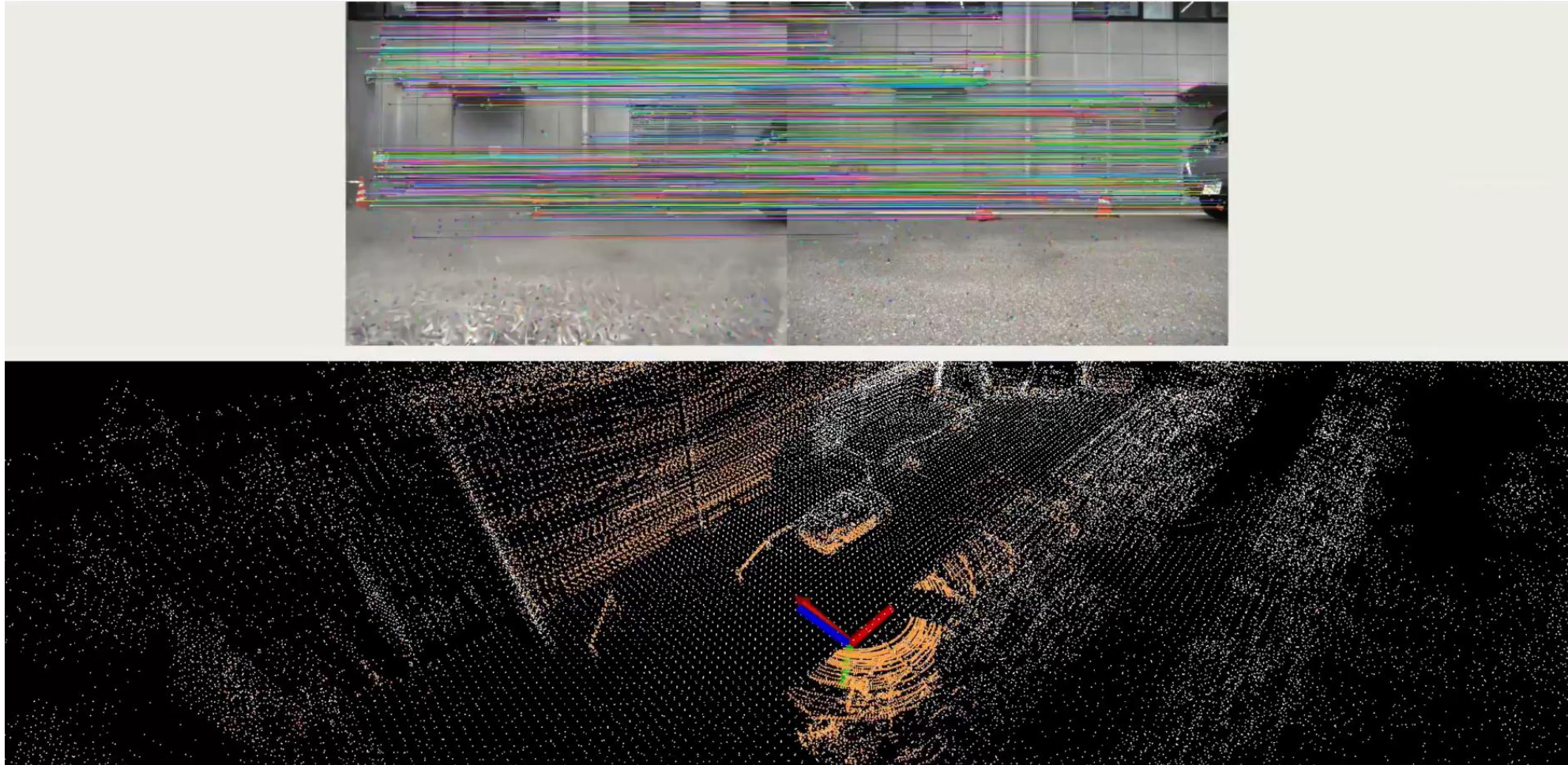
# Monocular Metric Localization Using GS Models

- GS models are used as maps for high-fidelity image rendering with metrically accurate depth maps.
  - Image and depth maps are rendered according to predicted poses.
  - 2D–3D correspondences are established between live camera images and the rendered maps.



# Monocular Metric Localization Using GS Models

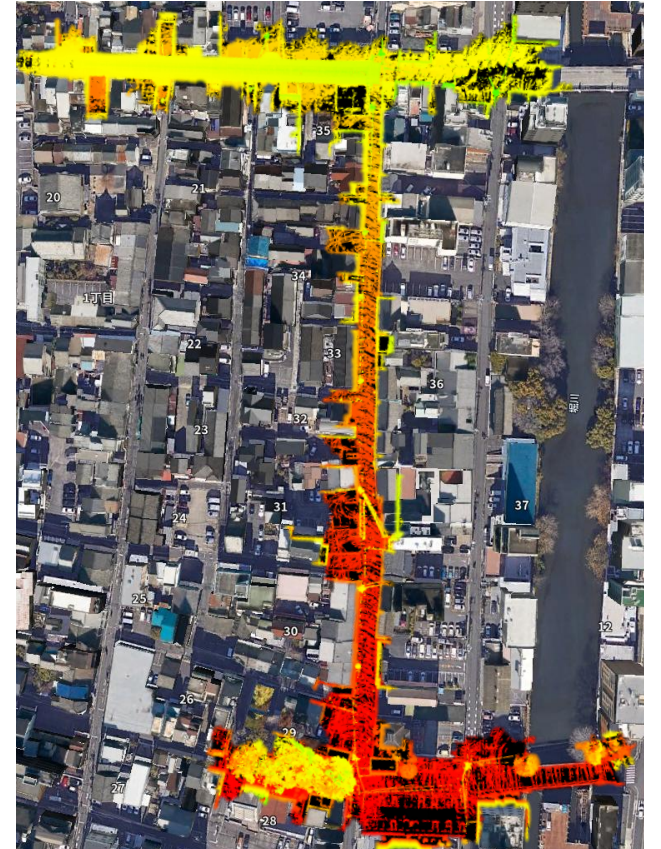
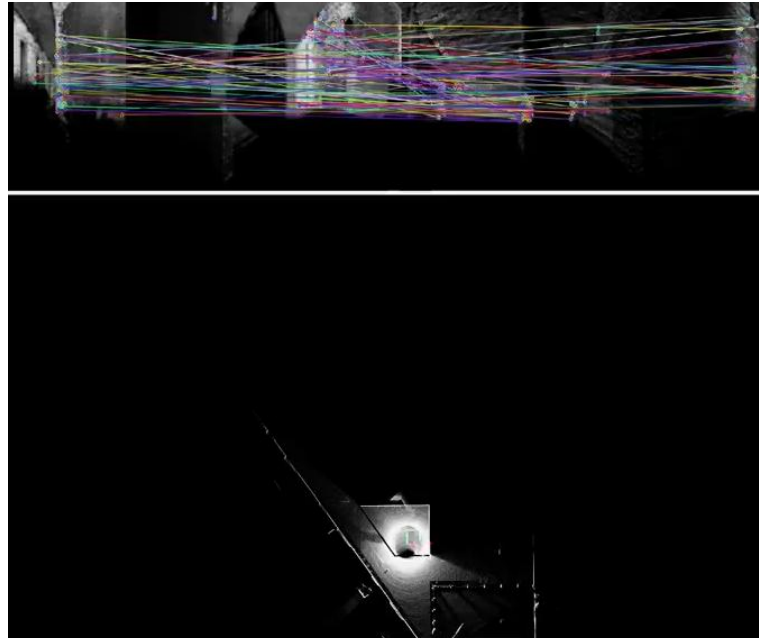
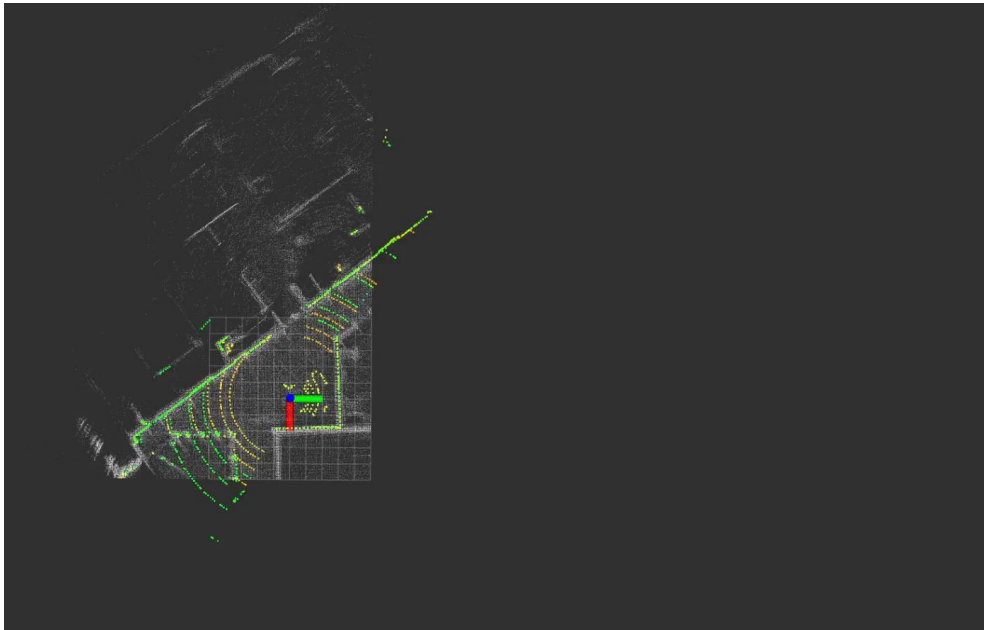
- Monocular localization (3 axes) vs. LiDAR-based localization (red arrow)
  - The accuracy of our method was comparable to that of the LiDAR-based method.



# Other SLAM and Localization Research

## □ Increasing Localization Flexibility (Pose-Sharing System)

- Localization outside of prebuilt maps (left)
- Trajectory estimation using a partial map (middle)
- Trajectory estimation in GNSS coordinates without a prior map (right)

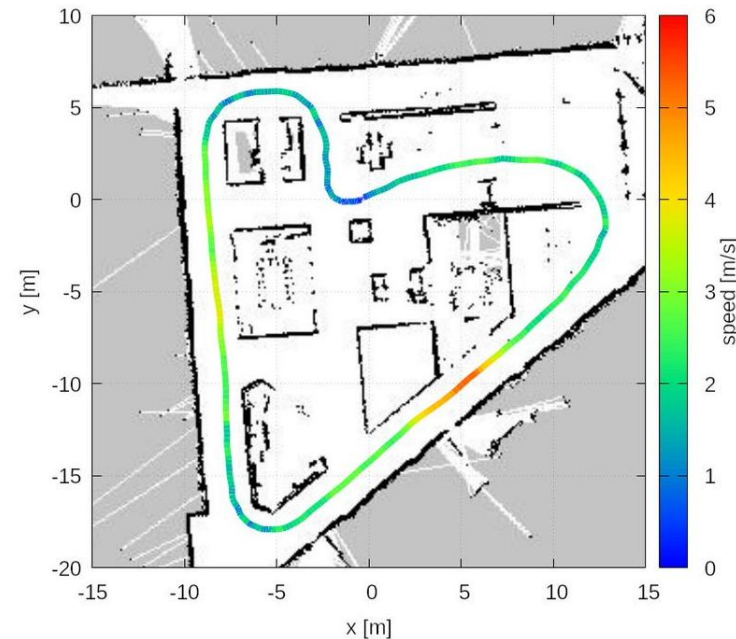


# 2D Small SLAM Sensor (developed by LOCT)

- ❑ The SLAM system (LIO, SLAM, and localization) is integrated into a small device (110 g).
  - Neither ROS nor Linux is required.
  - Users can operate the device via a web-based UI.
  - SLAM data (pose, map, sensor data, etc...) can be streamed via Ethernet using socket communication.



Power: 5 V / 2 A (USB Type-C)  
Operates with a mobile battery



Tracks motion at speeds up to 5 m/s

# Conclusion

## □ LiDAR-based SLAM and localization

- plain\_slam\_ros2 [https://github.com/NaokiAkai/plain\\_slam\\_ros2](https://github.com/NaokiAkai/plain_slam_ros2)
- LIO, SLAM, and localization

## □ Fusion with IMU, camera, and GNSS

- Tightly coupled fusion
- Monocular metric localization using Gaussian splatting
- Trajectory estimation in GNSS coordinates