AA 274A
Principles of Robot Autonomy I

Open-source Automated Driving Stack „Autoware Hands-on“

Stanford University

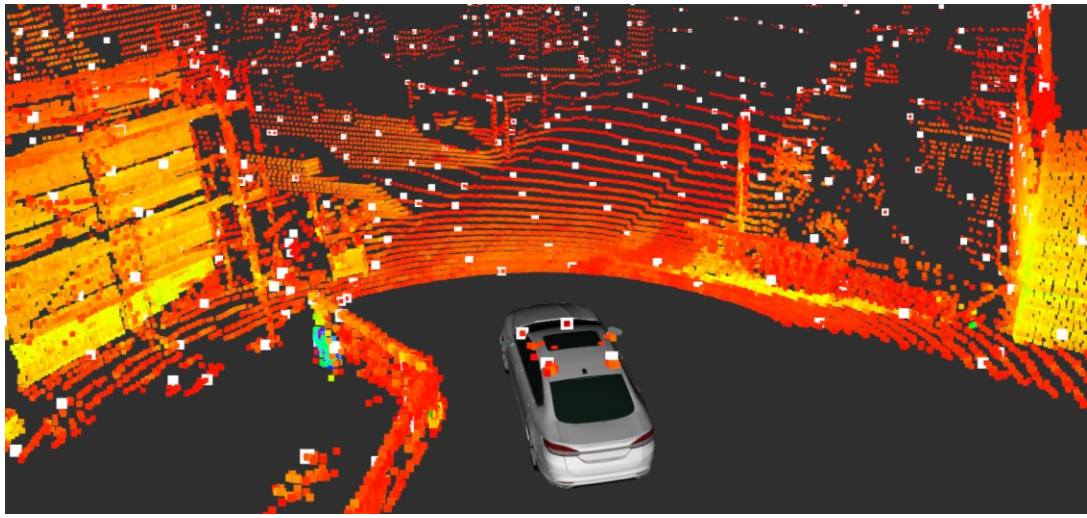
ASL
Autonomous Systems Lab

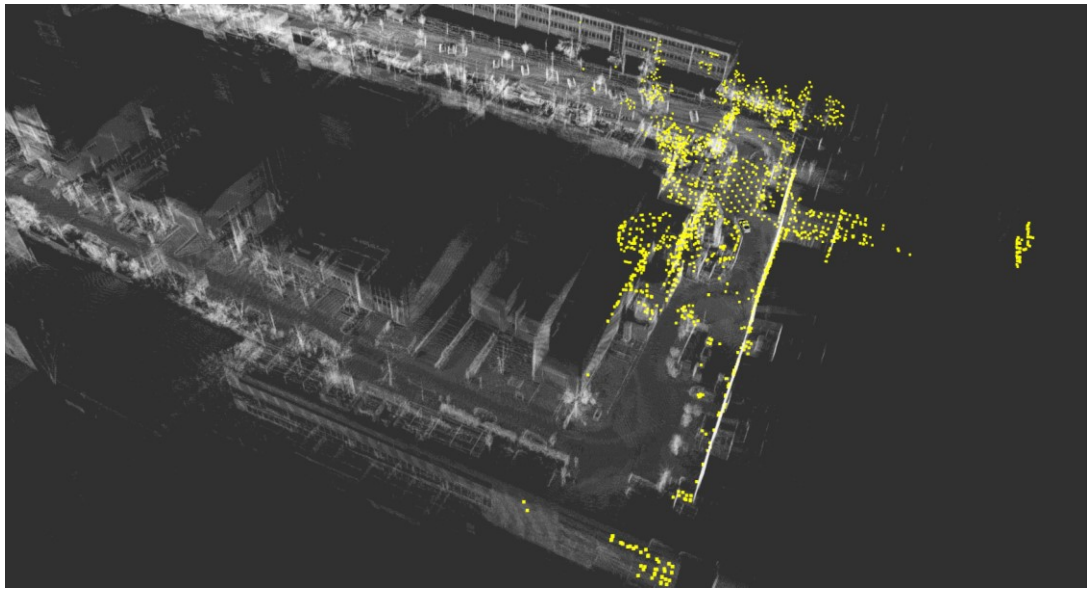
Agenda

- Localization
- Path planning

Localization pipeline

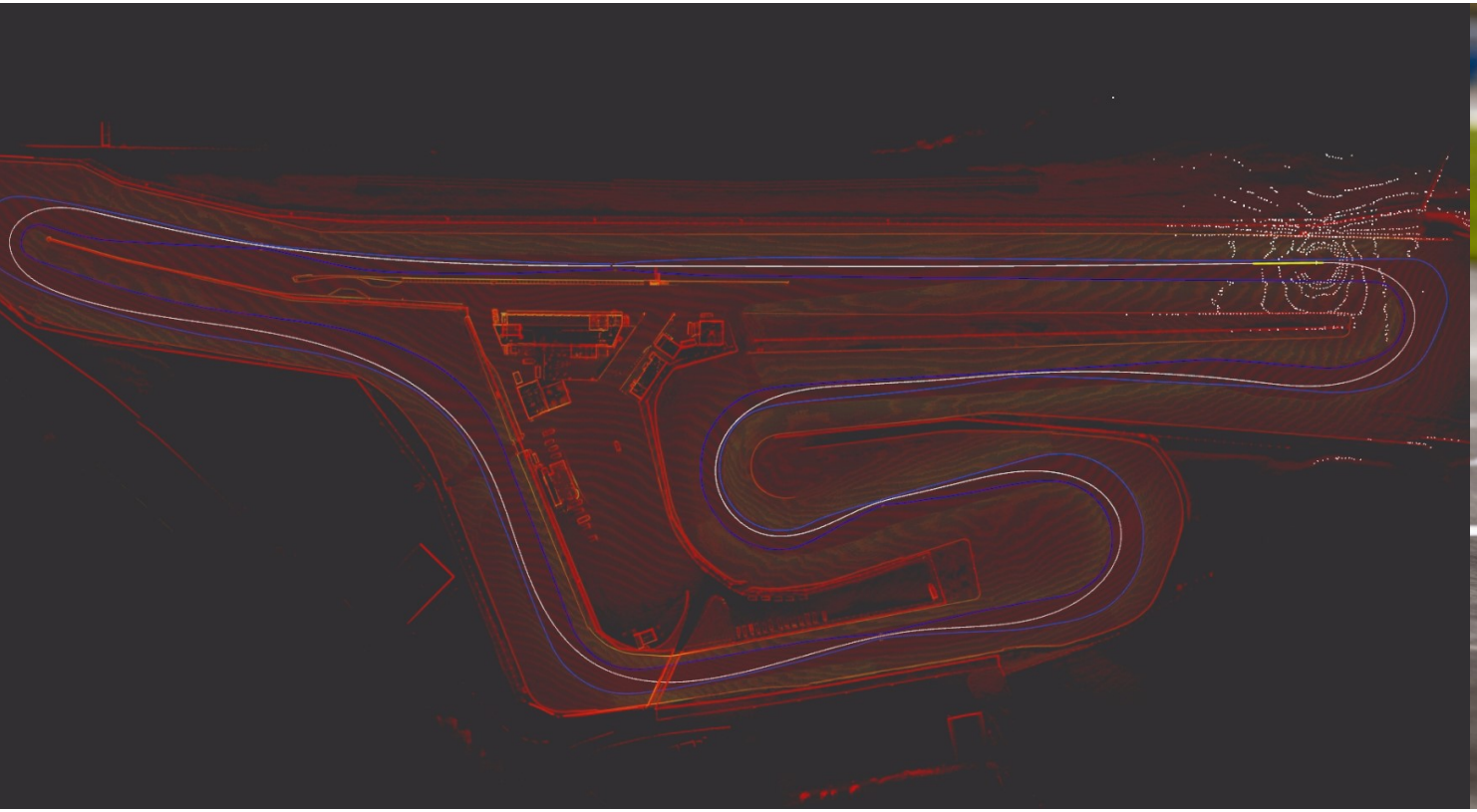
- Map loader **[points_map_loader]**
  - PCD loader from map
- Voxel Grid Filter **[voxel_grid_filter]**
  - Downsampling lidar data
  - Leaf size: 2m (60MB/s → ~1MB/s)
- Lidar based localization **[ndt_matching]**
  - NDT matching
  - Input: /filtered_points, /vehicle/twist
  - Output: /ndt_pose
- EKF Localization Fusion **[ekf_localizer]**
  - Input: /ndt_pose, /vehicle/twist
  - Output: /ekf_pose_with_covariance





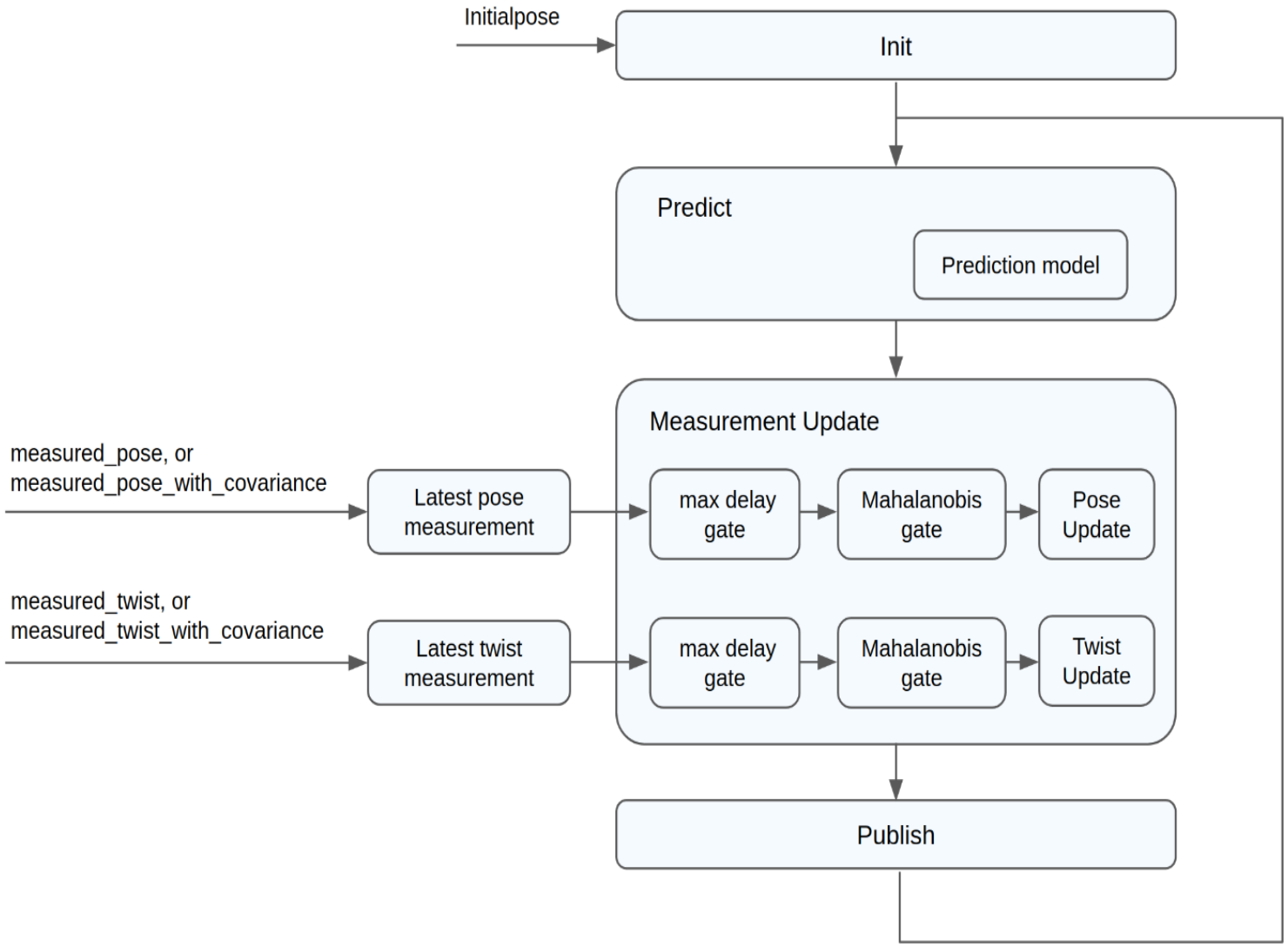
https://gitlab.com/autowarefoundation/autoware.ai/core_perception/tree/master/lidar_localizer/nodes/ndt_matching

Localization / Roborace / Croix-en-Ternois



Autonomous Racing Graz

EKF L


Initialpose
Init
Predict
Prediction model
Measurement Update
measured_pose, or
measured_pose_with_covariance
Latest pose
measurement
max delay
gate
Mahalanobis
gate
Pose
Update
measured_twist, or
measured_twist_with_covariance
Latest twist
measurement
max delay
gate
Mahalanobis
gate
Twist
Update
Publish

EKF Localizer / Interface

Input:

/devbot/twist	… twist from Devbot (velocity, yaw_rate)
/ndt_pose	… position from localization (lidar or noisy GPS data)

Output:

/ekf_pose_with_covariance	… output from the EKF for localization

Ground truth: /gps_local/pose

Localization modes

1) GPS based localization with noisy gps data:

localization_pose: /ndt_pose (gps_pose + noise)

lidar_localization_active: false

2) Lidar based localization (localization running online)

Localization_pose: /ndt_pose (ndt_localization)

lidar_localization_active: true

Localization mode: **vifware_launch/launch/localization_devbot/Devbot_localization.launch**

EKF localizer setting: **vifware_launch/launch/localization_devbot/ekf_localizer.launch**

After every change in a launch file you need to rebuild the source!

Tasks

1) Localization only with Odometry
2) Localization with GPS without noise + Odometry

    stddev_x_y: 0, mu_x_y: 0

3) Localization with GPS with noise + Odometry

    stddev_x_y: 1, mu_x_y: 0

4) Localization with GPS with noise incl. bias + Odometry

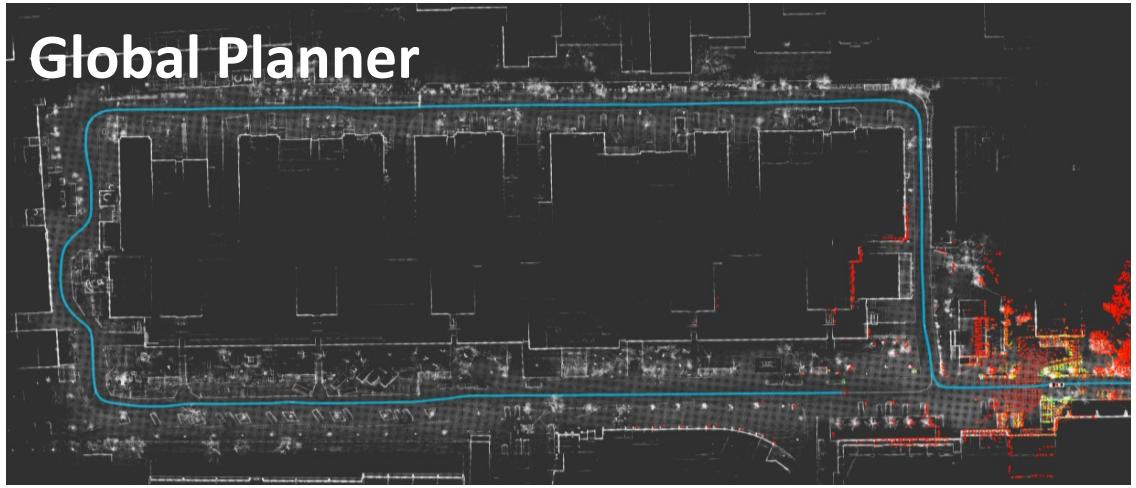
    stddev_x_y: 1, mu_x_y: 1

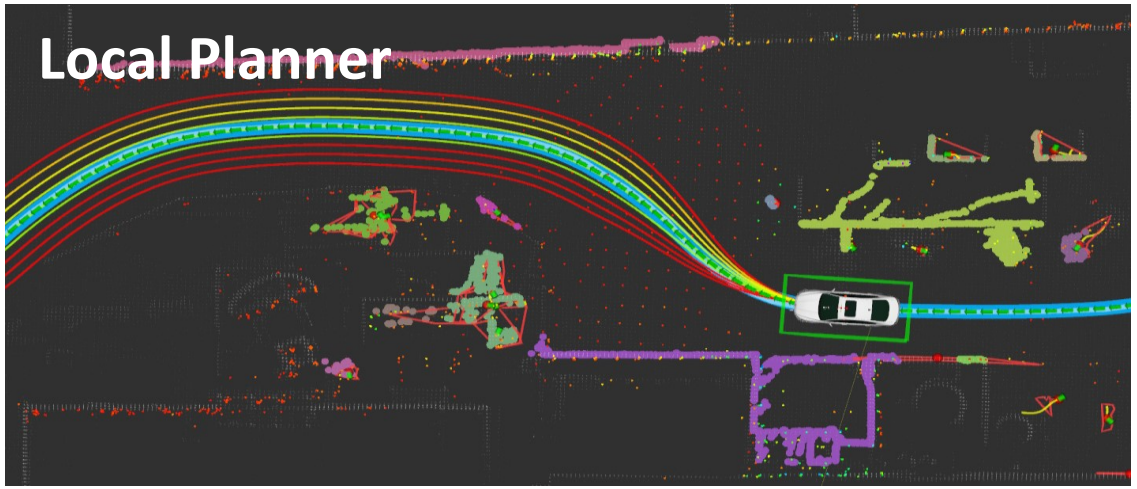
5) Localization with lidar + Odometry

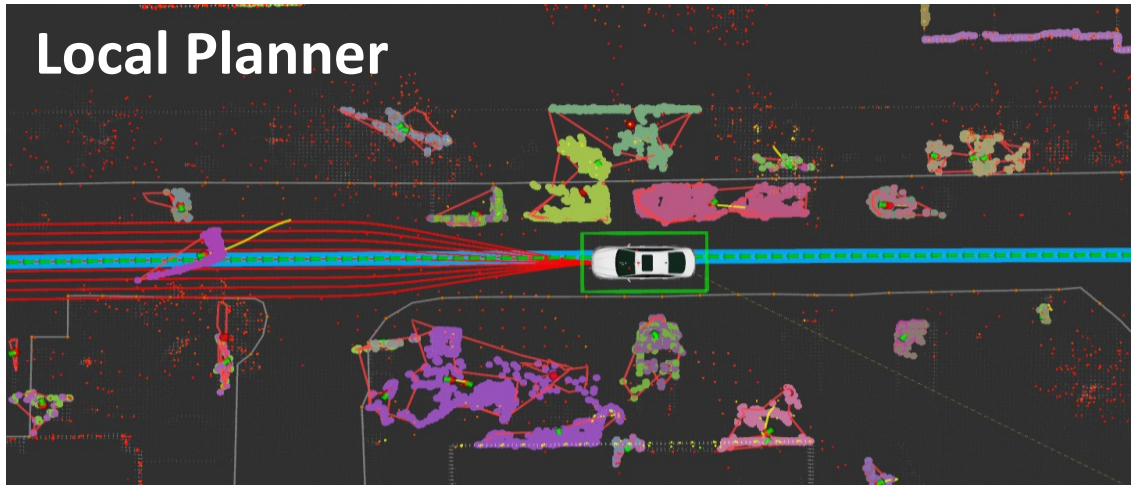
    →parameter tuning (lidar pose has an unknown time delay and unknown noise)

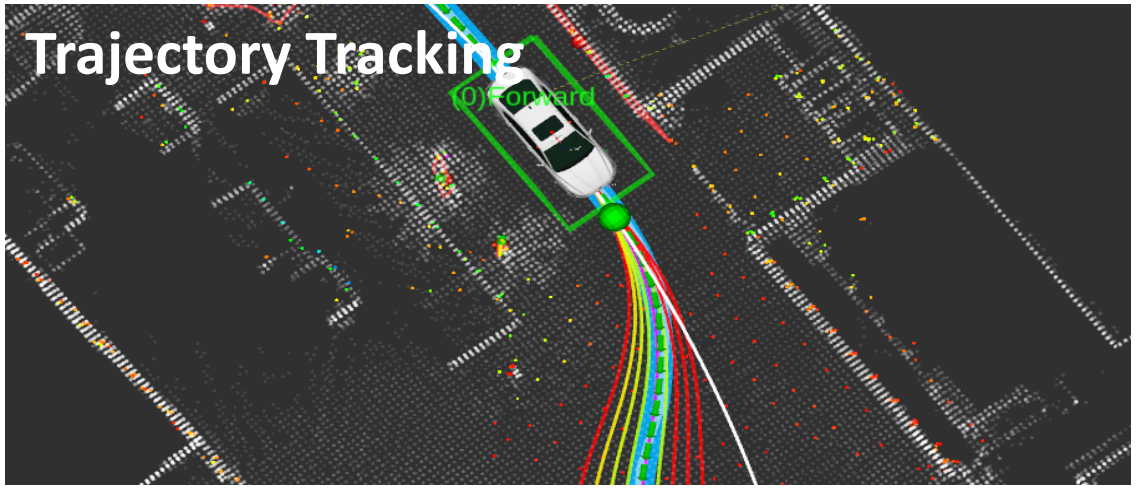
    Goal: the ekf_pose should match the gps_local/pose

Path Planning / Trajectory Tracking









Path planning

- Global planner **[op_global_planner]**
- Local planner **[op_trajectory_generator, op_motion_predictor, op_trajectory_evaluator, op_behavior_selector]**
  - Input: /tracked_objects, /global_path
  - Output: /final_waypoints
- Trajectory Tracking **[pure_pursuit or mpc_follower, twist_filter]**
  - Input: /final_waypoints
  - Output: /twist_cmd
- Autoware Simulator **[wf_simulator]**
  - Input: /twist_cmd
  - Output: /simulated_objects

Thanks for your attention!
Questions?

Daniel Watzenig, Markus Schratter
daniel.watzenig@v2c2.at
markus.schratter@v2c2.at

Stanford University

Autonomous Systems Lab