



# EKF Localization

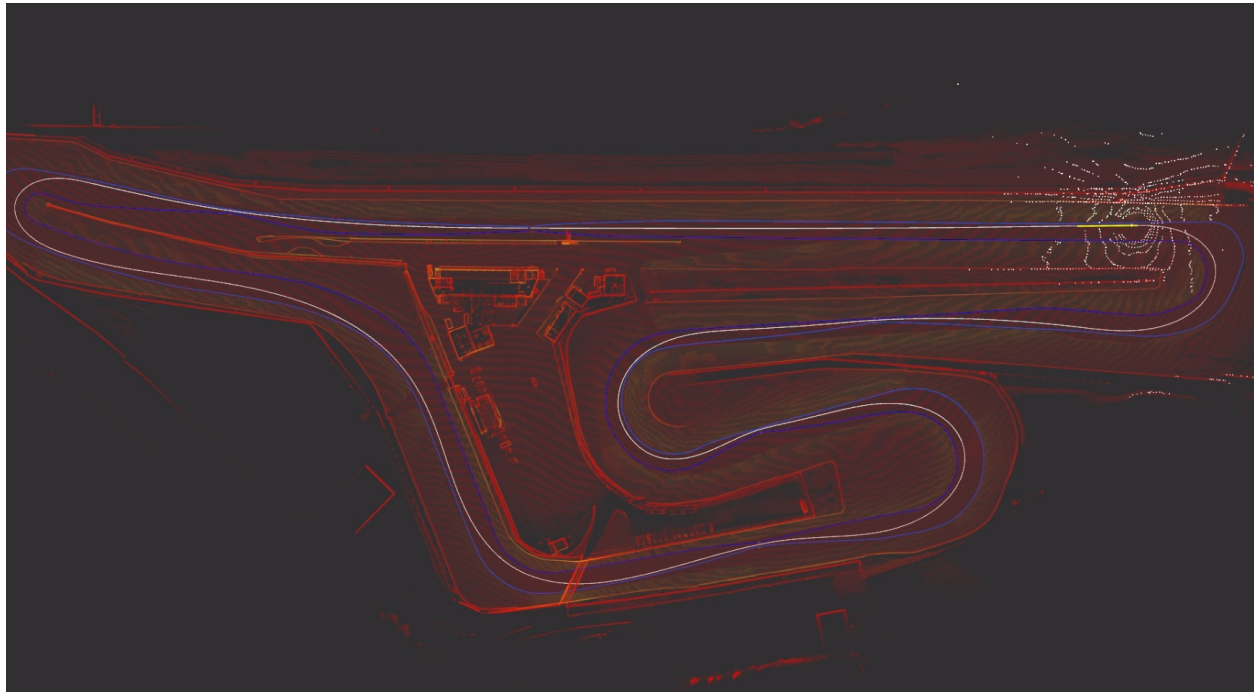
Open-source Automated Driving Stack „Autoware Hands-on“

[https://github.com/virtual-vehicle-research/aa274\\_autoware\\_ws](https://github.com/virtual-vehicle-research/aa274_autoware_ws)

# Agenda

- Demonstration
  - Autoware: Autonomous Driving Stack
  - Autonomous Racing: Localization / Sensor Fusion / Extended Kalman Filter

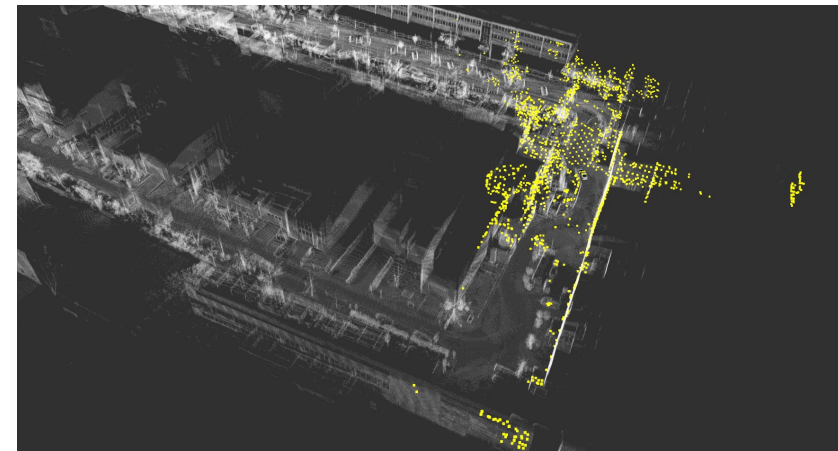
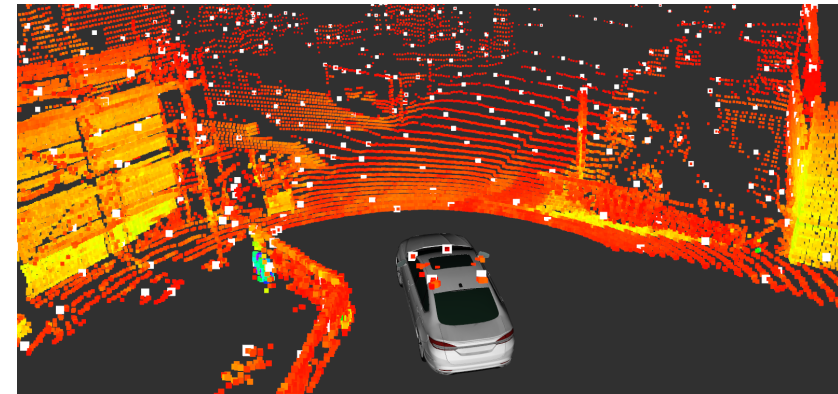
# Localization / Roborace / Croix-en-Ternois



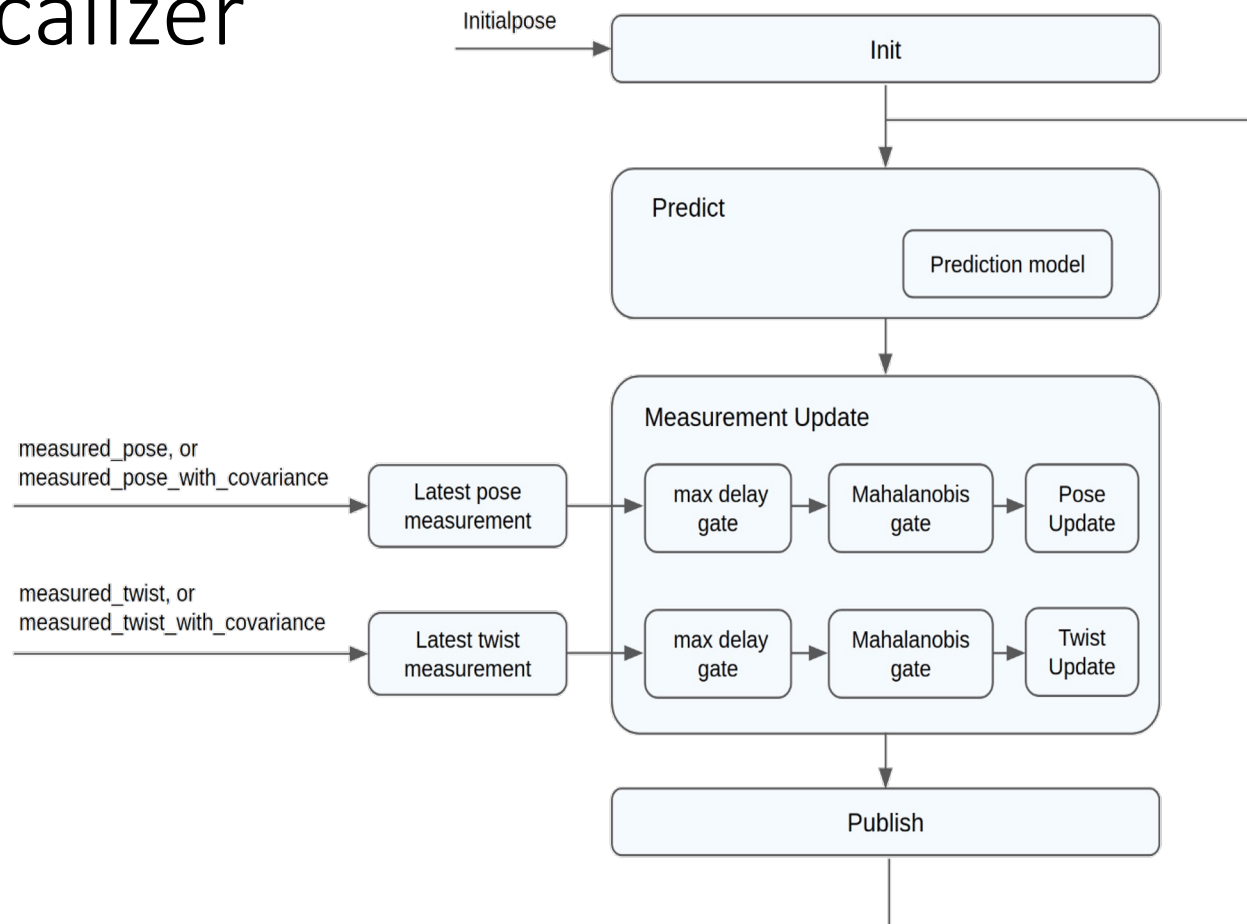
## Autonomous Racing Graz

# 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: /localization/downsample/pointcloud, /devbot/odom
  - Output: /localization/pose\_estimator/pose
- EKF Localization Fusion [**ekf\_localizer**]
  - Input: /localization/pose\_estimator/pose, /devbot /twist
  - Output: /localization/pose\_twist\_fusion\_filter/pose\_with\_covariance



# EKF Localizer



# EKF Localizer / Interface

## Input:

/devbot/twist                      ... twist from Devbot (velocity, yaw\_rate)  
/localization/pose\_estimator/pose    ... position from localization (lidar or noisy GPS data)

## Output:

/localization/pose\_twist\_fusion\_filter/pose                      ... localization output

## Ground truth:

/devbot/pose

# Localization modes

## 1) GPS based localization with noisy gps data:

```
/localization/pose_estimator/pose: RTK-GPS + noise
```

```
roslaunch arg_demos arg_demo_localization.launch
```

[GPS noise](#)

## 2) Lidar based localization

```
/localization/pose_estimator/pose: NDT-localization
```

```
roslaunch arg_demos arg_demo_localization.launch lidar_localization:=true
```

[Extended Kalman Filter Settings](#)

# Task 1: Localization only with Odometry

EKF input:

```
/devbot/twist (velocity, yaw_rate)
```

What do we expect?



## Task 2: Localization with GPS

EKF input:

<code>/devbot/twist</code>	(velocity, yaw_rate)
<code>/localization/pose_estimator/pose</code>	(GPS ground truth)

What do we expect?

# Task 3: Localization with GPS + Noise

EKF input:

`/devbot/twist` (velocity, yaw\_rate)

`/localization/pose_estimator/pose` (GPS + noise)

Noise:  $N(\mu, \sigma^2)$

What do we expect?

# Task 4: Localization with Lidar

EKF input:

<code>/devbot/twist</code>	(velocity, yaw_rate)
<code>/localization/pose_estimator/pose</code>	(NDT localization)

Issues:

- Processing time
- Unknown localization quality
- Alignment GPS - Lidar map

What do we expect?



Thanks for your attention!  
Questions?

Prof. Daniel Watzenig, Markus Schratter

[daniel.watzenig@v2c2.at](mailto:daniel.watzenig@v2c2.at)

[markus.schratter@v2c2.at](mailto:markus.schratter@v2c2.at)