DUFOMap: Efficient Dynamic Awareness Mapping

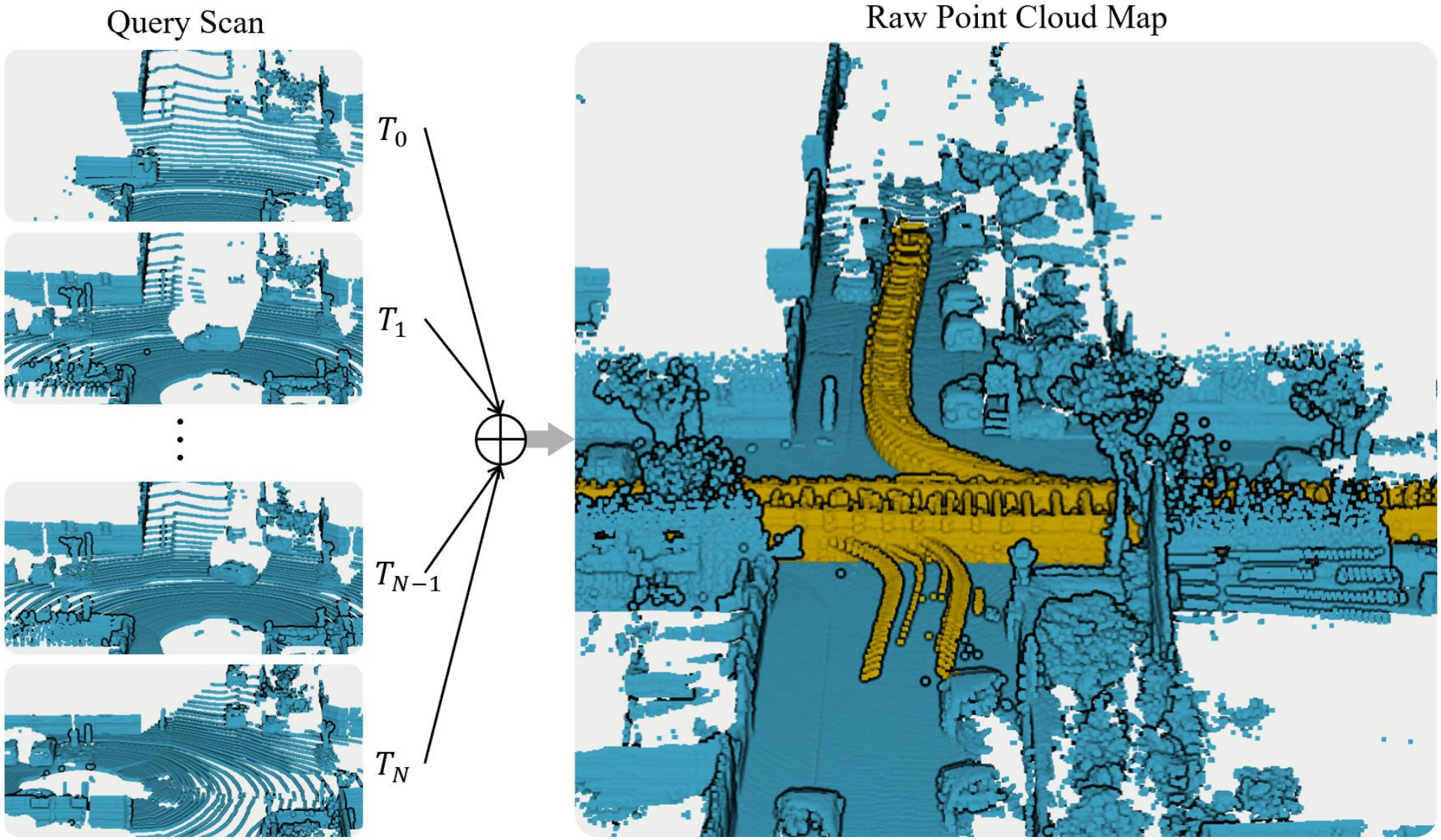
Daniel Duberg*, Qingwen Zhang*✉, Mingkai Jia, Patric Jensfelt

*co-first author, ✉corresponding author

Published at IEEE Robotics and Automation Letters

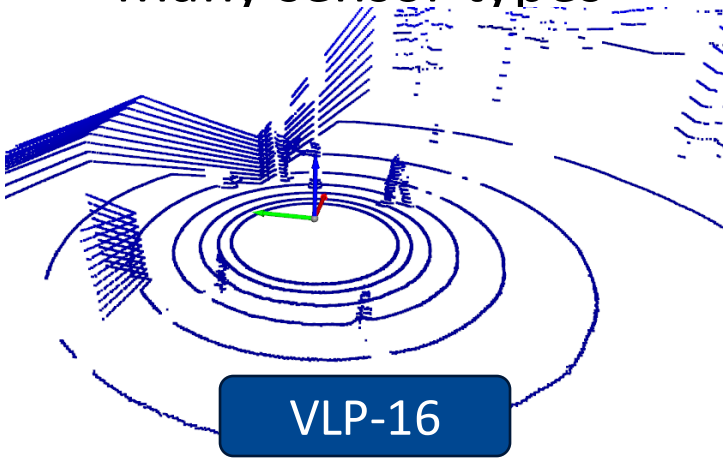
DOI: 10.1109/LRA.2024.3387658

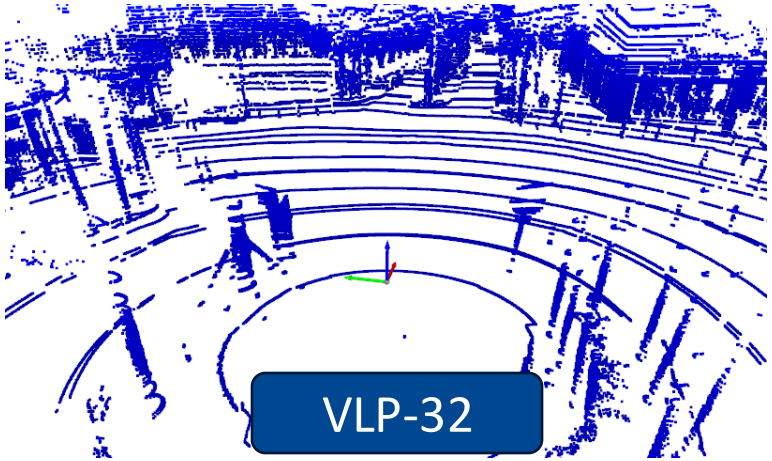
Introduction

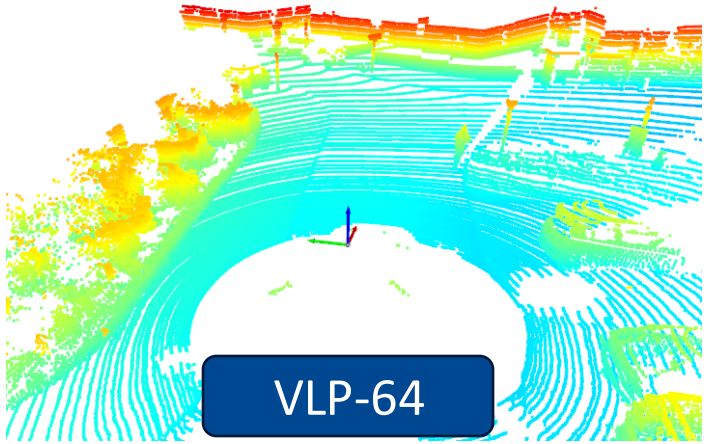


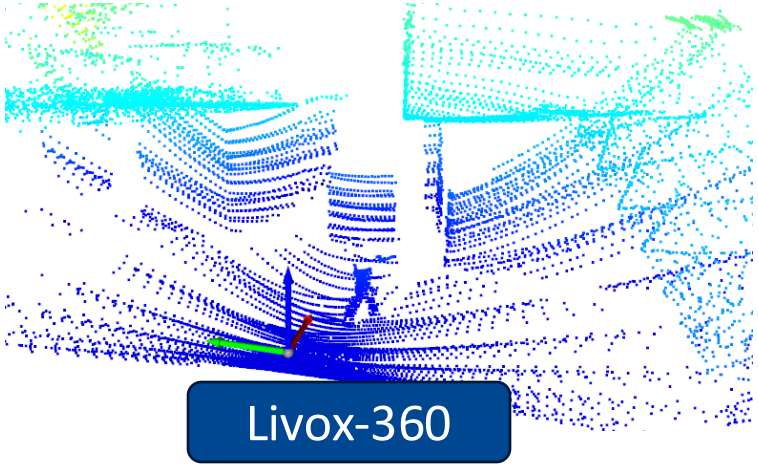
Motivation

- Many sensor types



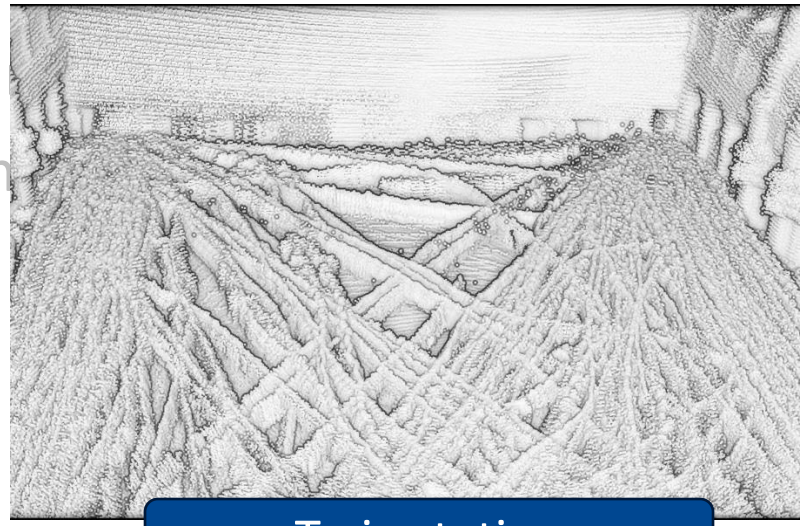




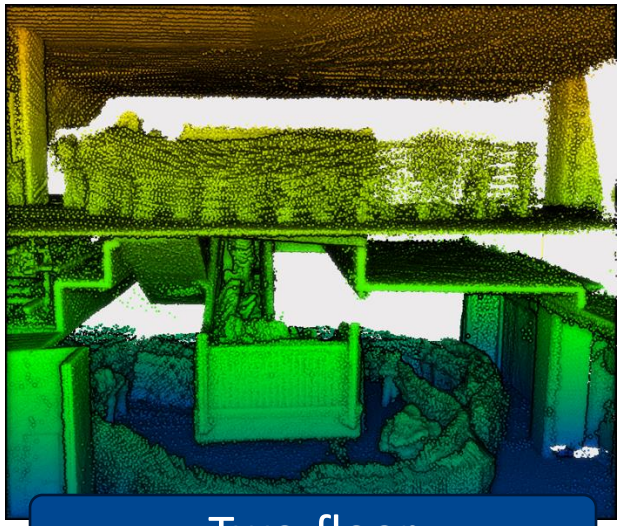


Motivation

- Many sensor types
- Different scenarios
- No para
- Real-tim

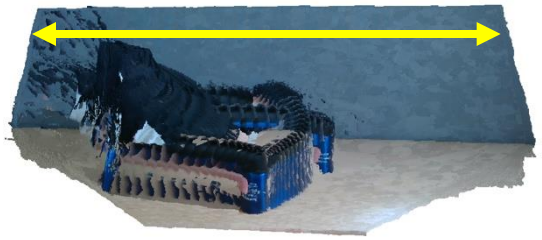


Train station



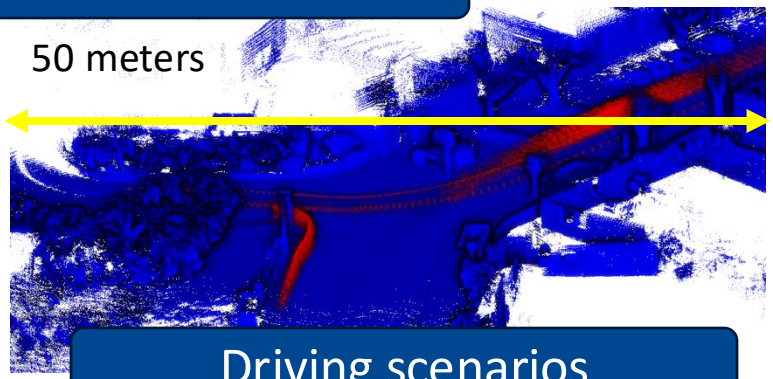
Two-floor

2 meters



Small-scale

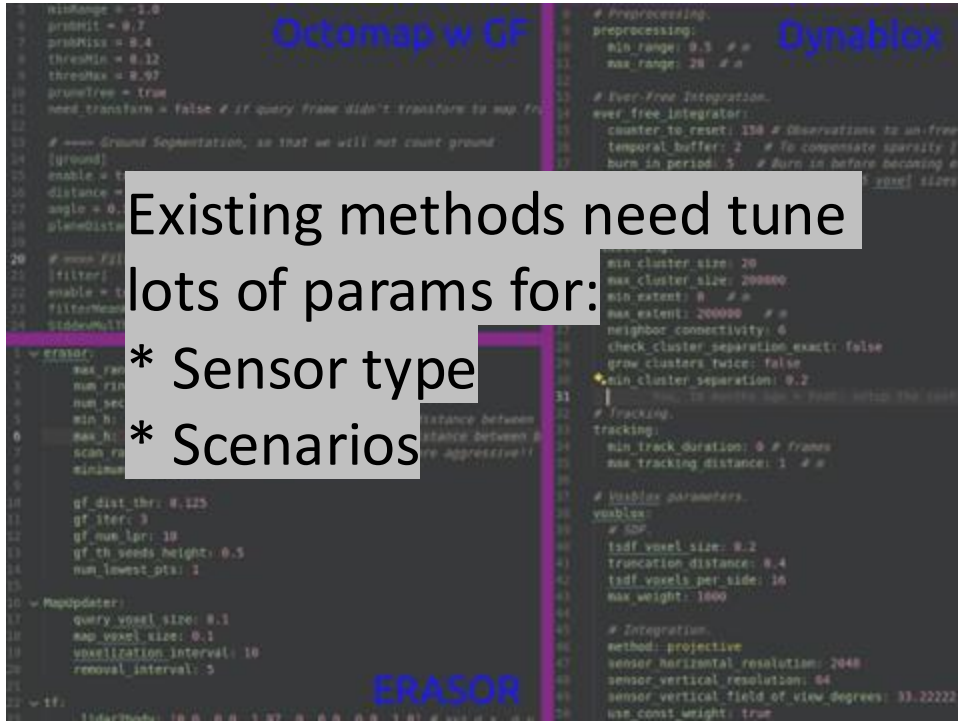
50 meters

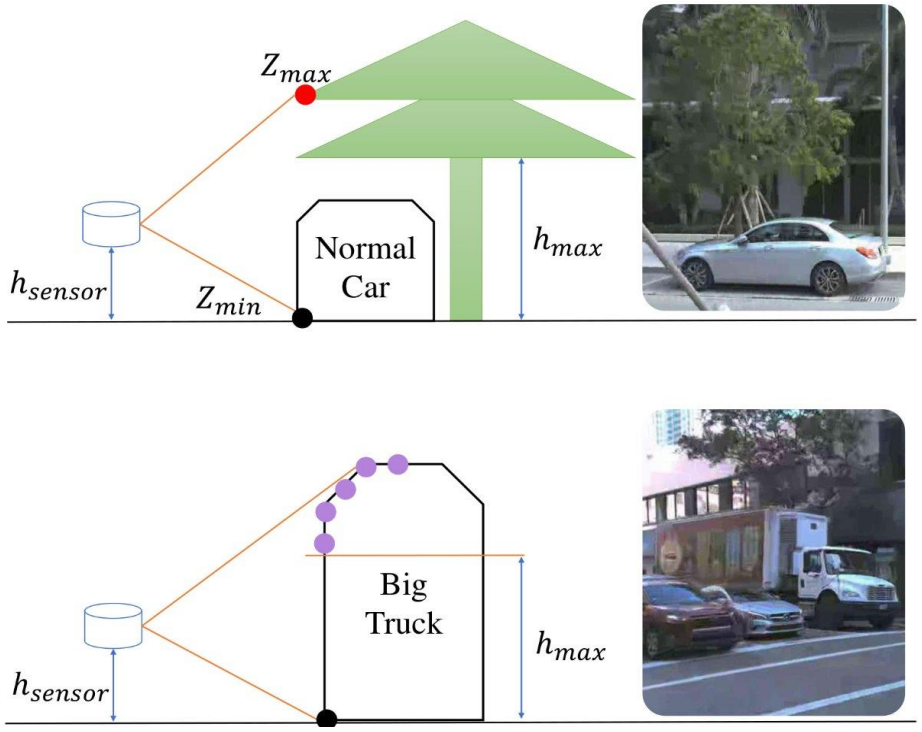


Driving scenarios

Motivation

- Many sensor types
- Different scenarios
- No parameter tuning
- Real-time operation





Motivation

- Many sensor types
- Different scenarios
- No parameter tuning
- Real-time operation

Faster to get result

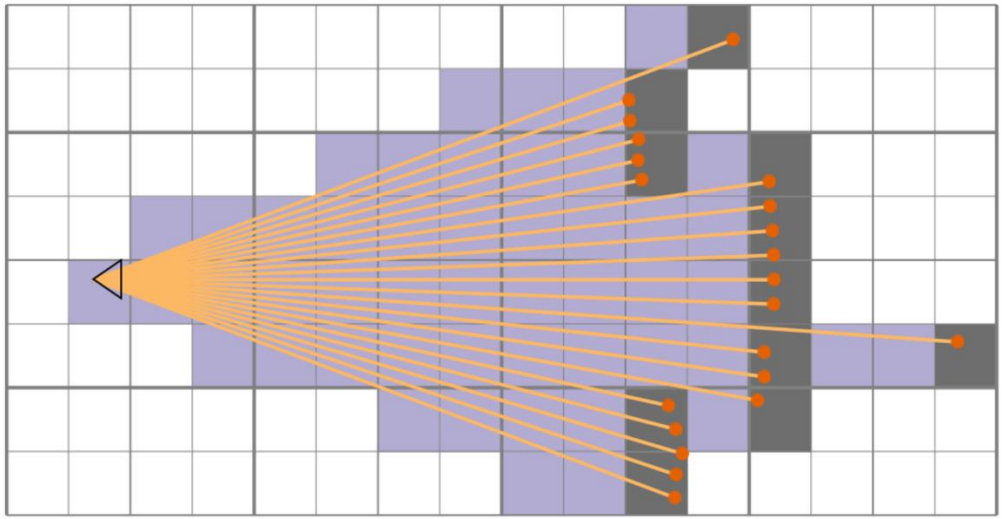
Method	Time (4000+ frames KITTI 01)
Octomap	214 mins ≈ 3.5 hours
Dynablox	9.4 mins
DUFOMap (Ours)	**4 mins**

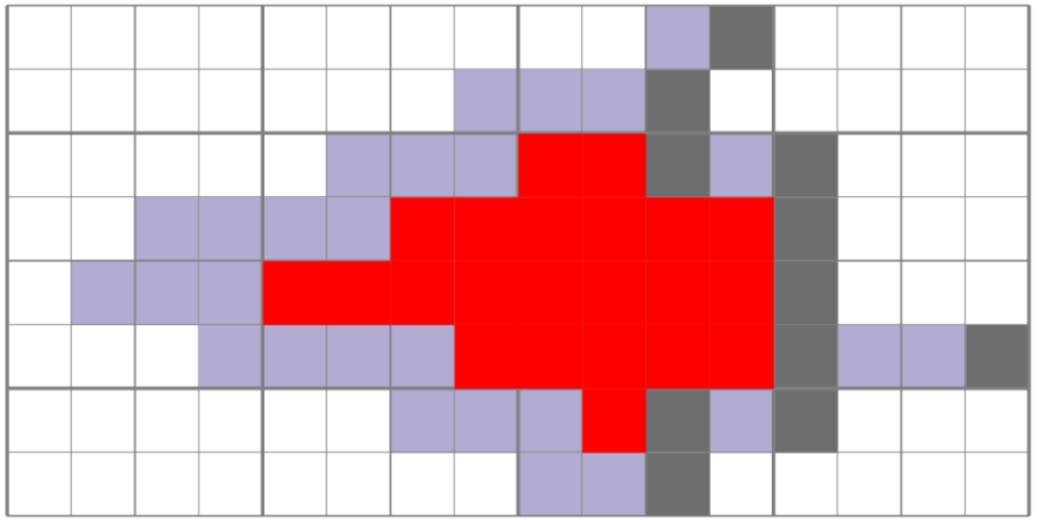
Online Application

Method

Method

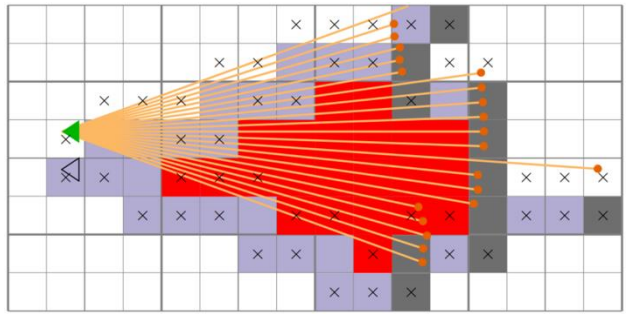
- Classify void regions
  - Ray-casting

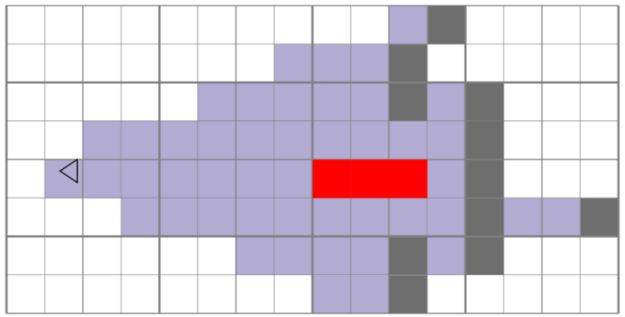


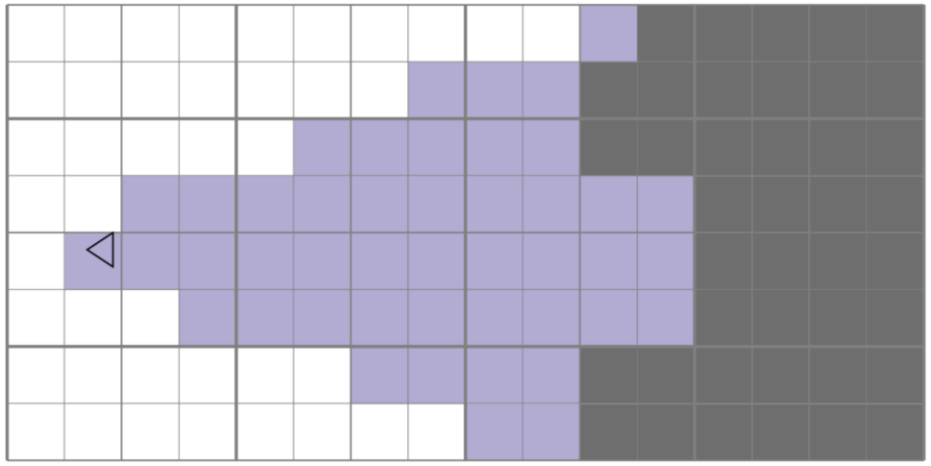


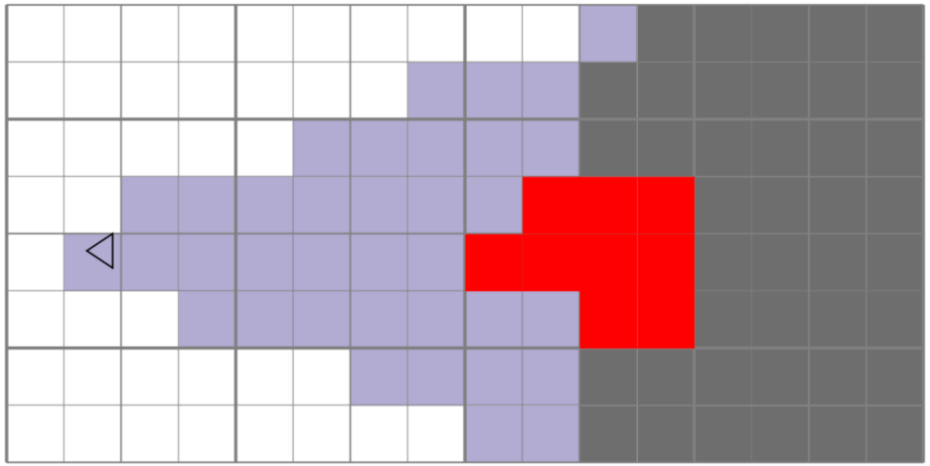
Method

- Classify void regions
  - Ray-casting
- Real-world scenarios
  - Sensor noise
  - Localization error
- Extend occupied regions







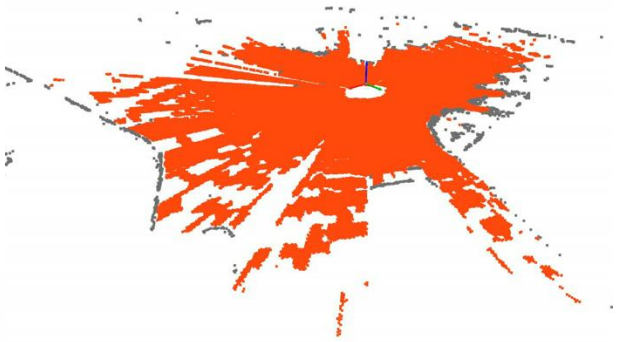


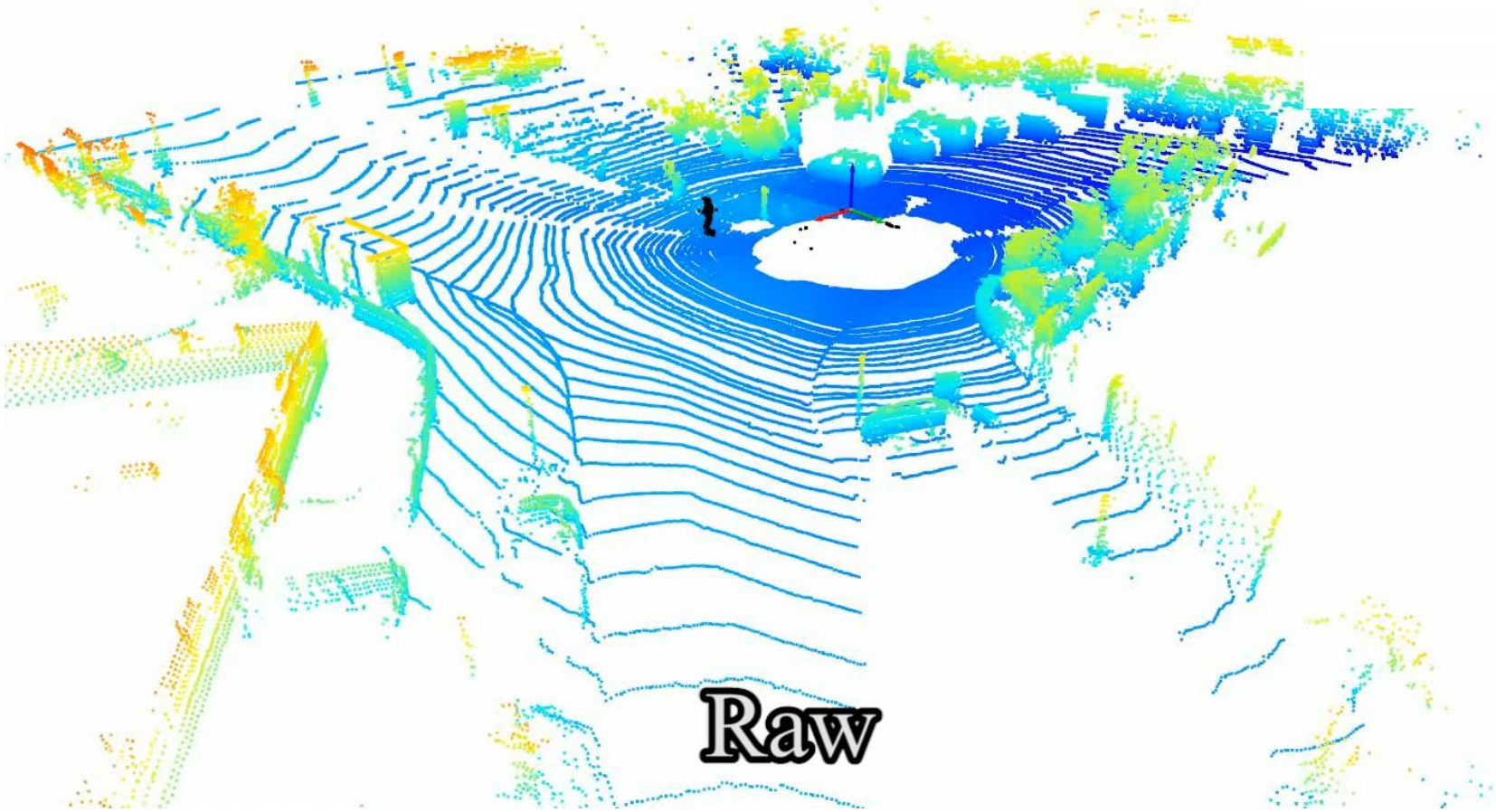
Method Steps

1. Frame accumulates
2. A void sub/full map
3. Point fall in

Method Steps

1. Frame accumulates
2. A void sub/full map
3. Point fall into void region = Dynamic





Experiment Results

w. same parameters always

Quantitative Results (DUFOMap w. same parameters always)

64 channels

32 channels

16 channels

	KITTI small town (00)			KITTI highway (01)			Argoverse 2 big city			Semi-indoor		
Methods	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑	SA ↑	DA ↑	AA ↑
Removert [8]	99.44	41.53	64.26	97.81	39.56	62.20	98.97	31.16	55.53	99.96	12.15	34.85
ERASOR [9]	66.70	98.54	81.07	98.12	90.94	<u>94.46</u>	77.51	99.18	87.68	94.90	66.26	79.30
OctoMap [16]	68.05	99.69	82.37	55.55	99.59	74.38	69.04	97.50	82.04	88.97	82.18	<u>85.51</u>
DUFOMap (Ours)	97.96	98.72	**98.34**	98.09	94.20	**96.12**	96.67	88.90	<u>92.70</u>	99.64	83.00	**90.94**
Dynablox [17]	96.76	90.68	93.67	96.33	68.01	80.94	96.08	92.87	**94.46**	98.81	36.49	60.05
DUFOMap* (Ours)	98.37	92.37	<u>95.31</u>	98.48	81.34	89.50	98.66	73.98	85.43	99.94	54.76	73.98

Autonomous Driving

Indoor robotics

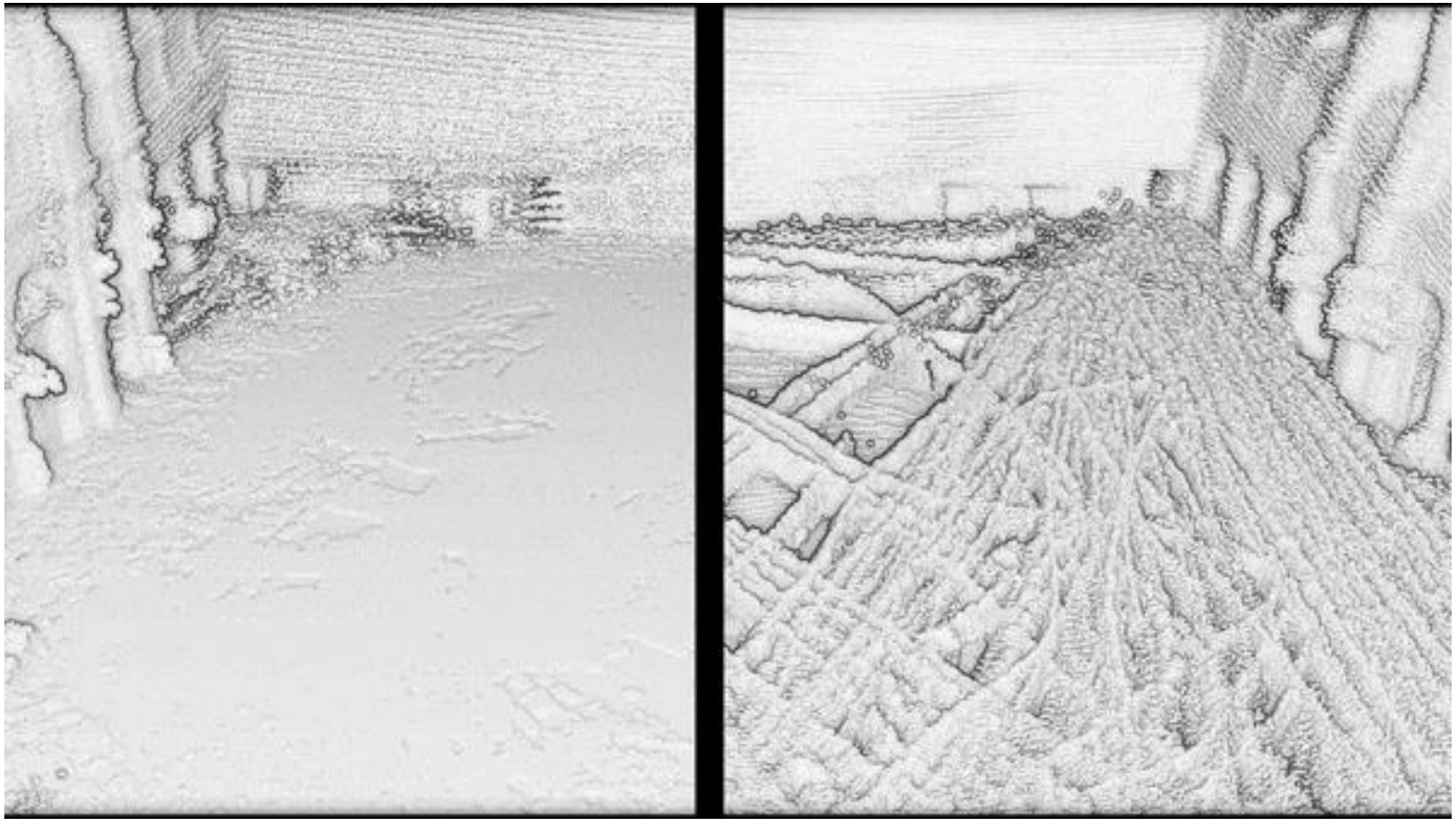
Quantitative Results (DUFOMap w. same parameters always)

TABLE II: Runtime comparison of different methods.

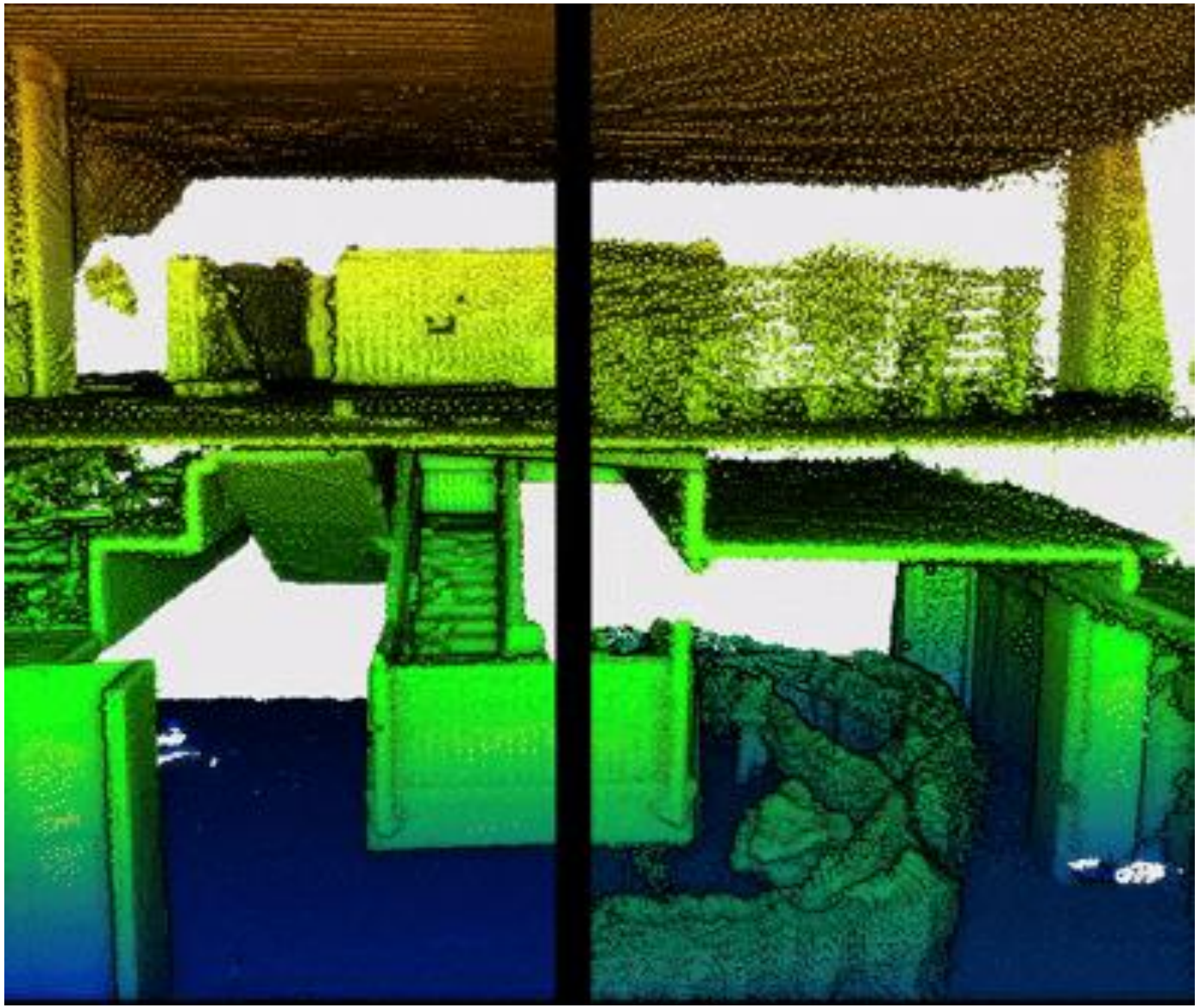
Methods	Run time per point cloud [s] ↓	
	KITTI highway	Semi-indoor
Removert [8]	0.134 ± 0.004	0.515 ± 0.024
ERASOR [9]	0.718 ± 0.039	0.064 ± 0.011
OctoMap [16]	2.981 ± 0.952	1.048 ± 0.256
Dynablox [17]	0.141 ± 0.022	0.046 ± 0.008
DUFOMap (Ours)	**0.062** ± **0.014**	**0.019** ± **0.003**

Fastest!

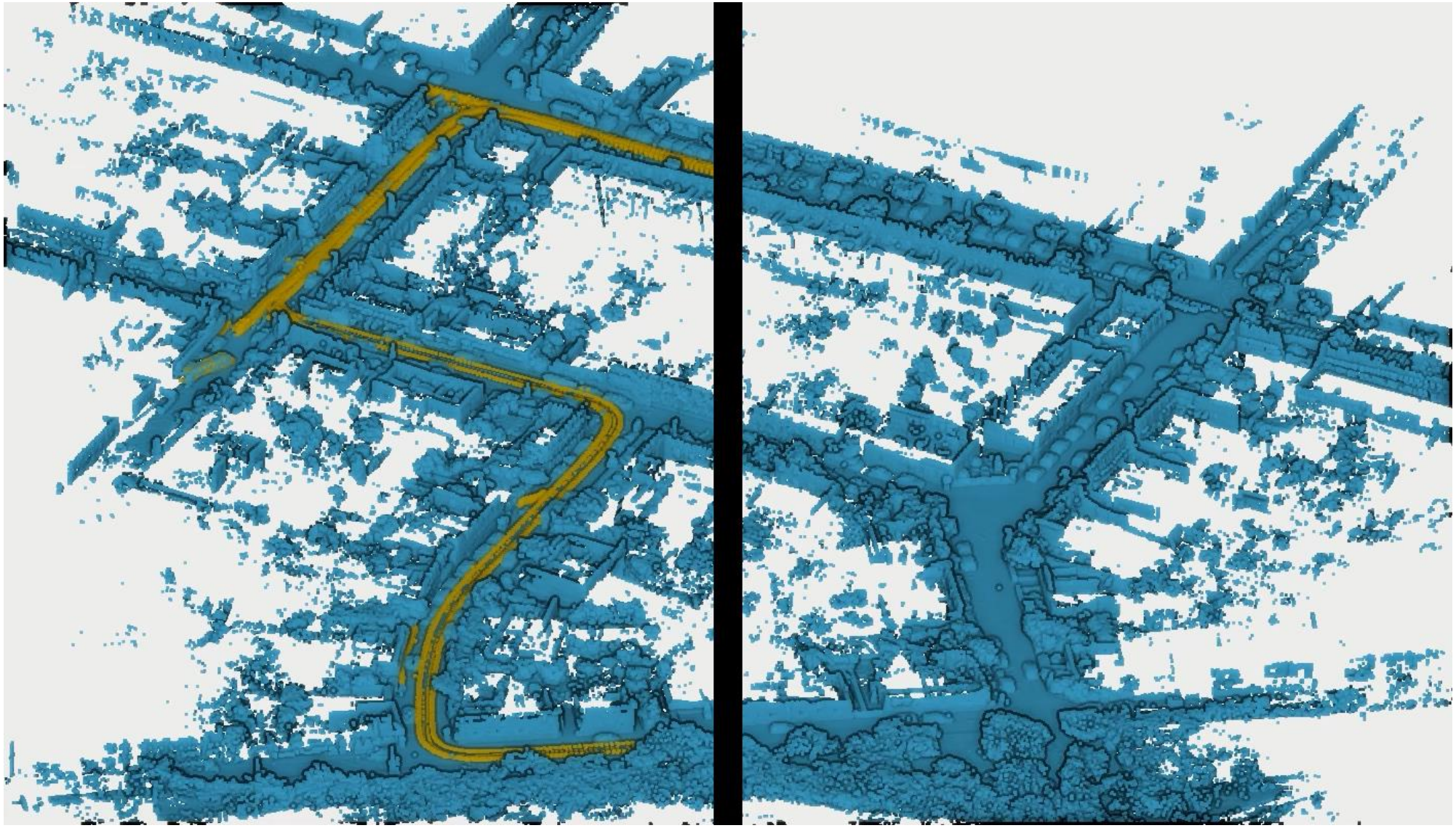
DUFOMap on Highly Dynamic VLP-128



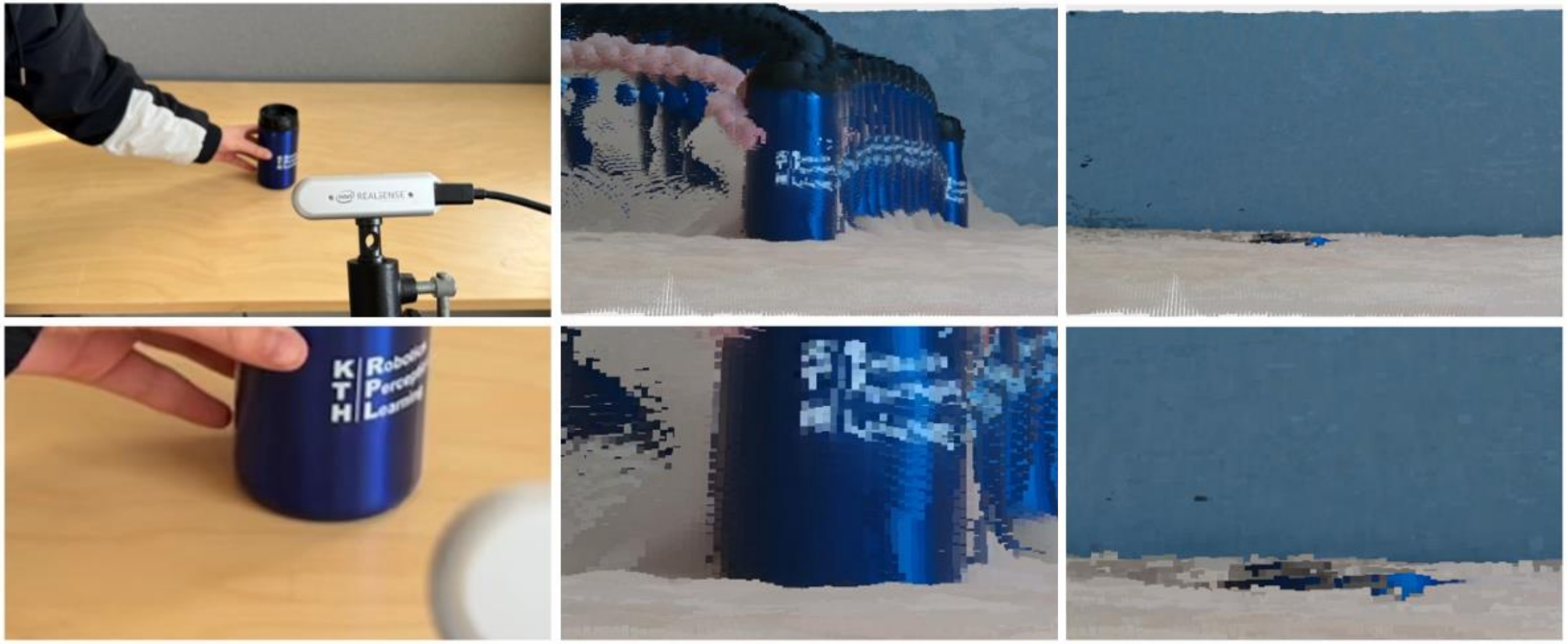
DUFOMap on Complex Structure Livox mid-360



DUFOMap on Driving Scenarios VLP-64



DUFOMap on Small-scale Table RGB-D (D435i)



DUFOMap on Open Campus leica-360

Conclusion

Our methods, **DUFOMap**, supports:

- Many sensor types
- Different scenarios
- No parameter tuning
- Real-time operation

Thanks for watching!

https://kth-rpl.github.io/**dufomap**

Follow us on **GitHub: KTH-RPL**

Scan me

DUFOMap and ***ALL*** **methods** we compared with.