SLICT: Multi-Input Multi-Scale **S**urfel-Based **L**idar-**I**nertial **C**ontinuous-**T**ime Odometry and Mapping

Thien-Minh Nguyen, Daniel Duberg, Patric Jensfelt

January 2023

Contents

Introduction

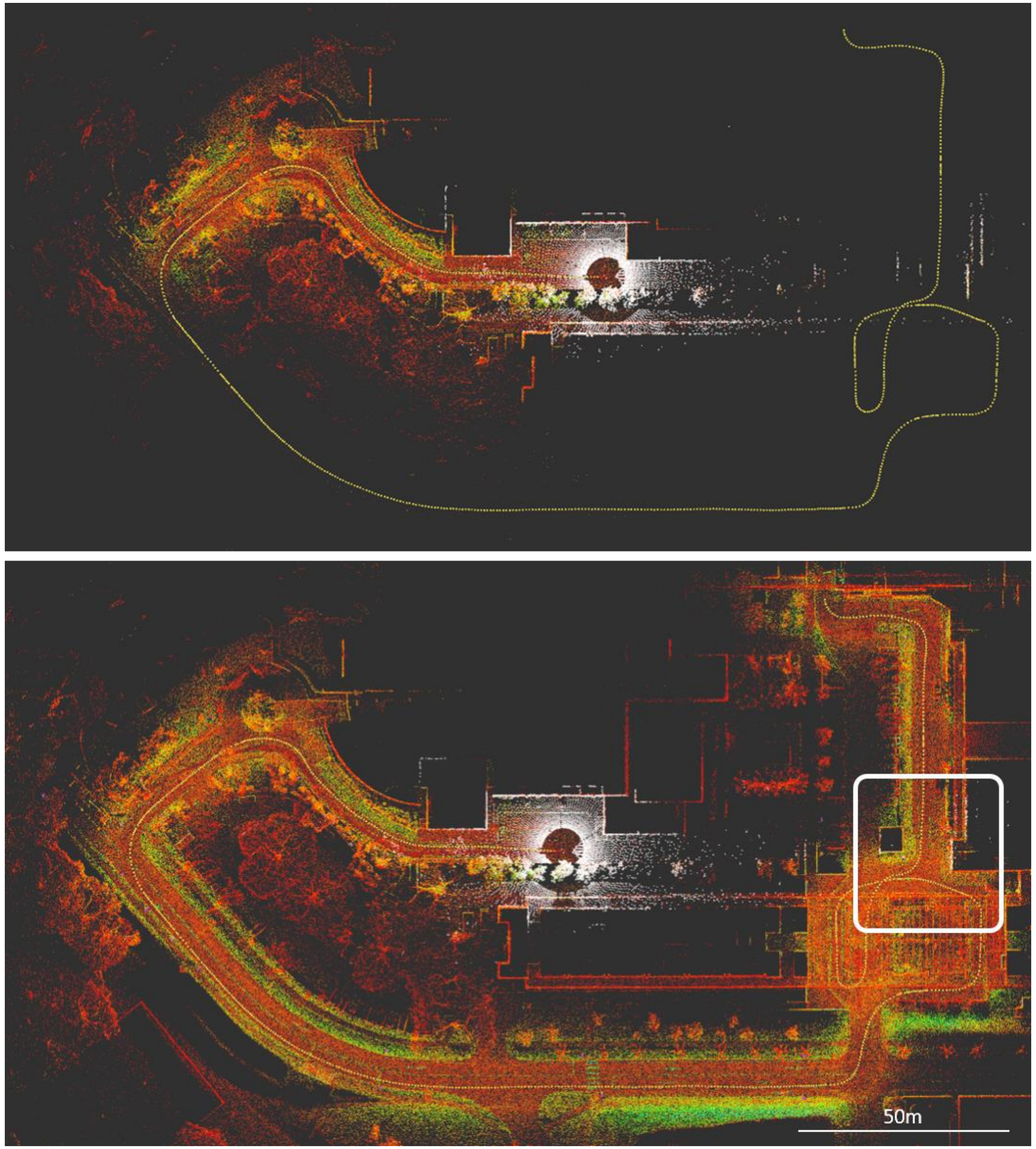
- ❑ Issue 1:

  <u>A bottleneck in mapping framework:</u>

  Traditional LOAM methods rely on **local map**, at **one voxel scale** → missing associations with features kept in earlier keyframes → drift can accumulate early.

  Why local map? (to bound the map building time)

  Why one scale only? (having to maintain multiple maps)

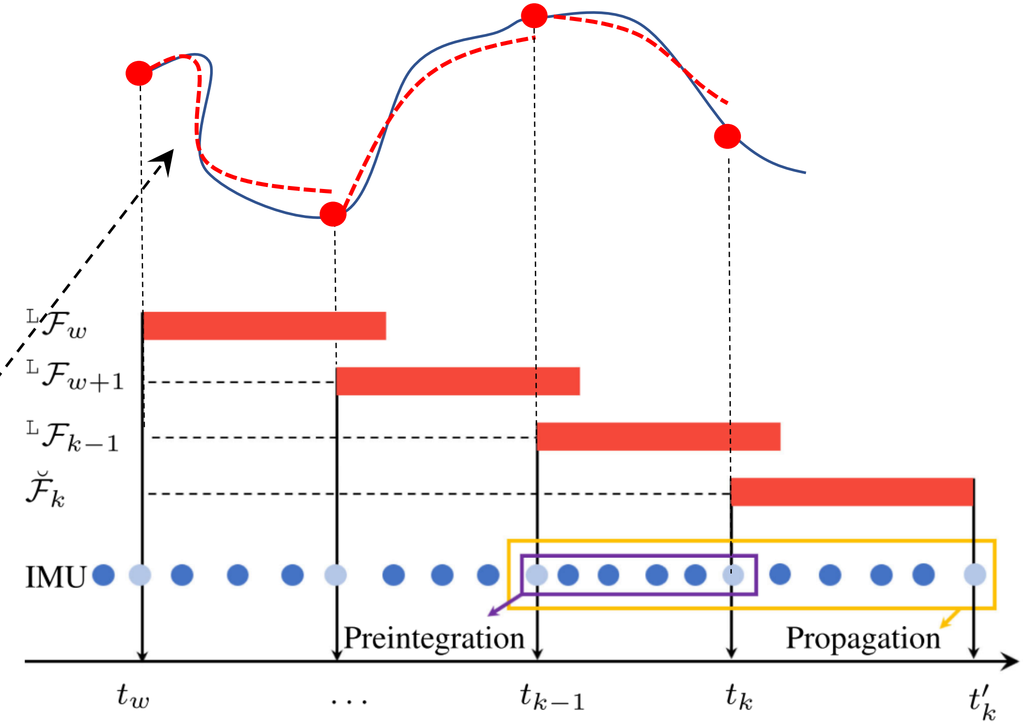


Introduction

❑ Issue 2:

<u>Discrete-time optimization issues:</u>

- The number of state estimates is equal to the number of pointclouds → Not representative of the dynamics.
- Factors are based on the lidar points that have been motion compensated by IMU propagation → propagation error on top of ranging error.



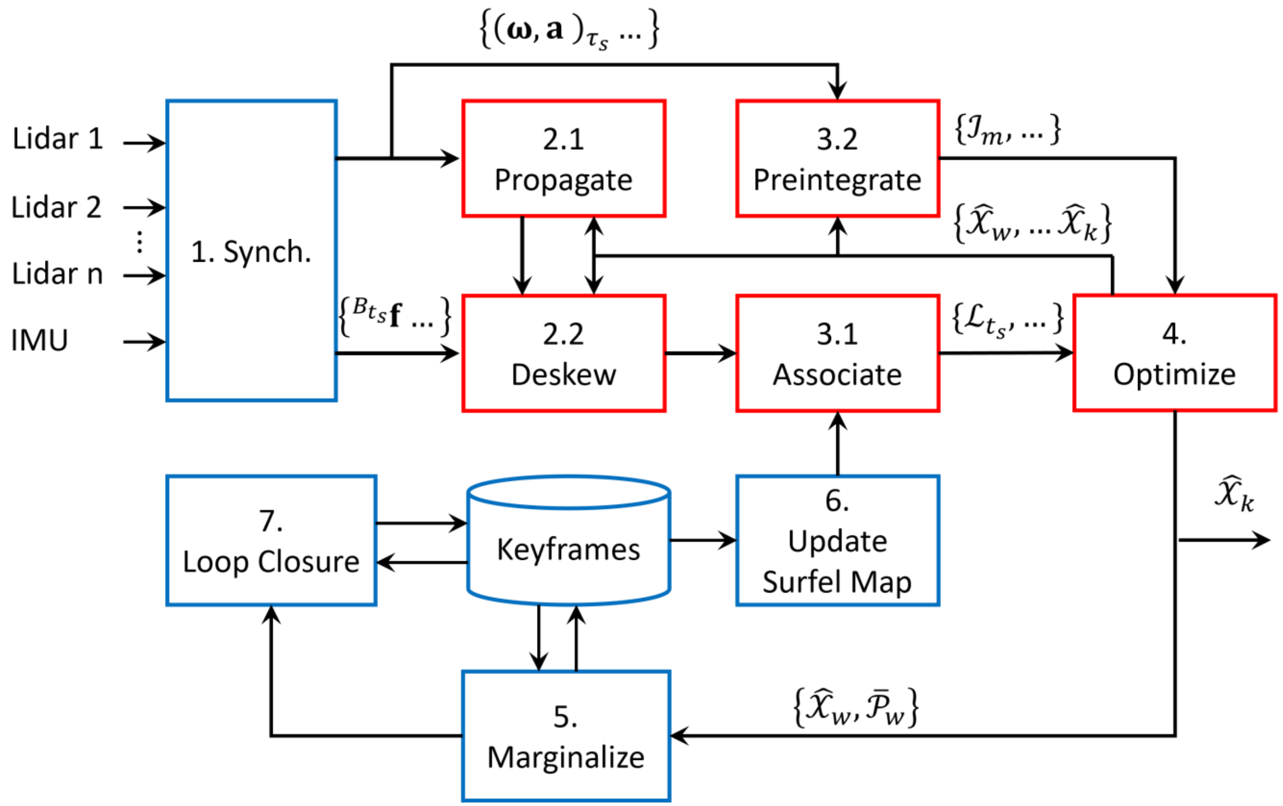
- Nguyen, Thien-Minh, Shenghai Yuan, Muqing Cao, Lyu Yang, Thien Hoang Nguyen, and Lihua Xie. "MILIOM: Tightly coupled multi-input lidar-inertia odometry and mapping." IEEE Robotics and Automation Letters 6, no. 3 (2021): 5573-5580.

Introduction

- SLICT Contributions:
  - UFOMap-based mapping framework:
    - ✓ Can be queried globally.
    - ✓ Can be incrementally updated.
    - ✓ Can query surfels of multiple scales simultaneously.
  - Joint optimization of continuous-time lidar factors (using raw lidar points) and IMU preintegration factors.
  - Supporting simple loop closure → full-fledged SLAM.
  - Open sourced: https://github.com/brytsknguyen/slict

Methodology

- System overview:
  - No feature extraction (~ direct method of FAST-LIO2).
  - Surfelization is done on the backend.
  - Blocks 2, 3, 4 can be done iteratively.



Methodology

- Octree-based Surfel Map:

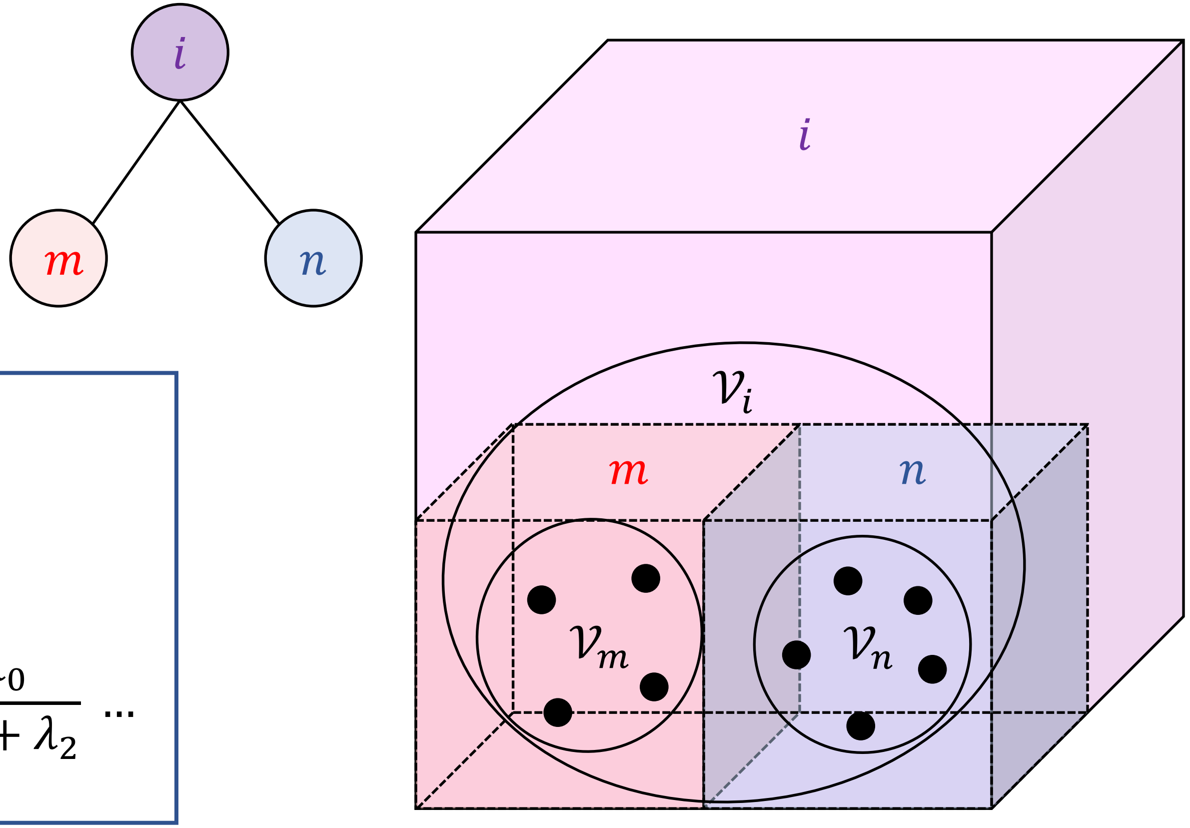
$\mathcal{V}_i = \{\mathbf{f}_1 \dots \mathbf{f}_{N_i}\}$

```
class myBeautifulSurfel
{
```

$$N_i \triangleq |\mathcal{V}_i|, \quad \mathbf{S}_i \triangleq \sum_{k=1}^{N_i} \mathbf{f}_i\,, \quad \mathbf{C}_i \triangleq \sum_{k=1}^{N_i} \mathbf{f}_i \mathbf{f}_i^\top - \frac{1}{N_i}\mathbf{S}_i\mathbf{S}_i^\top$$

$$\mu_i = \frac{1}{N_i}, \quad \Gamma_i = \frac{1}{N_i - 1}\mathbf{C}_\mathrm{i}, \quad \mathbf{n} = \nu_0, \quad p = 2\frac{\lambda_1 - \lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \quad \dots$$

```
}
```



- Multi-scale Surfel:

Given $\mathcal{V}_i = \mathcal{V}_m \cup \mathcal{V}_n$, $N_m, \mathbf{S}_m, \mathbf{C}_m, N_n, \mathbf{S}_n, \mathbf{C}_n$, the surfel of $\mathcal{V}_i$ can be computed by:

$\alpha \coloneqq 1/[N_m N_n (N_m + N_n)], \beta \coloneqq N_n S_m - N_m S_n$

$\mathbf{C}_i = \mathbf{C}_m + \mathbf{C}_n + \alpha\beta\beta^\top$

$N_i = N_m + N_n,$

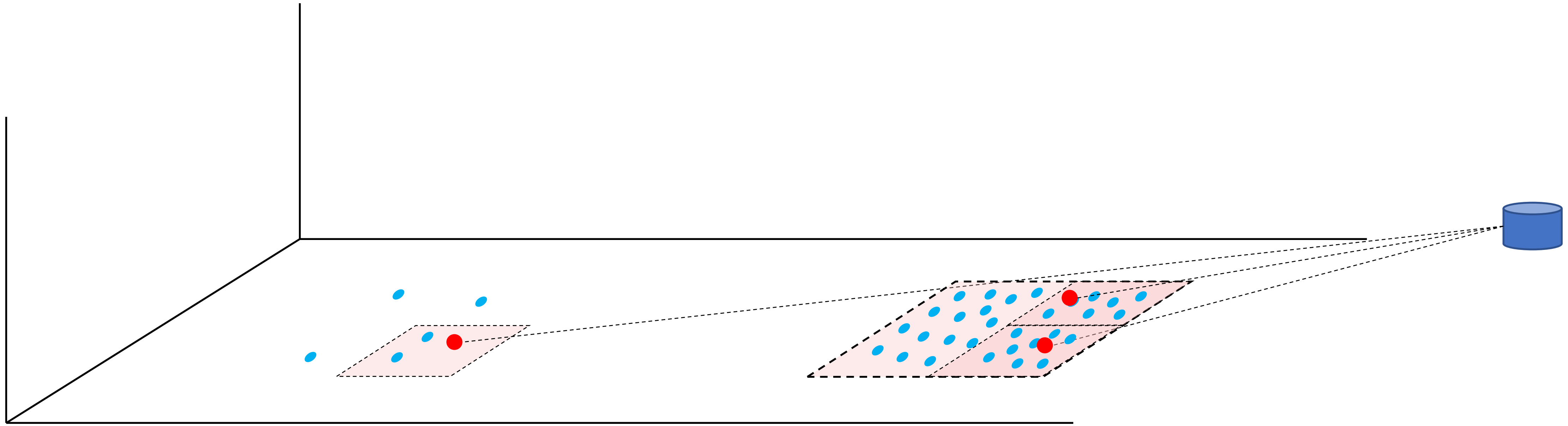
$\mathbf{S}_i = \mathbf{S}_m + \mathbf{S}_n$

Methodology

❑ Multi-Scale Association:

- Distant small-scale voxels may not have enough points and good planarity when the map of that area is new:

→ cost function may consist of factors from close-range points only → estimate optimized for short-term accuracy only.

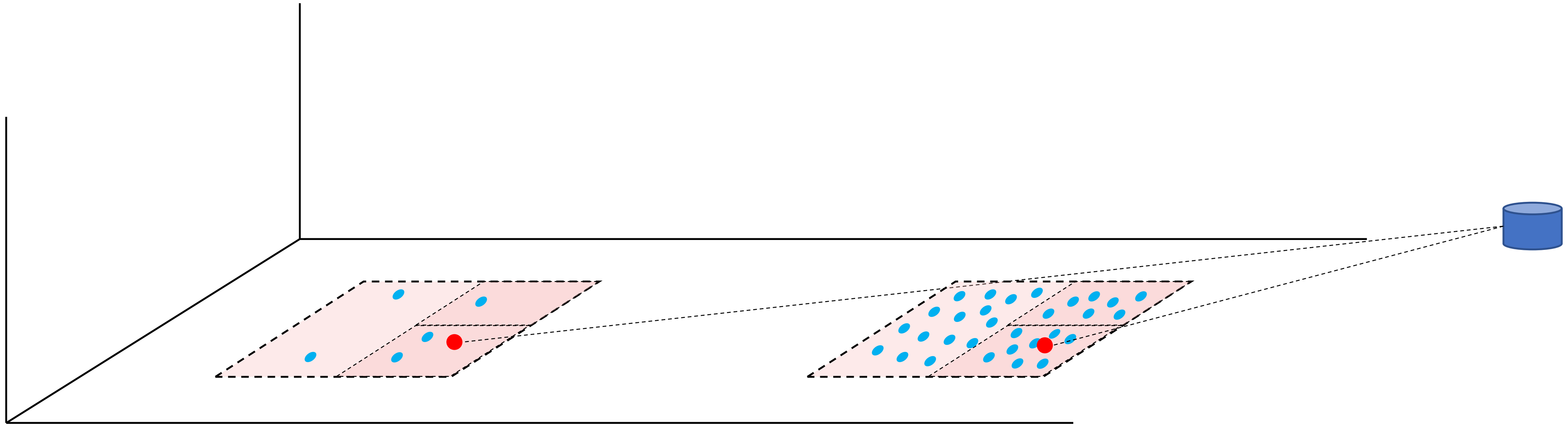


Methodology

❑ Multi-Scale Association:

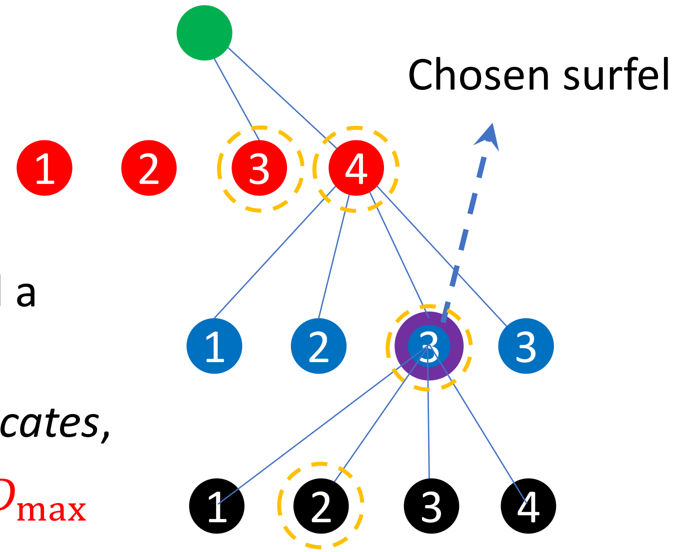
- Going up the scale, surfels may have enough points and planarity

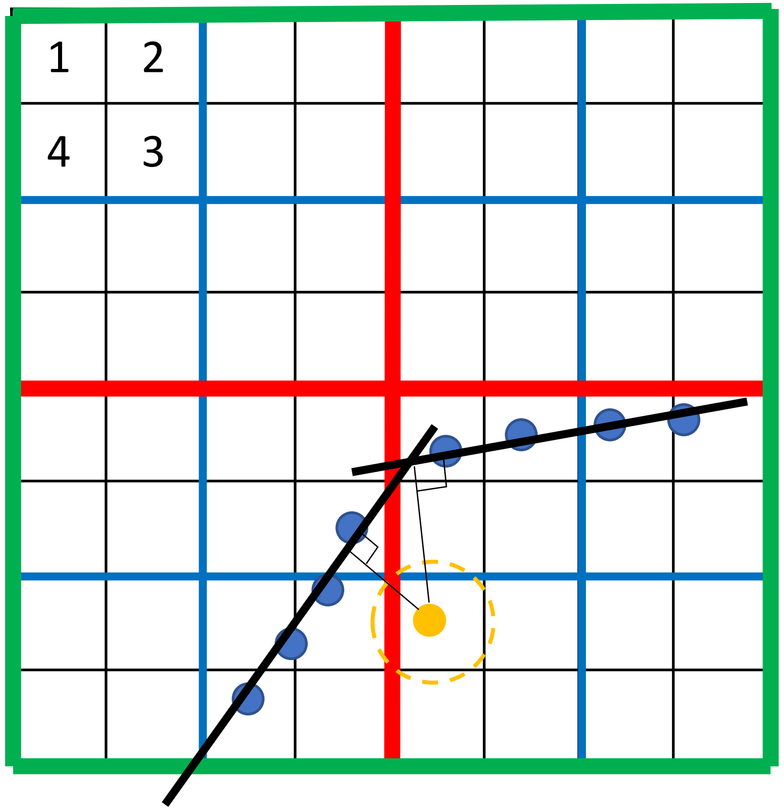
→ balanced optimization for both short term and long term accuracy.



Methodology

❑ Association Strategy:





Given a deskewed point $\breve{\mathbf{f}}$, and a surfel map, find a surfel $\mathcal{V}_i$ to build the cost factor.

- <u>1st stage:</u> Select all nodes satisfying the *predicates*,
  - Node's depth on the tree is between 1 to $D_{\max}$
  - Number of points $\geq N_{\min}$
  - Surfel planarity $\geq \rho_{\min}$
  - Intersect with a ball of radius r
- <u>2nd stage</u>
  - Sort all the nodes by the scales.
  - Starting from the smallest scale, find the surfel $\mathcal{V}_i$ with the shortest distance to $\breve{\mathbf{f}}$, denoted as d_i.
  - If $d_i < d_{\max}$, admit $\mathcal{V}_i$ to the buffer for construction of the cost factor.
  - Otherwise, move to the next scale.

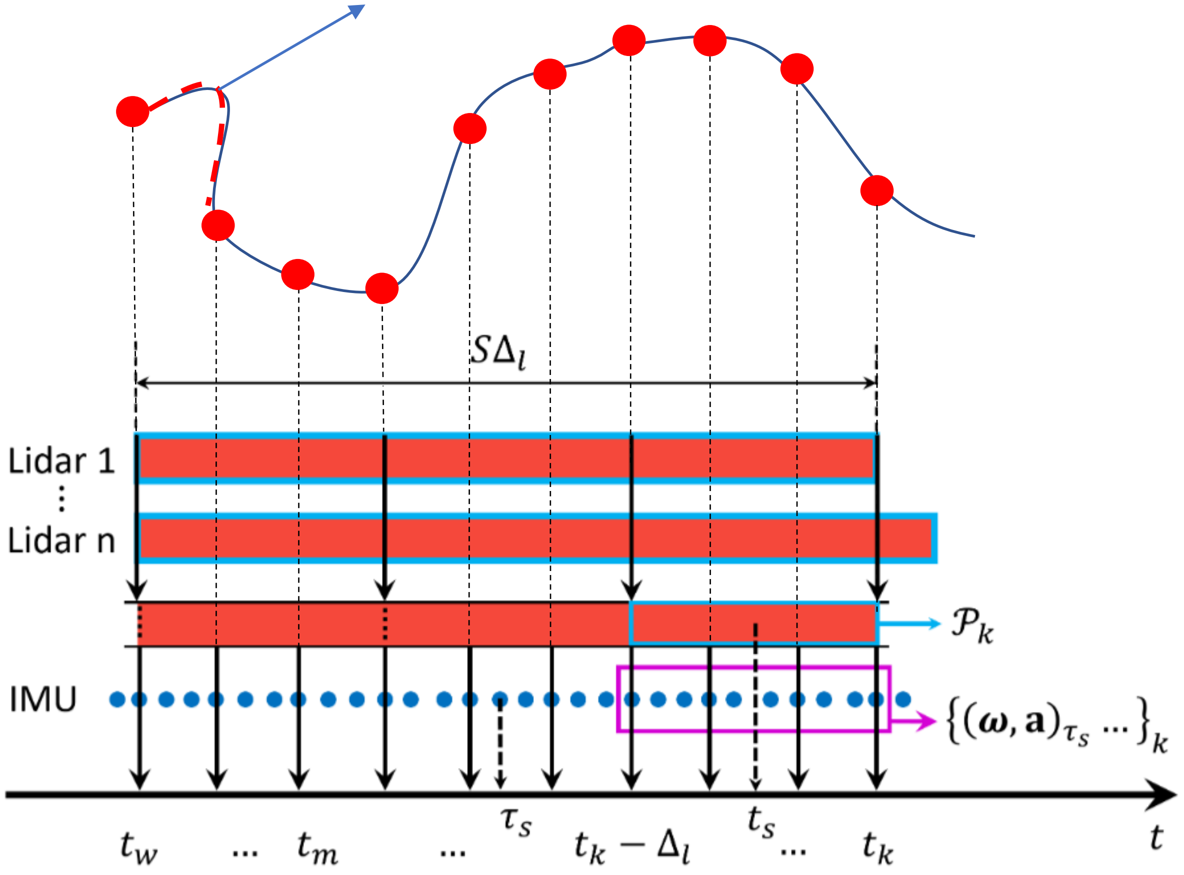
Methodology

- Increase "model capacity":

Add more than one state estimate per scan → better capture the dynamic, shorter propagation

- Cost function:

$$f(\hat{\mathcal{X}}) = \sum_{m=w}^{k-1} \left\| r_{\mathcal{J}}\left(\mathcal{J}_m, \hat{\mathcal{X}}_m, \hat{\mathcal{X}}_{m+1}\right) \right\|^2_{\mathrm{P}_{\mathcal{J}}} + \sum_{m=w}^{k-1} \sum_{\mathcal{L} \in \mathcal{A}_m} \left\| r_{\mathcal{L}}\left(\mathcal{L}\left({}^{B_{t_s}}\mathrm{f}, \mathbf{n}, \mu\right), \hat{\mathcal{X}}_m, \hat{\mathcal{X}}_{m+1}\right) \right\|^2_{\mathrm{P}_{\mathcal{L}}}$$



- Continuous-Time lidar factor (using a linearly changing trajectory model)

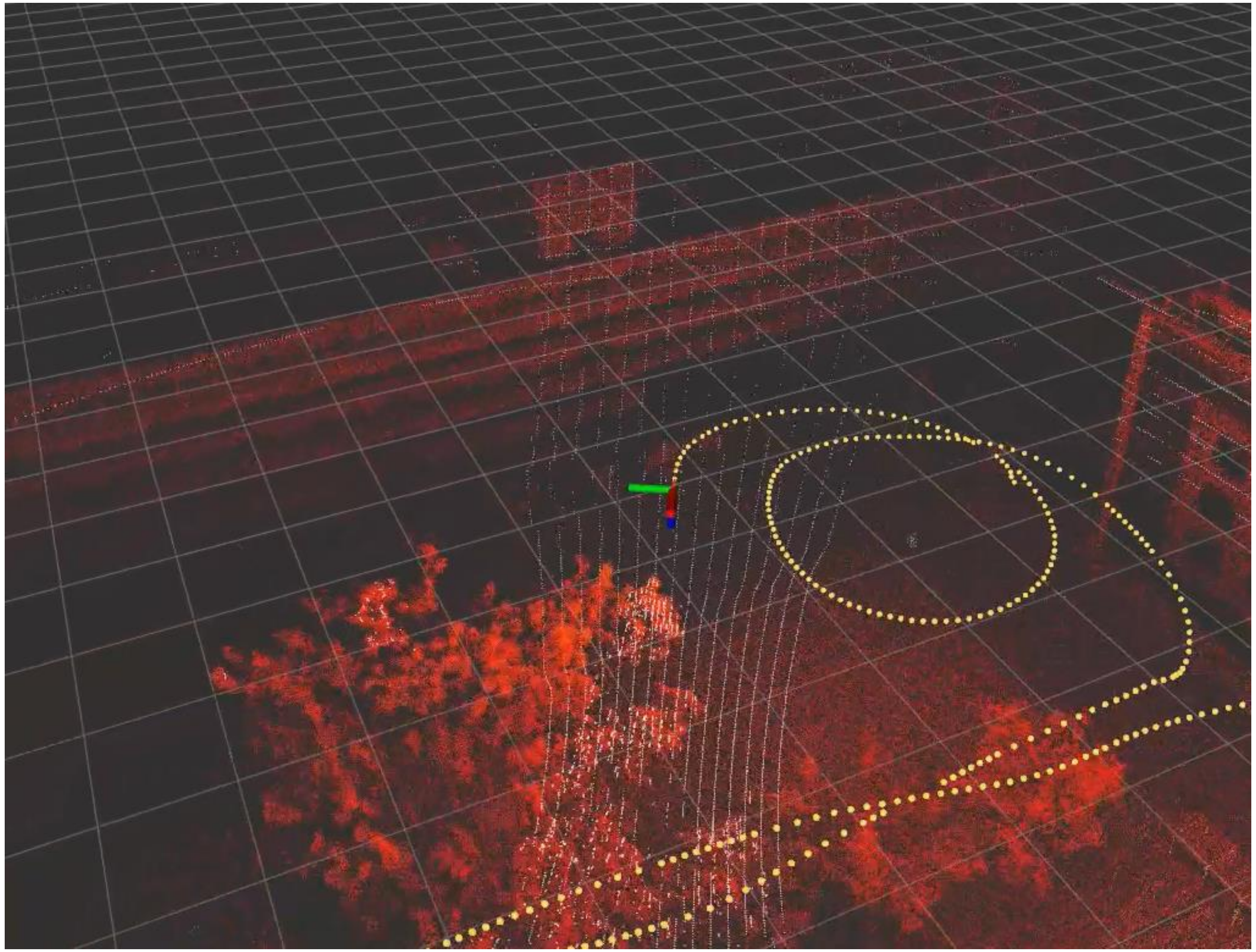
$$r_{\mathcal{L}} = \mathbf{n}^{\top}\left(\hat{\mathbf{R}}_m \mathrm{Exp}\left(\frac{t_s - t_m}{t_{m+1} - t_m} \mathrm{Log}\left(\hat{\mathbf{R}}_m^{-1} \hat{\mathbf{R}}_{m+1}\right)\right) {}^{B_{t_s}}\mathrm{f} + \hat{\mathbf{p}}_m + \frac{t_s - t_m}{t_{m+1} - t_m}\left(\hat{\mathbf{p}}_{m+1} - \hat{\mathbf{p}}_m\right) - \mu\right)$$

Experiments

- Benchmarking:

TABLE I: ATE of SLICT compared with other methods on NTU VIRAL datasets (unit [m]). The best results are in **bold**, second best are underlined. 'x' denotes divergence.

Dataset	MARS	LIO-SAM	Voxel-Map	FAST-LIO2	SLICT
eee_01	0.2471	0.0624	0.0699	0.0585	**0.0316**
eee_02	0.1033	0.0457	0.0506	0.0318	**0.0249**
eee_03	0.0927	0.0403	0.0631	0.0351	**0.0275**
nya_01	0.0555	2.0960	0.0508	0.0305	**0.0229**
nya_02	0.0624	x	0.0425	0.0286	**0.0227**
nya_03	0.0831	0.0468	0.0494	0.0315	**0.0260**
sbs_01	0.1370	0.0444	0.0535	0.0324	**0.0298**
sbs_02	0.1256	0.0461	0.0525	0.0322	**0.0291**
sbs_03	0.1588	0.0494	0.0498	0.0428	**0.0335**
rtp_01	x	0.2571	9.7416	0.0494	**0.0447**
rtp_02	0.2329	0.1091	2.4479	0.1151	**0.0466**
rtp_03	0.1377	0.0576	0.0792	0.0543	**0.0501**
tnp_01	0.0734	x	0.0326	0.0432	**0.0287**
tnp_02	0.0681	0.0330	0.0247	0.0590	**0.0201**
tnp_03	0.0665	**0.0283**	0.0331	0.0468	0.0383
spms_01	x	0.1620	11.3792	0.0686	**0.0610**
spms_02	x	0.6641	x	**0.0821**	0.1000
spms_03	19.8650	1.0071	x	**0.0603**	0.0661



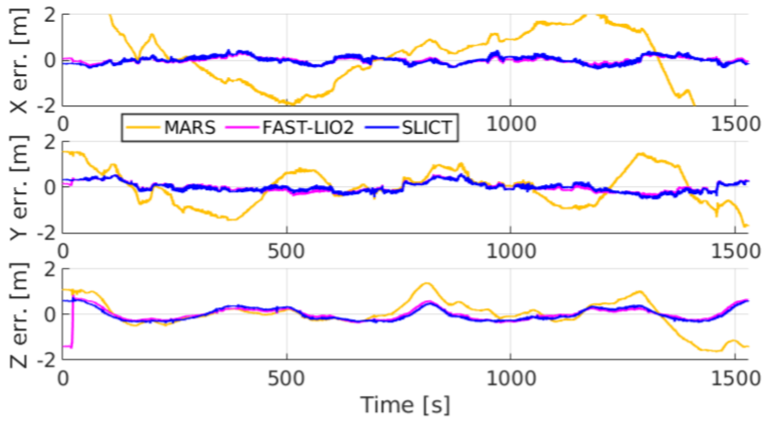
- NTU VIRAL dataset: multi-input, accurate ground truth, challenging aerial scenario, 100m x 100m area.
- Most accurate in most sequences.
- No Loop Closure Needed.

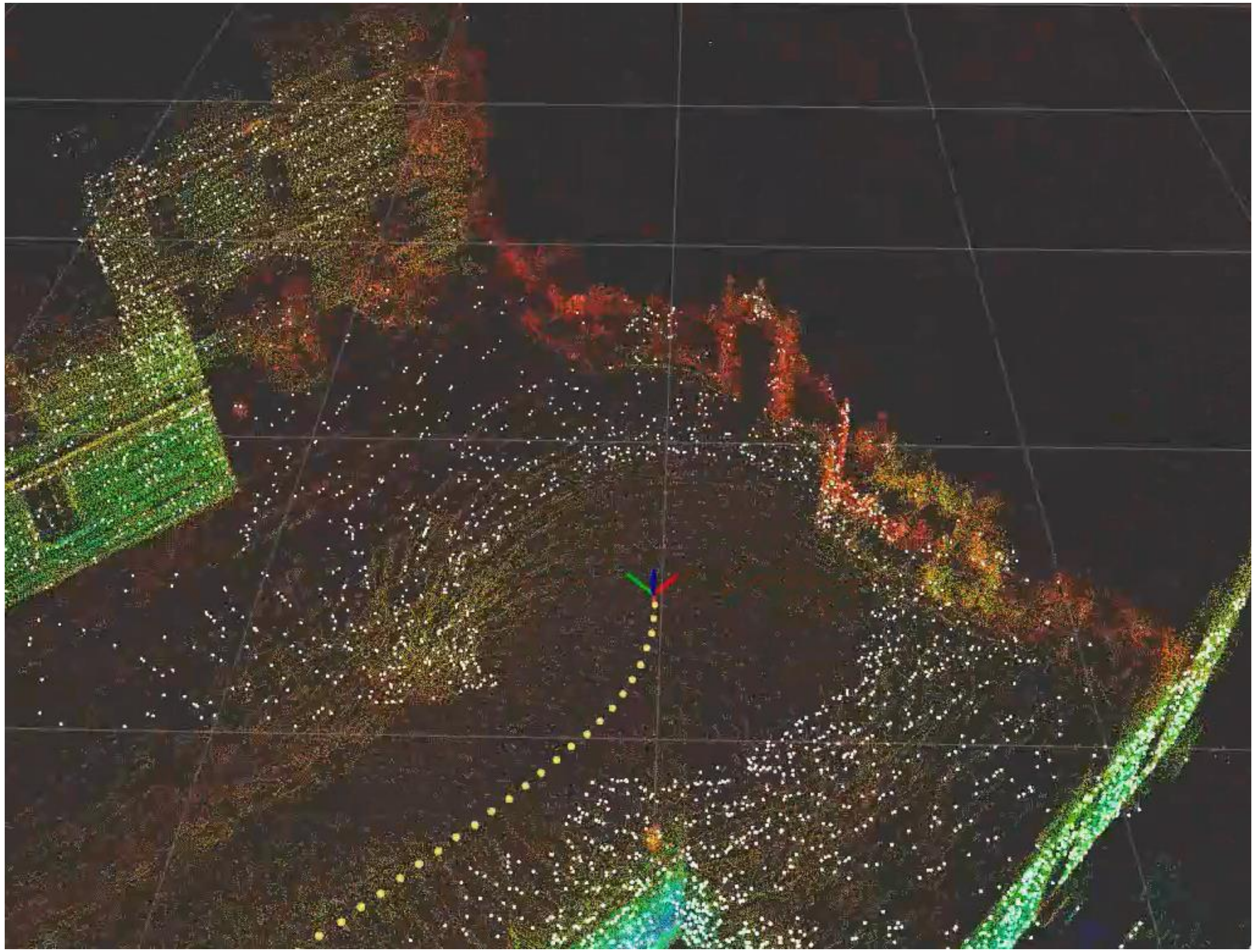
Experiments

❑ Benchmarking:

TABLE II: ATE of SLICT and other methods on Newer College Dataset (unit [m]). The best results are in **bold**, second best are underlined. 'x' demotes a divergent experiment.

Dataset	MARS	LIO-SAM	Voxel-Map	FAST-LIO2	SLICT
01_short_exp	2.1521	-	x	0.3883	**0.3843**
02_long_exp	6.0030	-	x	0.3659	**0.3496**
05_quad_dynamics	0.3729	-	17.0007	0.3443	**0.1155**
06_dynamic_spin	x	-	19.9993	**0.0800**	0.0844
07_parkland	x	-	x	0.1356	**0.1290**





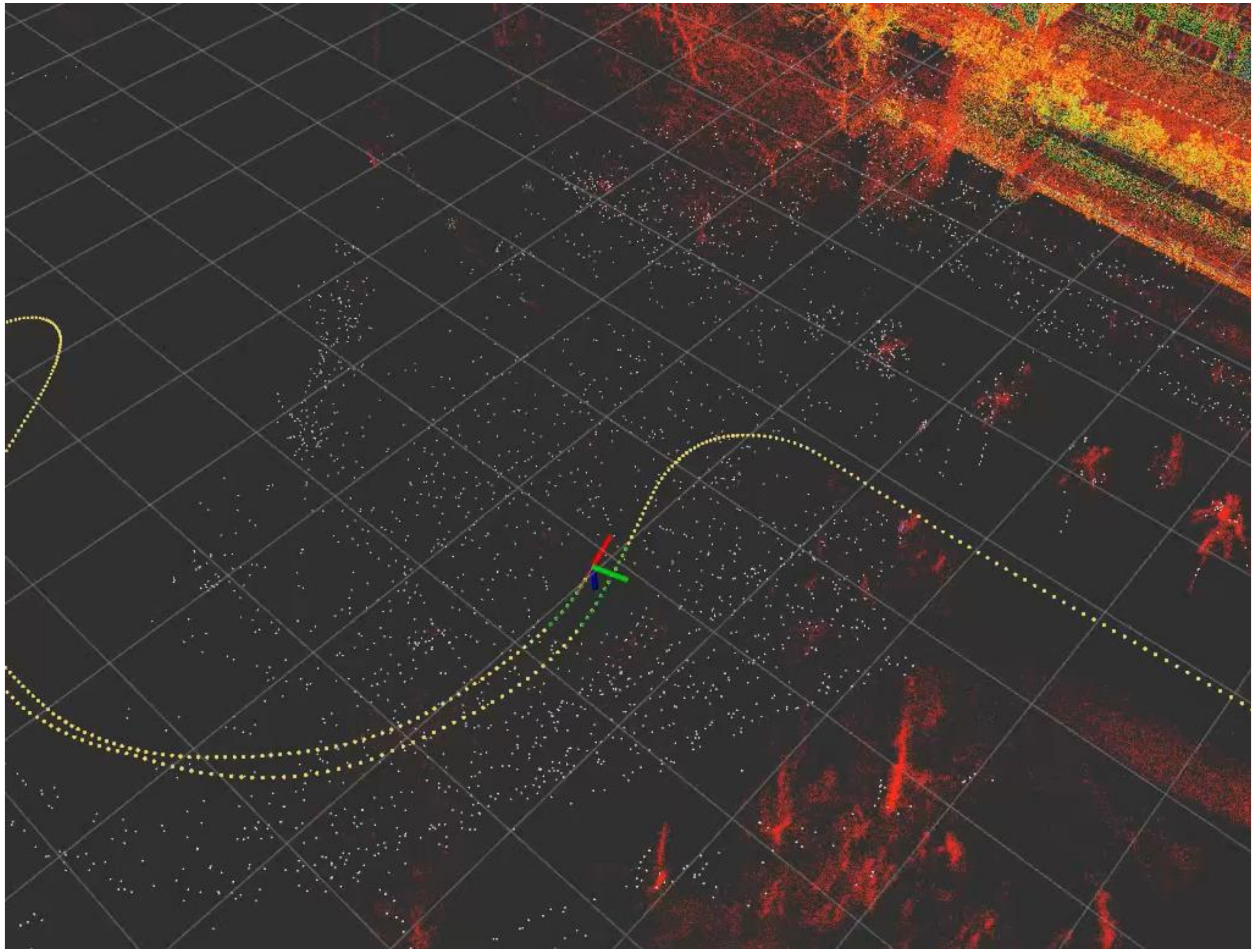
- Newer College Dataset: 200m x 200m area.
- Most accurate in most sequences. Error more noticeable, mostly in z direction.
- No Loop Closure Needed

Experiments

- Benchmarking:

TABLE III: ATE of SLICT and other methods on in-house dataset (unit [m]). The best results are in **bold**, second best are underlined. 'x' demotes a divergent experiment. LC denotes experiments with loop-closure and pose-graph optimization.

Dataset	LIO-SAM	Voxel-Map	FAST-LIO2	SLICT	LIO-SAM (LC)	SLICT (LC)
seq_01	4.0678	7.8550	1.7658	**1.0778**	1.2931	**0.7437**
seq_02	3.8518	x	1.2244	**0.7372**	0.9685	**0.5401**
seq_03	x	9.5255	1.1653	**0.5789**	x	**0.6226**

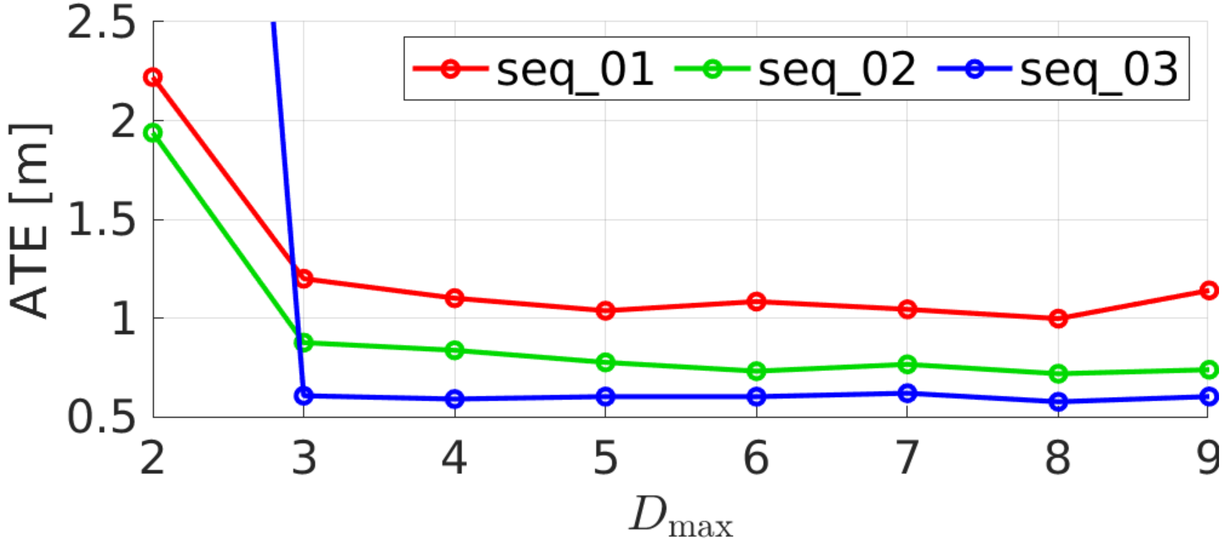


- MCD VIRAL, NTU ATV: Spinning lidar + Epicycle Lidar. 400m x 600m area, 2km route, very high speed.
- Most accurate in most sequences. Error very noticeable, especially in z direction.
- Loop closure is used and shows effectiveness.

Experiments

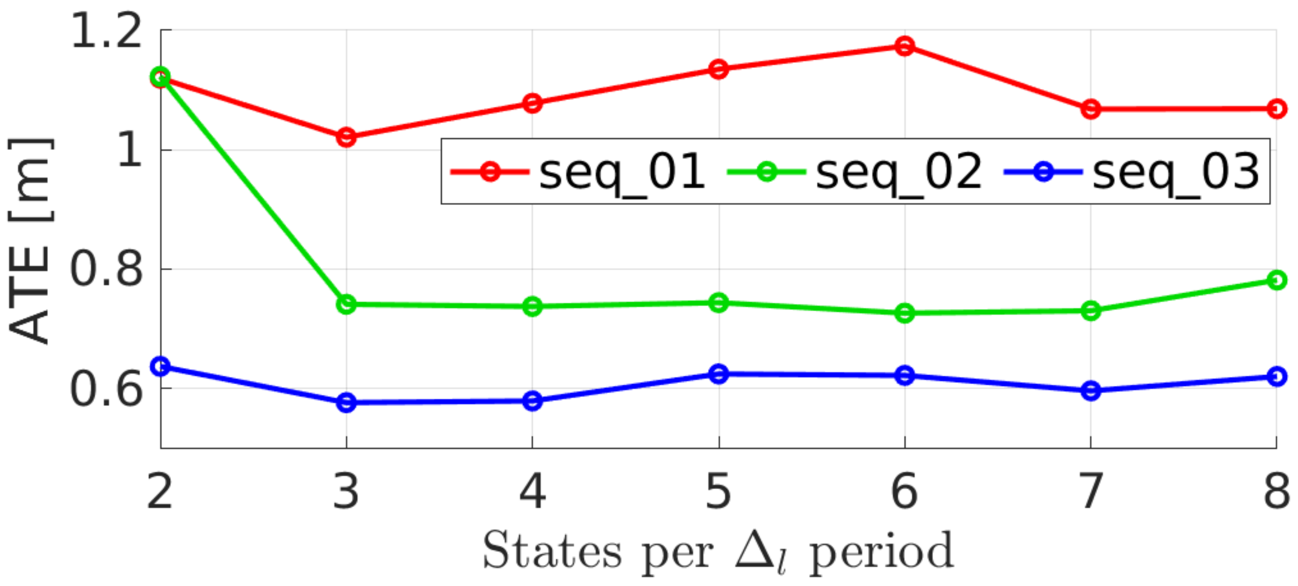
❑ Ablation Study – Effect of Maximum Associable Scales:

- Changing $D_{\max}$ from 2 to 9 → maximum associable scale varies from $2^2 \times 0.05 = 0.2m$ to $2^9 \times 0.05 = 25.6m$.
- Error reduces when $D_{\max}$ increases.
- Beyond $2^5 \times 0.05 = 1.6m$, the maximum associable scale no longer has influence.



Experiments

- Ablation Study – Effect of Number of States Per Scan:
  - Changing the number of state estimates per scan in the sliding window.
  - Ablation study shows that error reduces when having more state estimates per scan.
  - 3 and 7 states per scan seem to have the best performance.



Thank you