RECAP: 3D Traffic Reconstruction

Presenter: Yunxiang Chi
10/23/2025

Background

Accident Reconstruction

Most traffic accidents results are from police officer
If run into legal dispute -> employ a certified accident reconstructionist

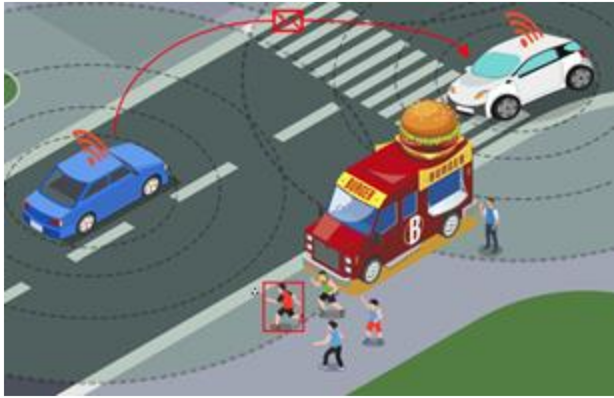
- Assessment of damage
- Reports of witness
- Crash recorders for speed, throttle, position, etc.

-> virtual video that recreates the approximate scene

[1] https://www.patrickdaniellaw.com/blog/accident-reconstructionist/
[2] https://ivanmacella.com/en/what-traffic-collision-reconstruction-mean/
[3] https://www.atlantaeng.com/3daccidentsimulations.html

3D traffic reconstruction pipeline

Frame: traffic participants + surrounding scene → **Volumetric videos**: 3D time evolution of traffic → **Downstream tasks:** Accident reconstruction, traffic analytics, model training, cooperation, etc.



[1] http://www.designwareinc.com/3d_anim.htm
[2] Vehicle-to-Everything Cooperative Perception for Autonomous Driving; Huang et al.

Requirements for 3D traffic reconstruction

Accuracy

a. **Reconstruction error:** average distance between a point in reconstructed point cloud and that in the ground-truth point cloud
b. **Requirement:** Autonomous vehicles can position themselves within 10-20 cm via sensors

Coverage

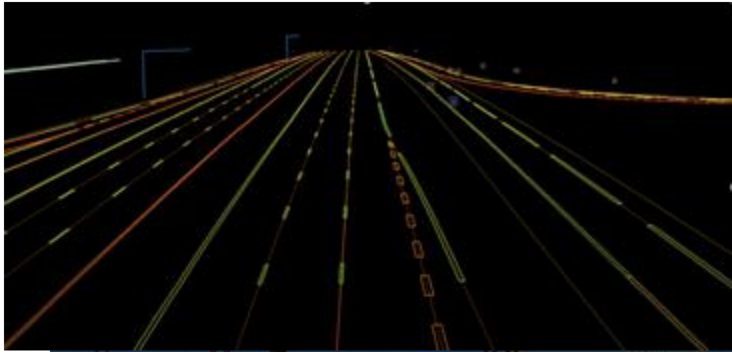
a. As large a spatial region with the accuracies above as possible for longer time duration of trajectories

Time-to-reconstruction

a. Police officer use reconstruction for the accident report
b. Average responde times >10mins

Non-reliance on HD Maps

a. High cost of collecting and updating HD maps over large spatial areas
   i. Hardware: cost $200k-300k sensor vans
   ii. Streaming requirements: 7+ figures per update
   iii. Time cost: weeks of update cycles for roads change(work zones, lane shifts, etc.)



[1]https://www.autonomousvehicleinternational.com/features/the-road-to-everywhere-are-hd-maps-for-autonomous-driving-sustainable.html

Sensors for vehicle traffic reconstruction

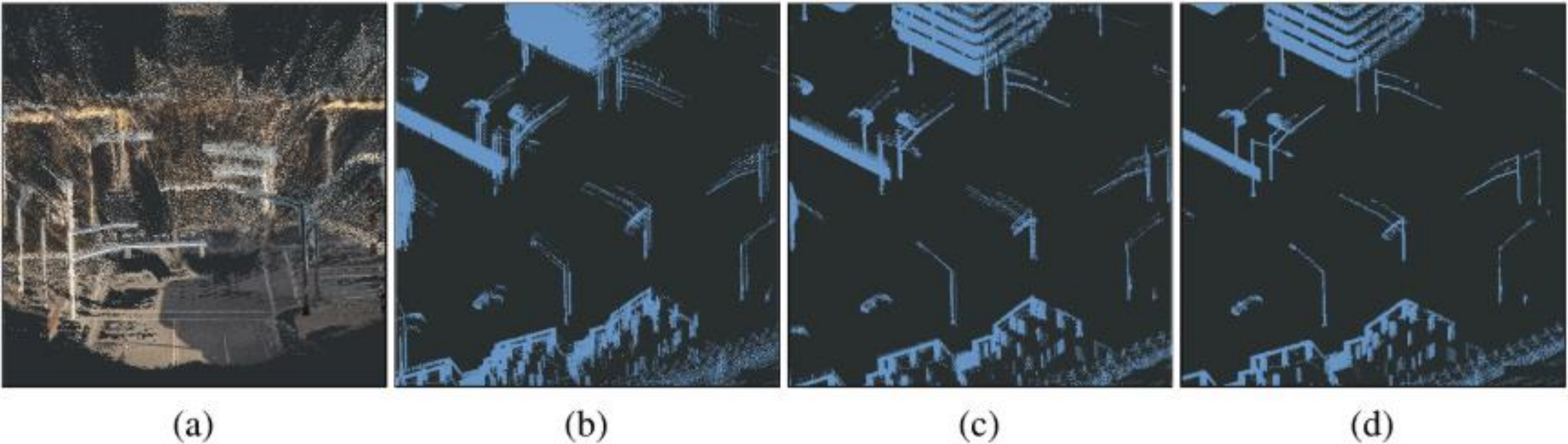
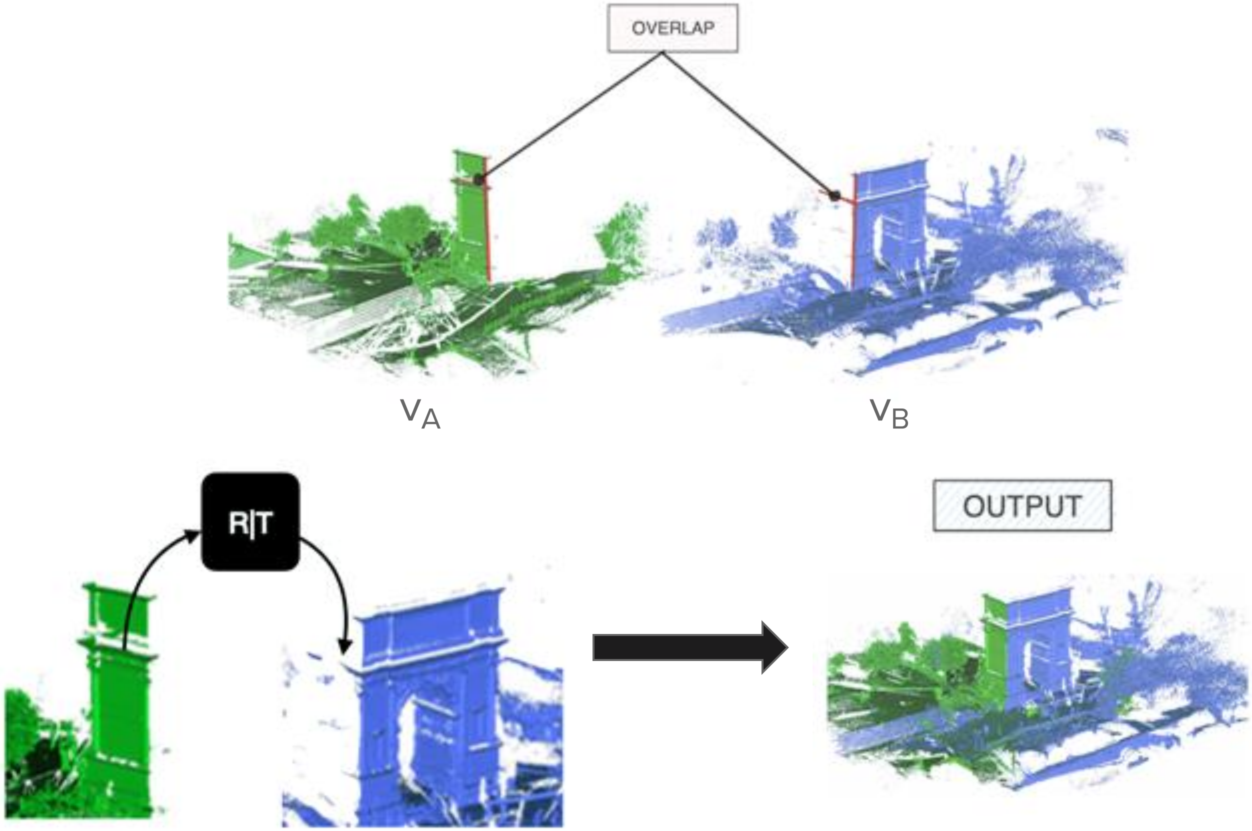


FIGURE 2: (a) Point cloud from ColMap. Moving objects (*e.g.*, vehicles or pedestrians) are absent in the point cloud. (b) Point cloud from GPS+IMU pose (error: ***1.13m***). (c) Point cloud from HD map positioning (error: ***20cm***). Both (b) and (c) have blurry alignments while (b) is worse than (c). (d) Ground-truth point cloud.

Photogrammetry: CARLA simulator + ColMap photogrammetry tool -> works for static objects but not moving objects + long processing time(4 hours for 115 images from one vehicle)
GPS+IMU: Transform lidar points into the GPS coordination -> fast(~1 min), sub-meter error
HD Maps: Mentioned in the previous slide

Promising approach-Point clouds registration



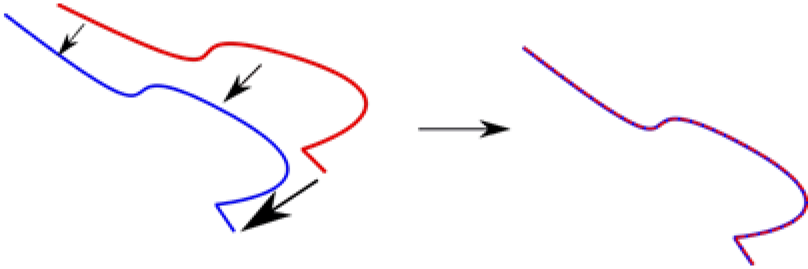
[1] https://www.thinkautonomous.ai/blog/point-cloud-registration/

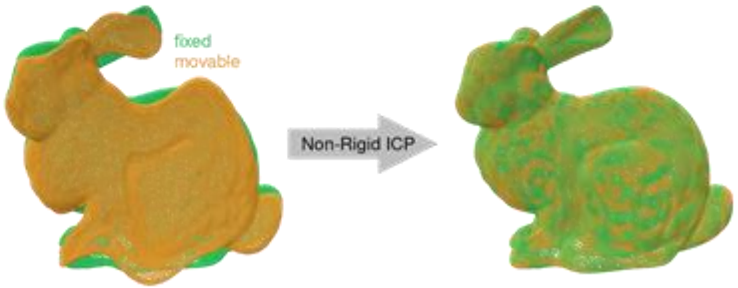
Point clouds registration Cont.

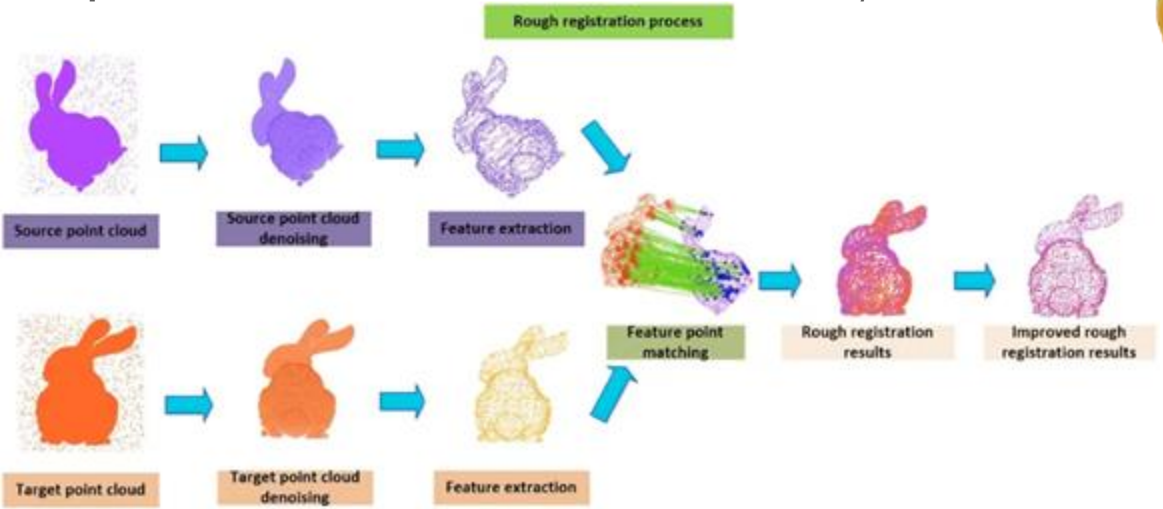
Input: Two point clouds, A (source) and B (target).
Initialize: Start with an initial guess of the transformation T_0
Iterate:

a. **Correspondence**: For every point in A, find its closest point in B (within overlap region).
b. **Alignment**: Compute the rigid transform T that minimizes the mean squared distance between pairs.
c. **Update**: Apply new T to A & repeat until convergence.

Output: The final transformation T* that best aligns A to B.



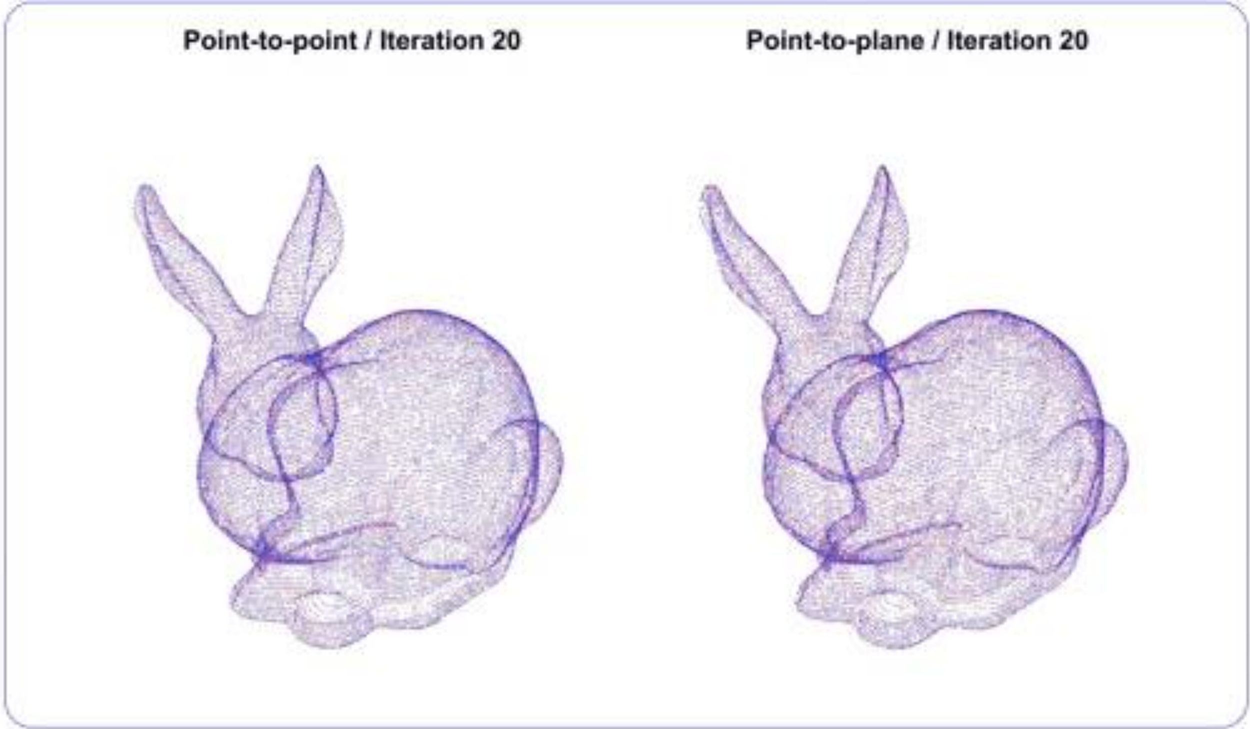




[1] https://learnopencv.com/iterative-closest-point-icp-explained/
[2] https://graphics.stanford.edu/~smr/ICP/comparison/chen-medioni-align-rob91.pdf

Iterative Closest Point

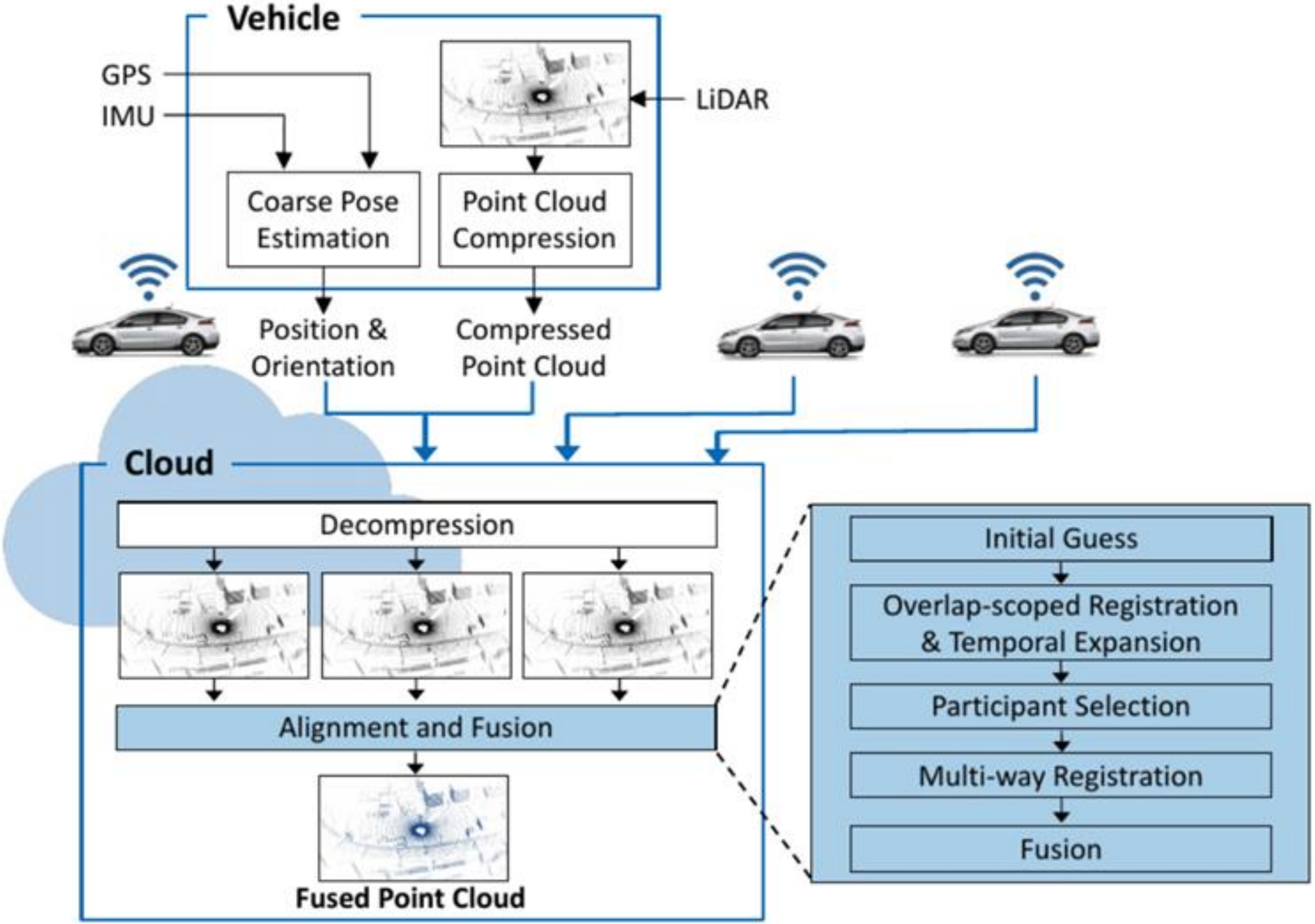
Comparison of point-to-point and point-to-plane error metric

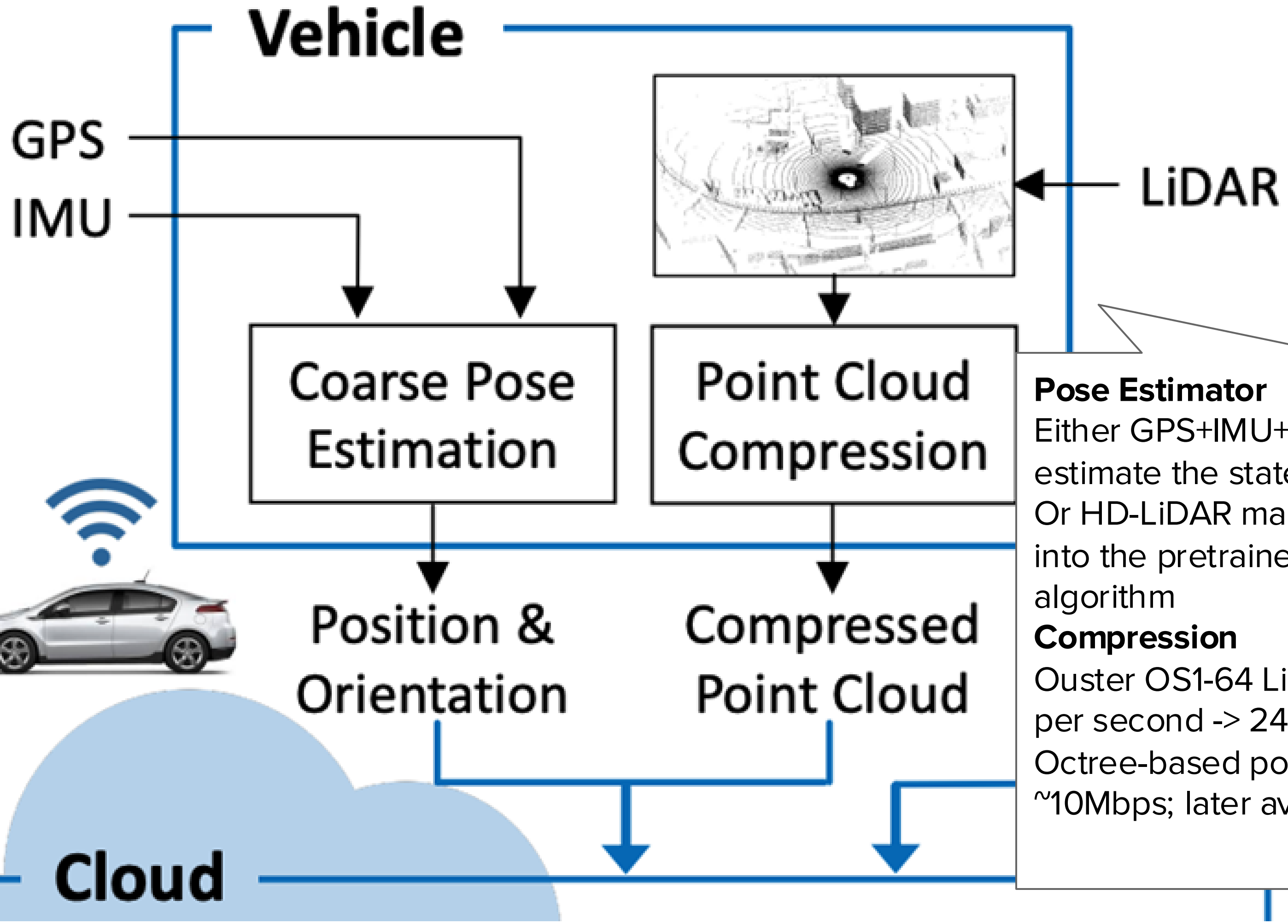


Challenges

1. How to find the overlapping area in the point clouds quickly
2. How to ensure the low delay of streaming toward cloud
3. How to ensure the overall time and hardware cost is low
4. How to enable larger spatial coverage

Design

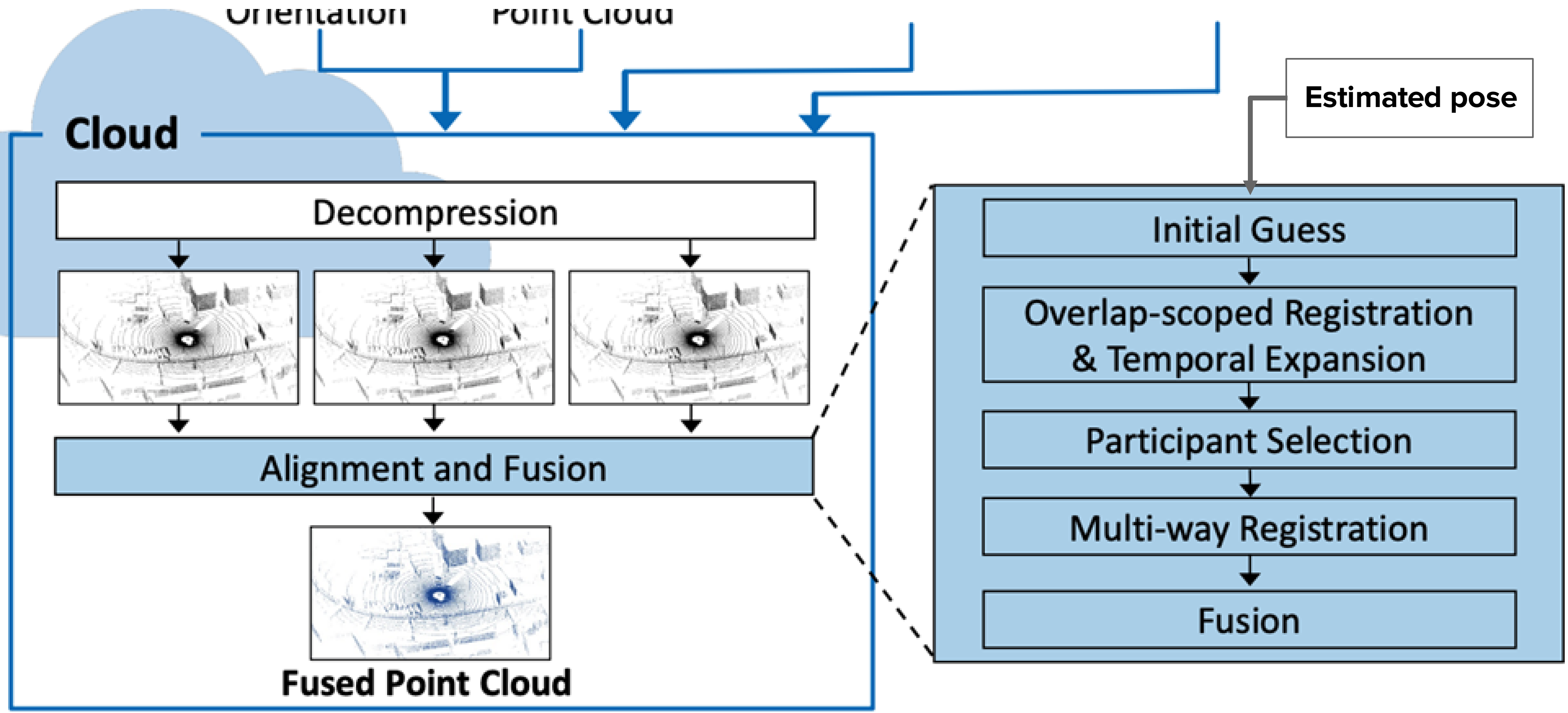

Vehicle
GPS
IMU
LiDAR
Coarse Pose Estimation
Point Cloud Compression
Position & Orientation
Compressed Point Cloud
Cloud
Decompression
Alignment and Fusion
Fused Point Cloud
Initial Guess
Overlap-scoped Registration & Temporal Expansion
Participant Selection
Multi-way Registration
Fusion



Pose Estimator
Either GPS+IMU+Kalman filter(continuously estimate the state of a system)
Or HD-LiDAR map matching(feeds HD maps into the pretrained ICP) + fast registration algorithm

Compression
Ouster OS1-64 LiDAR generates 10 point clouds per second -> 240Mbps
Octree-based point-cloud compression -> ~10Mbps; later avg 20Mbps, max 34Mbps

[1] Octree-based Point-Cloud Compression; Schnabel et al.


Orientation
Point Cloud
Estimated pose
Cloud
Decompression
Alignment and Fusion
Fused Point Cloud
Initial Guess
Overlap-scoped Registration
& Temporal Expansion
Participant Selection
Multi-way Registration
Fusion

Overlap-scoped Registration

Before alignment, crops all point that's below a height *h* from the LiDAR sensor

For two point cloud p_i and p_j:

- Set nominal LiDAR range r
- For every p_i, if there's at least one point in p_j within threshold δ_c **OR** it lies within r of vehicle j, then it's a potential correspondence
- Add each potential correspondence q_i into $q_{i,j}$(i's overlapped point cloud with j)
- Similarly, compute $q_{j,i}$
- Apply the ICP to $q_{i,j}$ and $q_{i,j}$, output new T*

After alignment, generate fused point clouds with new T using original point clouds to recover the cropped objects

(a) 0.07 m (b) 2.28 m (c) 0.32 m

FIGURE 4: Importance of overlap. (a) ICP applied to two point clouds with high overlap. The number below the figure shows reconstruction error (lower is better). (b) ICP applied to two point clouds with minimal overlap, resulting in high error. (c) ICP applied to two minimally overlapped point clouds after using overlap-scoped registration, resulting in lower error.

Overlap-scoped Registration Cont.

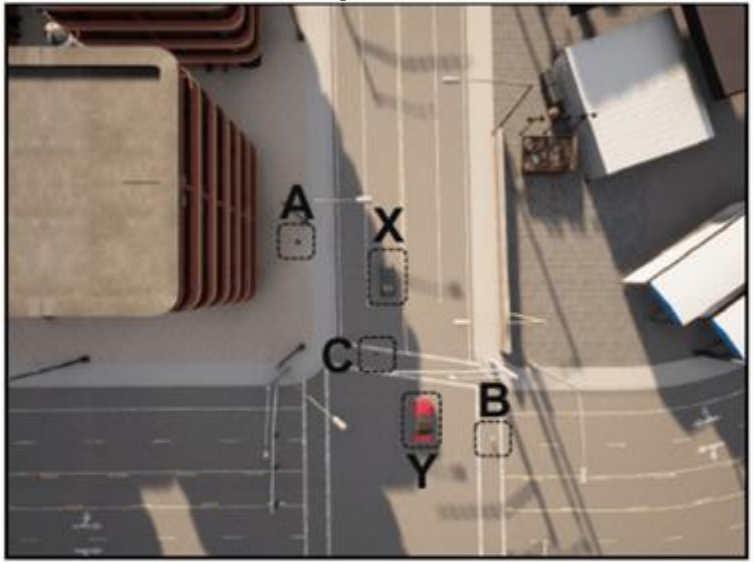
Vehicles: X, Y; Pedestrians: A, B, C

Range of X: {A, C}; Range of Y: {B, C}

Output: Relative positions of pedestrians & Denser point cloud for C

Limitations: *Loss of Coverage*

as vehicles move aways from each other, deg of overlap decreases to 0 quickly.



Temporal Expansion

Backward expansion:

If vehicles overlapped in the past (at t’ < t), fuse frames from [t’, t] to expand the vehicle now far away.

Forward expansion:

If they will overlap in the near future (t’ > t), pre-fuse frames from [t,t′]. RECAP does not expand if they never meet

P.S. Still apply small object removal before expansion

(a) (b)

FIGURE 6: (a) When two vehicles (**X**, **Y**) are far away, RECAP can *expand* **Y**’s point cloud across time, then align it with **X**’s point cloud. (b) In the composite point cloud, RECAP can find the relative positions of **A**, **B** and **C** after temporal expansion.

(a) Before (b) After

FIGURE 7: Before and after temporal expansion of vehicle **Y**.

Temporal Expansion Cont.

Focus on one vehicle(e.g. Y) that produces a sequence of LiDAR frame over time: $\{p_1, p_2, \dots, p_n\}$

Start with first frame p_1: line 1

If the two frames overlap enough -> skip

When overlap drops:

- Take the furthest frame that still overlapped p_{j-1}
- Run ICP between p_F and p_{j-1}
- Fuse

Then move the reference forward

Algorithm 1: Efficient Temporal Expansion

Input : A sequence of successive point clouds $\{p_1, \cdots, p_n\}$ from a single vehicle.

Output: A fused point cloud p_F which combines the input point clouds.

`/* Initialize loop indices & fused point cloud */`

1 $i \leftarrow 1, j \leftarrow i+1, p_F \leftarrow p_1$;

2 **while** $i < n$ **do**

`/* Find j with correspondence count < ρ */`

3 **if** $c_{i,j} \geq \rho$ **then**

4 $j \leftarrow j+1$;

5 continue;

6 **end**

`/* If j reaches the end, run the last ICP */`

7 **if** $j > n$ **then** $j = n+1$;

`/* Run ICP and update fused point cloud */`

8 $T = \text{icp}(p_F, p_{j-1})$;

9 $p_F \leftarrow \text{fuse}(p_{j-1}, \text{transform}(p_F, T))$;

10 $i \leftarrow j-1, j \leftarrow i+1$;

11 **end**

12 **return** p_F

ICP & Expansion

- Extract overlapped point clouds $q_{i,j}$ and $q_{j,i}$ (overlap-scoped registration)
- If the count of potential correspondences is more than ρ, run ICP between $q_{i,j}$ and $q_{j,i}$ without expansion.
- Else
    - if i and j met at time t'(past or future), expand p_i and p_j till/from time t'
    - if the expanded point clouds have sufficient potential correspondences, run overlap-scoped registration on the expanded point clouds.

Multi-way Registration

Motivation: After all pairwise ICPs, some alignments are accurate but some are noisy -> poor reconstruction if we feed all of them into global coordination

Participant Selection:

1. For every pair(i,j), check number of correspondences $c_{i,j}$.
   If $c_{i,j} > \mu$, keep edge; else, drop the edge
2. Build graph: nodes=vehicles, edges=reliable ICP links
3. Find maximum clique-the largest fully connected subset of vehicles-as participants.

Multi-way Registration

Form pose graph:

- Node i: unknown global pose of vehicle i
- Edge (i,j): known relative transform $T_{i,j}$ from pairwise ICP.

Define error term for each edge and solve the optimization problem -> set of global transformations T_i that minimize total misalignment

Evaluations

Setup

- Software:
  - Point Cloud Library for pairwise ICP and Open3D for pose graph optimization
  - SensorLog to collect GPS and gyroscope data
  - CARLA simulator for realistic traffic environment, participants behavior, and sensors from traces
- Hardwares:
  - Intel i9-9900K CPU(16 cores, 3.6GHz)
  - Ouster OS0-64 or OS1-64, GPS and gyroscope on each vehicle
  - Xsens GNSS MTi-680G RTK for ground-truth pose
- Scenarios
  - 4-way intersection, T-junction, roundabout
  - Num of vehicles: 3-13 for each; three rounds with randomized traffic flow for each
- Baselines:
  - GPS+IMU
  - GPS+IMU+KF
  - HDMap
  - Different ICP optimization approaches: SAC-IA, FGR, Go-ICP

3D TRAFFIC RECONSTRUCTION

CARLA SIMULATOR

4-WAY INTERSECTION

CARLA SIMULATOR

ROUNDABOUT

CARLA SIMULATOR

T-JUNCTION

REAL-WORLD

OFF-CAMPUS (E)

REALWORLD

OFF-CAMPUS (D)

	# of Vehicles					
Scheme	**3**	**5**	**7**	**9**	**11**	**13**
SAC-IA	2.59	8.16	8.08	8.37	8.70	9.02
FGR	3.22	6.89	6.54	7.84	7.81	9.47
Go-ICP	2.82	7.76	7.47	7.68	7.97	9.27
GPS+IMU	0.95	1.10	1.14	1.18	1.14	1.23
GPS+IMU+KF	0.62	0.68	0.69	0.71	0.68	0.83
HDMap	0.14	0.19	0.20	0.27	0.22	0.40
RECAP	0.07	0.10	0.12	0.13	0.12	0.15

TABLE 3: Avg. reconstruction error (m) of RECAP and baselines.

Real-world Trace	**A**	**B**	**C**	**D**	**E**
Avg. Reconstruction Error (m)	0.17	0.26	0.21	0.15	0.23

TABLE 4: Avg. reconstruction error (m) from five real-world traces.

	# of Vehicles					
Scene Type	**3**	**5**	**7**	**9**	**11**	**13**
4-way Intersection	0.07	0.10	0.12	0.13	0.12	0.15
T-junction	0.12	0.10	0.14	0.15	0.11	0.11
Roundabout	0.06	0.10	0.12	0.10	0.09	0.07

TABLE 5: Avg. reconstruction error (m) for different traffic scenes.

Component	# of Vehicles		
	3	7	13
Overlap + Expansion (p)	1004.51	2709.54	7100.13
Pairwise ICP (p)	345.19	441.78	556.24
Decompression (v)	76.10	81.29	88.75
Initial Guess + Crop (v)	1.99	2.29	2.55
Selection + Pose Optimization	1.14	1.50	2.55
Total	1428.94	3236.40	7750.22

TABLE 6: Avg. latency per frame for each components (ms).

	# of Vehicles		
Scene Type	**3**	**7**	**13**
4-way Intersection	3.65 (2.24)	10.13 (2.84)	15.33 (3.11)
T-junction	4.77 (2.30)	8.47 (2.99)	21.39 (6.32)
Roundabout	2.30 (2.17)	5.66 (3.26)	14.97 (5.61)

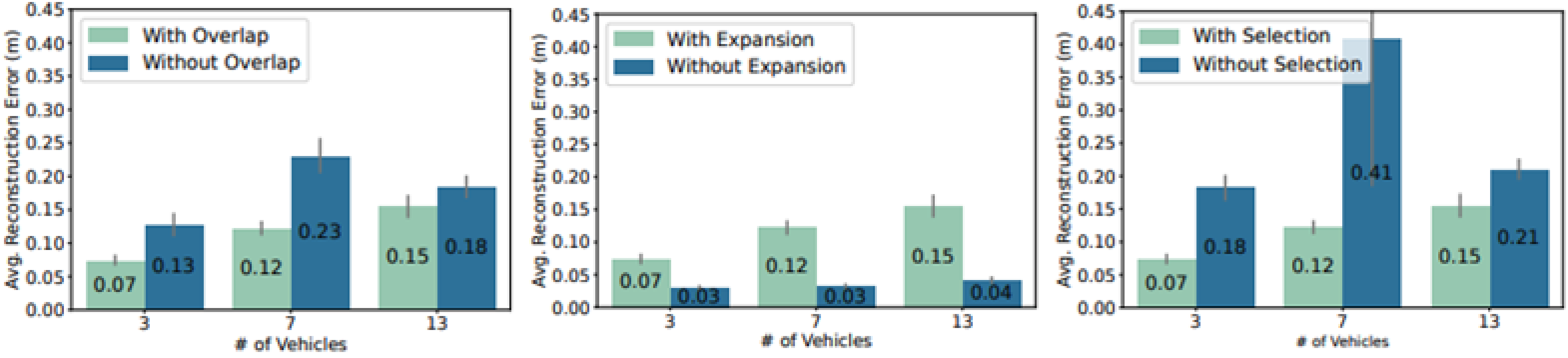
TABLE 7: Avg. time-to-reconstruction (minutes) for 150 frames.

# of Vehicles	3	7	13
With Overlap	3.65	10.13	15.33
Without Overlap	5.47	17.48	19.54

TABLE 8: Avg. time-to-reconstruction (minutes) with and without overlap extraction.

# of Vehicles	3	7	13
With Expansion	7,387	10,568	12,434
Without Expansion	5,928	6,647	7,754

TABLE 9: Avg. spatial coverage (m^2) with and without expansion.



(a) (b) (c)

FIGURE 8: Avg. reconstruction error (m) with and without (a) overlap-scoped registration, (b) temporal expansion, and (c) participant selection.

# of Vehicles	3	7	13
HD Map Positioning	0.09	0.15	0.18
GPS+KF Positioning	0.07	0.12	0.15

TABLE 13: Reconstruction error using HD maps as an initial guess.

μ	2000	3500	5000	6500	8000
Reconstruction Error (m)	0.15	0.13	0.11	0.09	0.08
# of Participants	7	6	6	4	4

TABLE 12: Reconstruction error and the number of participants varying the threshold μ. We use 5000 for μ.

Future Works & Opinions

Limitations

1. Time synchronization: RECAP uses NTP -> 10ms clock offset for 40mph vehicle -> 18cm displacement error
2. Memory and storage overhead when operate in real-world
3. No LTE exist, like lone underground tunnel (as Raheem mentioned)
4. Near realtime reconstruction by using accelerators(e.g. GPU)

Opinions

1. Great paper overall
    a. Nice story, research question, explanation, etc.
    b. Cool work with comprehensive experiments
2. Non-practical for real-world application recently
    a. Strong assumption of 5G deployment on V2X
    b. Memory & storage issue
    c. Cost: as far as I know, the cost of lidar system is less than $1500 over the world; but OS0-64 & OS1-64/128 will take more than $10,000 at least

Perusall