Robust Real-time Multi-vehicle Collaboration on Asynchronous Sensors

Qingzhao Zhang†*, Xumiao Zhang†*, Ruiyang Zhu†*, Fan Bai‡, Mohammad Naserian‡, Z. Morley Mao†

(*equal contributions) †University of Michigan ‡General Motors

Oct. 3, 2023
Why cooperative perception?

- Limited sensing on occluded or far-away objects

Occluded pedestrian

Far-away obstacles
Motivation 1: synchronization problem

- In multi-vehicle collaboration, the LiDAR images to be merged is not captured on the same timestamp.

**Consumer** is the vehicle receiving LiDAR data; **provider** is the vehicle providing LiDAR data.
Motivation 2: inaccurate blind spot estimation

- Existing systems trend to share sensor data about blind spots only.
  - However, inaccurate blind spot estimation compromise the sharing efficiency
  - e.g., AutoCast\(^1\) estimate blind spots based on observed objects and naive ray casting.

---

Overview

- Q: Synchronization problem?
  - A: Prediction
    - Leverage prediction algorithms to synchronize LiDAR point clouds.

- Q: Accurate blind spot estimation?
  - A: On-demand data sharing
    - Let consumers proactively request data they need.
For all CAVs, share occupancy maps

- The map labels occupied, free, and occluded areas
For consumers, prepare data requests

- Make a plan of data sharing for the next LiDAR cycle
  - *i.e., which producer share which area*
For producers, share requested data

- Share the latest point cloud on the requested areas, and synchronize the point clouds to the requested timestamp.

List of areas and a timestamp

Data request

LiDAR point cloud

Point cloud prediction

Synced LiDAR point cloud
Execute all processes in parallel

- Compared with single-CAV perception, the only delay is from data fusion.
RAO Perception Benefits and performance

- RAO achieves the best perception accuracy compared with EMP\(^1\) and AutoCast\(^2\).
  - *We used various simulated and real-world datasets,*
  - *We used PointPillars as the perception model.*

<table>
<thead>
<tr>
<th>Traffic Scene</th>
<th>Perception AP@0.5/ AP@0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Local-only</td>
</tr>
<tr>
<td>DAIR-V2X-C</td>
<td>48.99/40.78%</td>
</tr>
<tr>
<td>CARLA-SUMO</td>
<td>48.63/37.17%</td>
</tr>
<tr>
<td>- Town05</td>
<td>40.68/30.18%</td>
</tr>
<tr>
<td>- Town06</td>
<td>65.46/48.30%</td>
</tr>
<tr>
<td>- Town10HD</td>
<td>40.12/32.58%</td>
</tr>
<tr>
<td>Mcity</td>
<td>51.51/41.13%</td>
</tr>
</tbody>
</table>

System Overhead - Latency & Data Volume

- The total avg latency of all the modules is 80.82 ms (14.40 ms variance)
- RAO can process LiDAR at regular full frame rate of 10 FPS
- RAO incurs similar data overhead compared to the STOA approach

![Graph showing runtime (ms) vs. module type]

<table>
<thead>
<tr>
<th>Metrics</th>
<th>EMP</th>
<th>AutoCast</th>
<th>RAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiDAR Points</td>
<td>8320±3228</td>
<td>3140±2171</td>
<td>3110±2501</td>
</tr>
<tr>
<td>Control data (KB)</td>
<td>&lt;0.1</td>
<td>&lt;0.1</td>
<td>1.77±0.50</td>
</tr>
<tr>
<td>Total Volume (KB)</td>
<td>24.37±9.46</td>
<td>9.17±6.36</td>
<td>10.90±7.32</td>
</tr>
</tbody>
</table>
Summary

- RAO is a **real-time occlusion-aware** cooperative perception system running on **asynchronous** sensors.

- RAO tackles two problems in existing cooperative perception:
  - Use prediction methods to mitigate sensor asynchroniziation.
  - Use on-demand data sharing to optimize data scheduling.

Thank You!