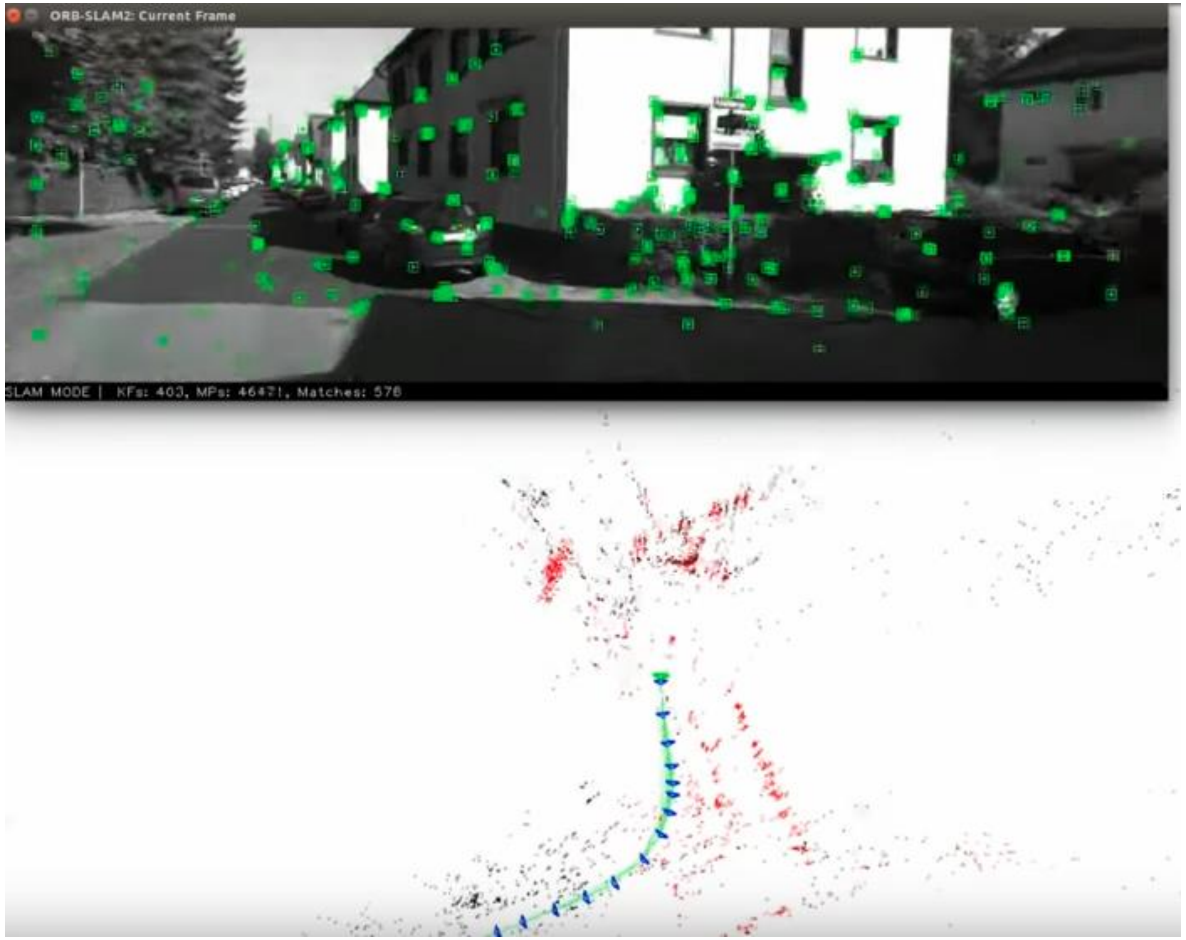
CloudSLAM: Edge Offloading of Stateful Vehicular Applications

Kwame-Lante Wright*, Ashiwan Sivakumar*, Peter Steenkiste*, Bo Yu‡, Fan Bai‡

**Carnegie Mellon University, ‡General Motors*

What is SLAM?

- Simultaneous Localization and Mapping (SLAM)
  - Generates 3D map of the environment
  - Estimates the pose (location and orientation) of a vehicle
  - Based on sensors such as stereo video or LIDAR



Raúl Mur-Artal, J. M. M. Montiel and Juan D. Tardós. **ORB-SLAM: A Versatile and Accurate Monocular SLAM System**.

SLAM Challenges for Vehicles

- Installing high-performance compute infrastructure in a vehicle is complex and costly
- Storage requirement does not scale well
- Simplifying the SLAM implementation to limit resource usage lowers quality of results

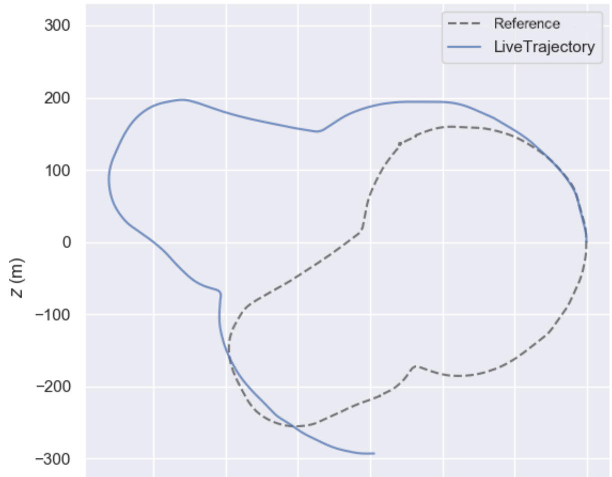
CloudSLAM Goals

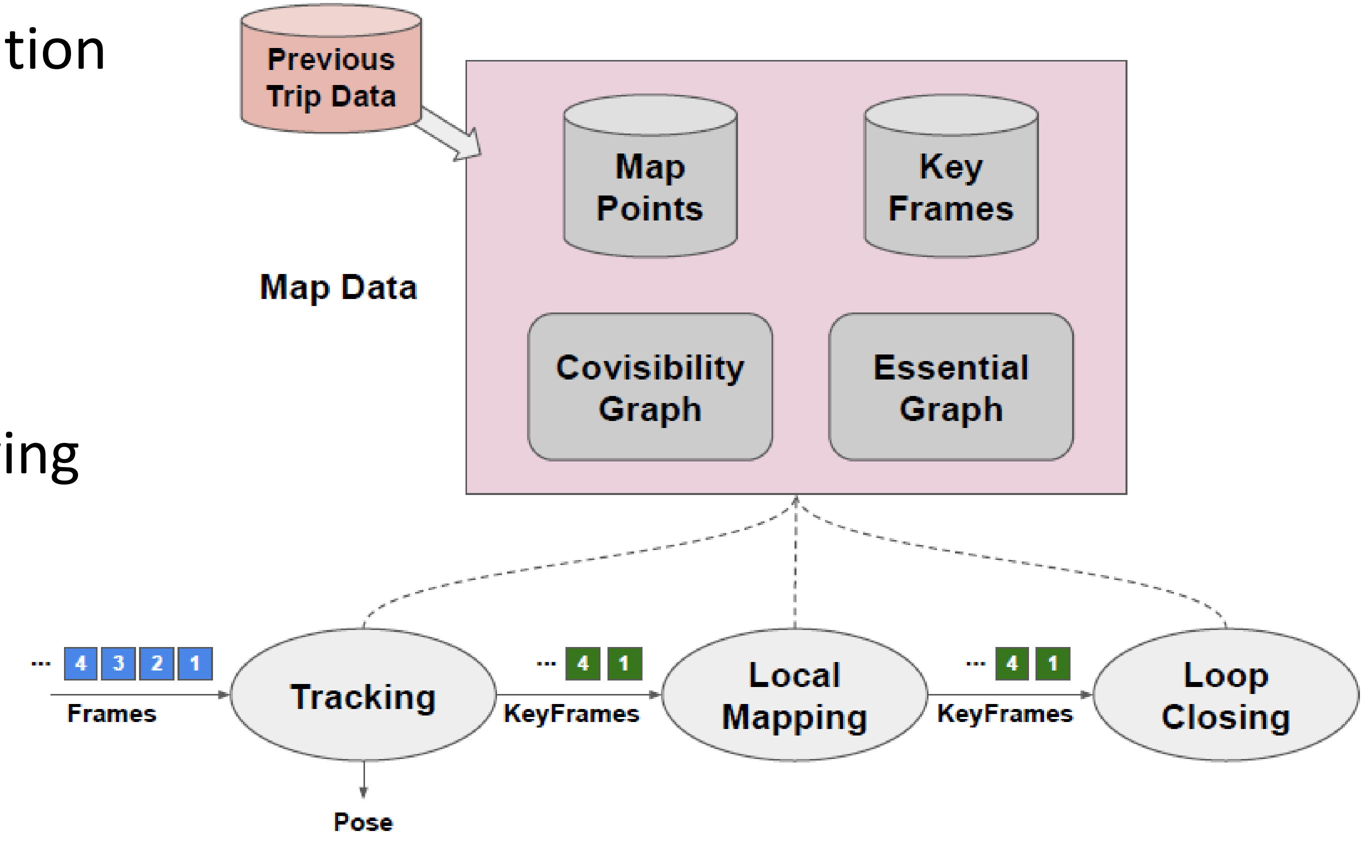
- Develop an offloading architecture for stateful, latency-sensitive applications

1. Utilize edge cloud resources to reduce CPU & memory load on the vehicle
2. Maintain accuracy similar to an unmodified SLAM implementation
3. Minimize network usage

Case Study: ORB-SLAM2

- State-of-the-art SLAM implementation
- Primary Modules
  - Tracking
  - Local Mapping
  - Loop Closing
- Previous trip data critical to achieving high accuracy





Options for Using the Cloud

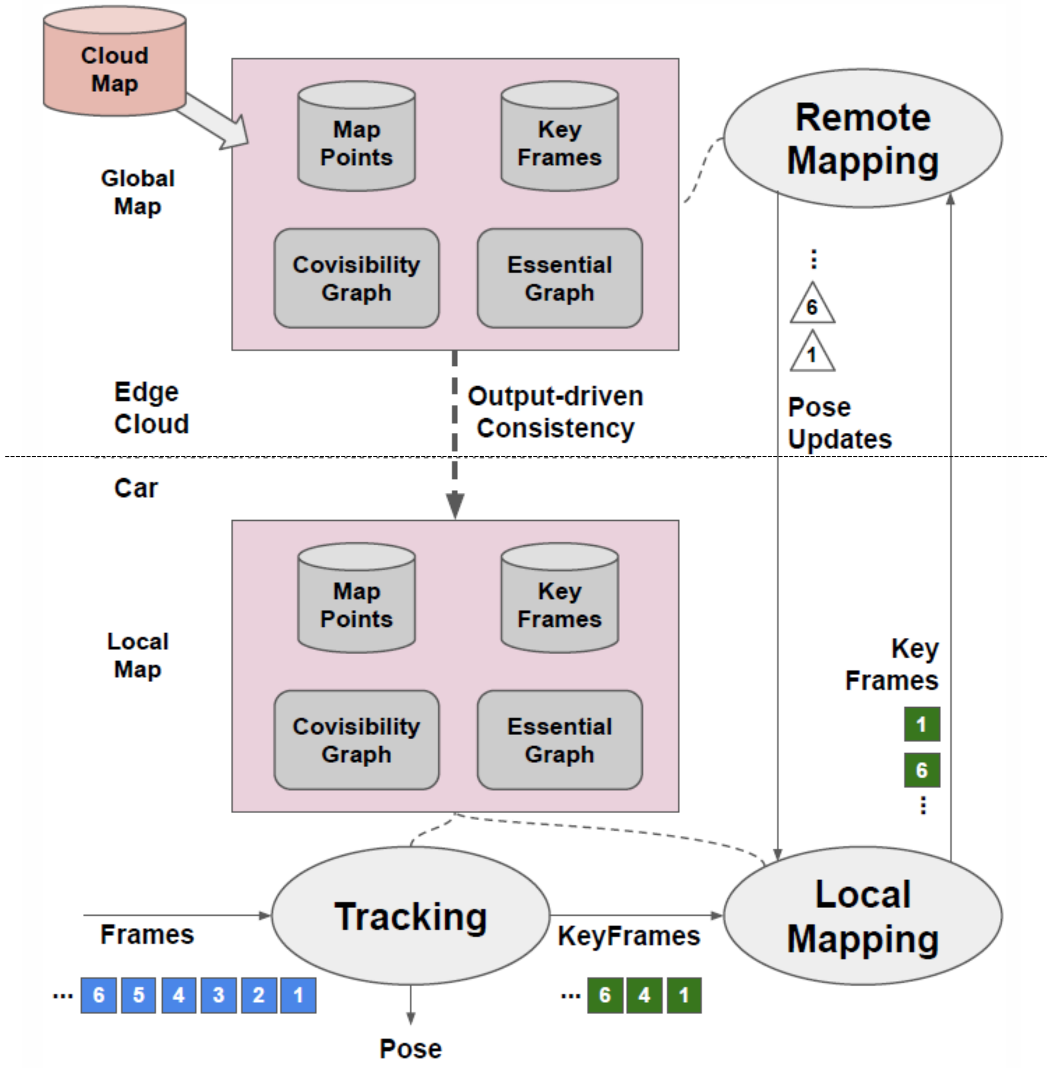
- Offloading is simplest option but is not practical
  - Run SLAM fully in cloud
  - Requires too much bandwidth
  - Highly susceptible to network delay
- Partitioning is effective if done right
  - Frequently used but fast tasks executed on vehicle
    - Tracking & Local Mapping Modules
  - Slow but infrequently used tasks executed in cloud
    - Loop Closing Module
  - Uses bandwidth more efficiently
  - Tolerant of network delay

ORB-SLAM's average performance on KITTI-05

Module	# of Frames	Avg. Time (s)
Tracking	2761	0.058
Local Mapping	725	0.168
Loop Closure	3	0.644

CloudSLAM System Design

- Loop Closing functionality moved into new Remote Mapping Module running in edge cloud
  - Reduces computation on vehicle while maintaining previous trip data to improve accuracy
- Map state is replicated: global map stored in cloud, local map stored on vehicle
  - Only recent data is relevant to Tracking & Local Mapping modules
- Challenges
  - Map state management
  - Limiting bandwidth usage
  - Maintaining accuracy



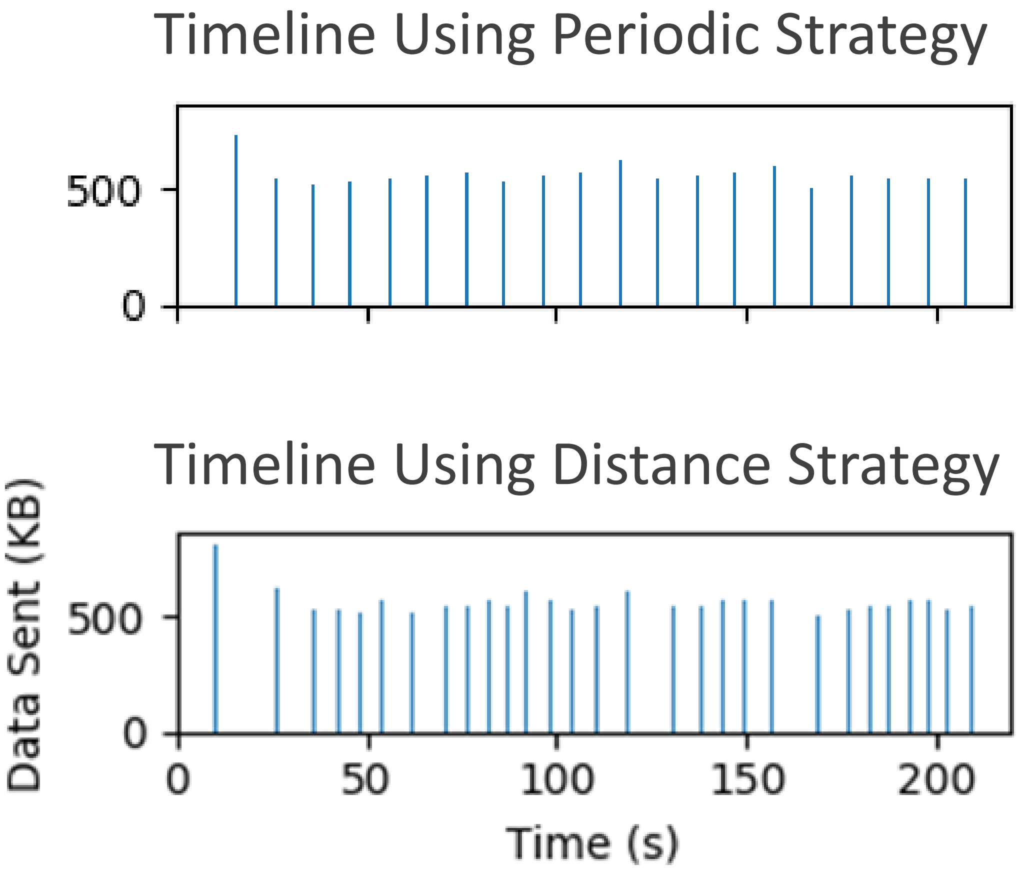
Map State Management

- ORB-SLAM's modules all read and write to the same complex data structures
  - Traditional consistency models not suitable because of bandwidth usage and/or delays
- Consistency requirements for local and global map are loose
  - ORB-SLAM execution is not repeatable
    - two executions of the same video input will generate different results
  - Construction of map is based on sensor data, which itself is noisy
- *Output-driven Consistency* designed to focus on our actual needs
  - What we really care about is consistency of the pose output
  - Send keyframes from vehicle to edge as necessary
  - Feedback applied to manage tradeoff between high accuracy & low bandwidth

Limiting Bandwidth Usage

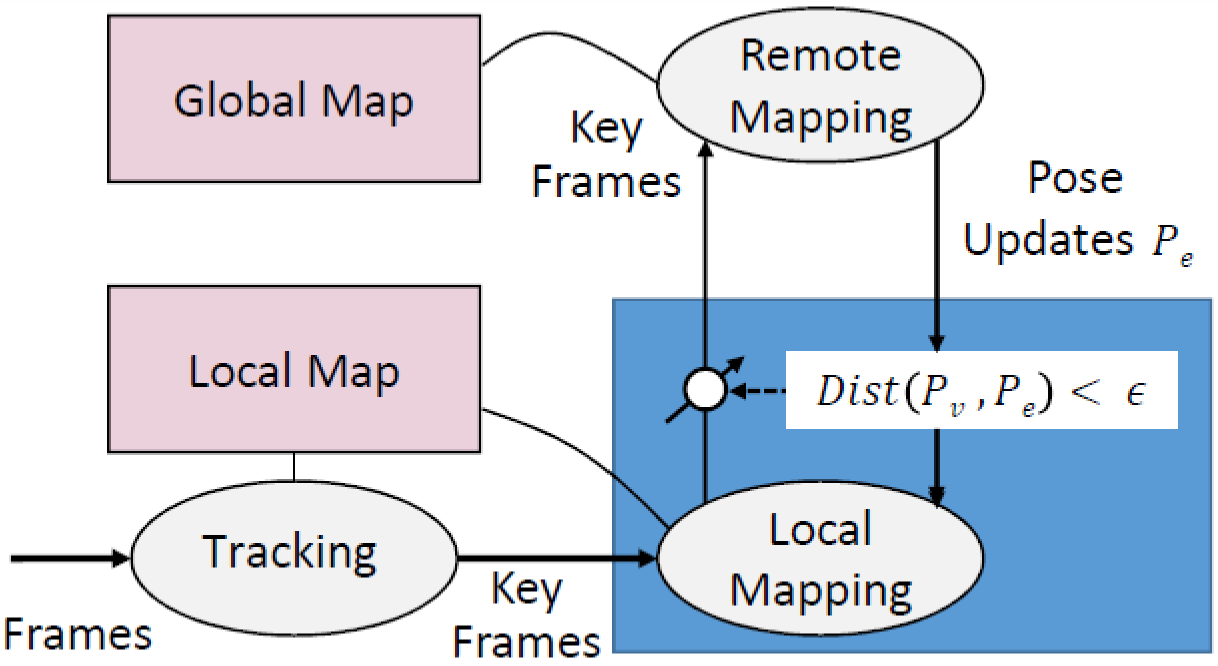
- Selectively sending keyframes reduces bandwidth consumption
  - Redundant information in consecutive images

- How to select which keyframes to send?
  - **Periodic Strategy** - send keyframes at a fixed time interval
    - For example, send keyframe once every 10 seconds
  - **Distance Strategy** - send keyframes at a fixed distance interval
    - For example, send keyframe once every 10 meters
    - Varies based on vehicle speed and therefore is more bandwidth efficient



Maintaining Accuracy

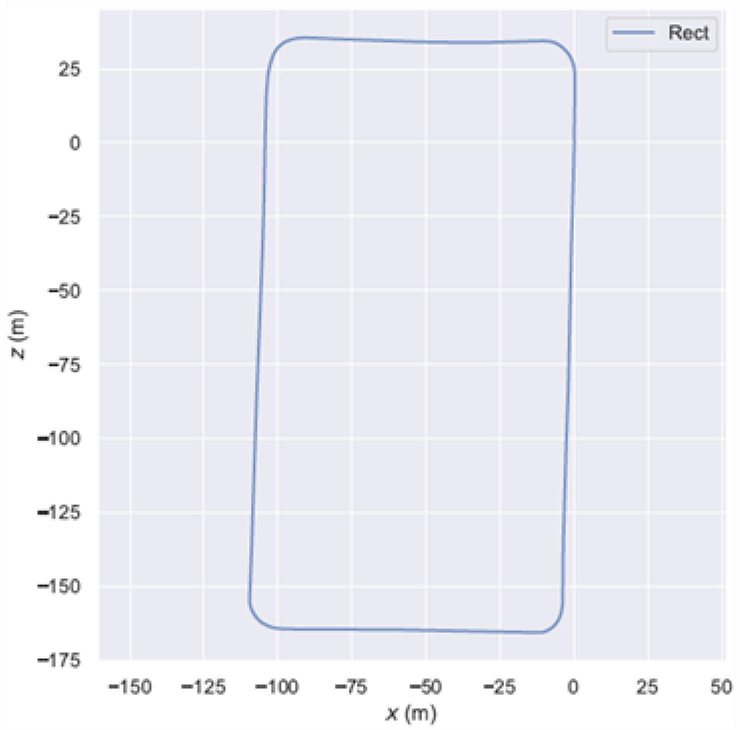
- **Adaptive Strategy** uses magnitude of pose corrections as an indicator of error in the pose output
  - Drives map consistency based on pose updates
    - If pose corrections are large, more keyframes are sent to improve consistency
  - Implemented as an extension of Distance Strategy
    - Dynamically tunes distance threshold based on pose correction magnitude
    - Multiplicative-increase, multiplicative-decrease



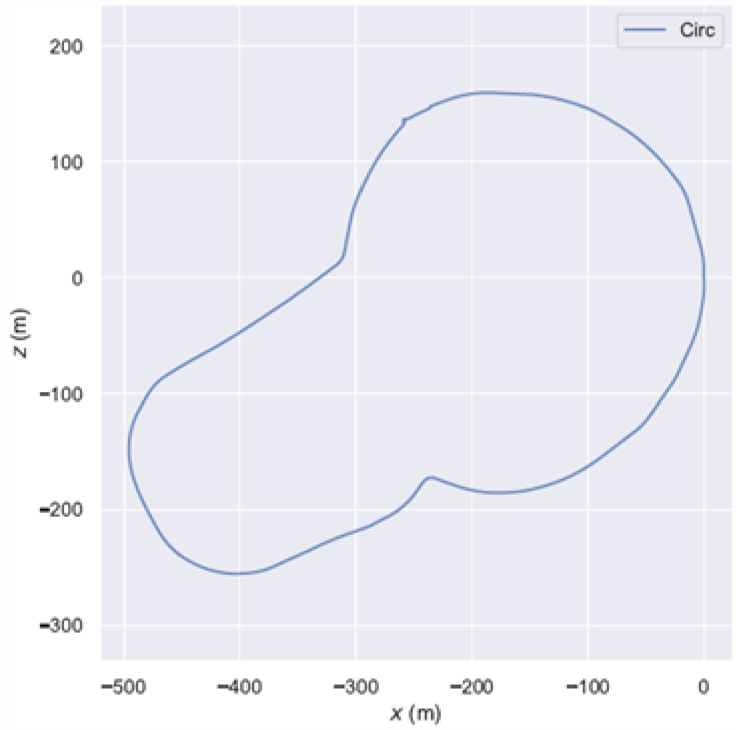
Evaluation Traces

- Rectangular Trace
  - Corporate campus
  - Duration: 128 secs
  - Top Speed: 15 mph

- Circular Trace
  - Suburban community
  - Duration: 200 secs
  - Top Speed: 24 mph



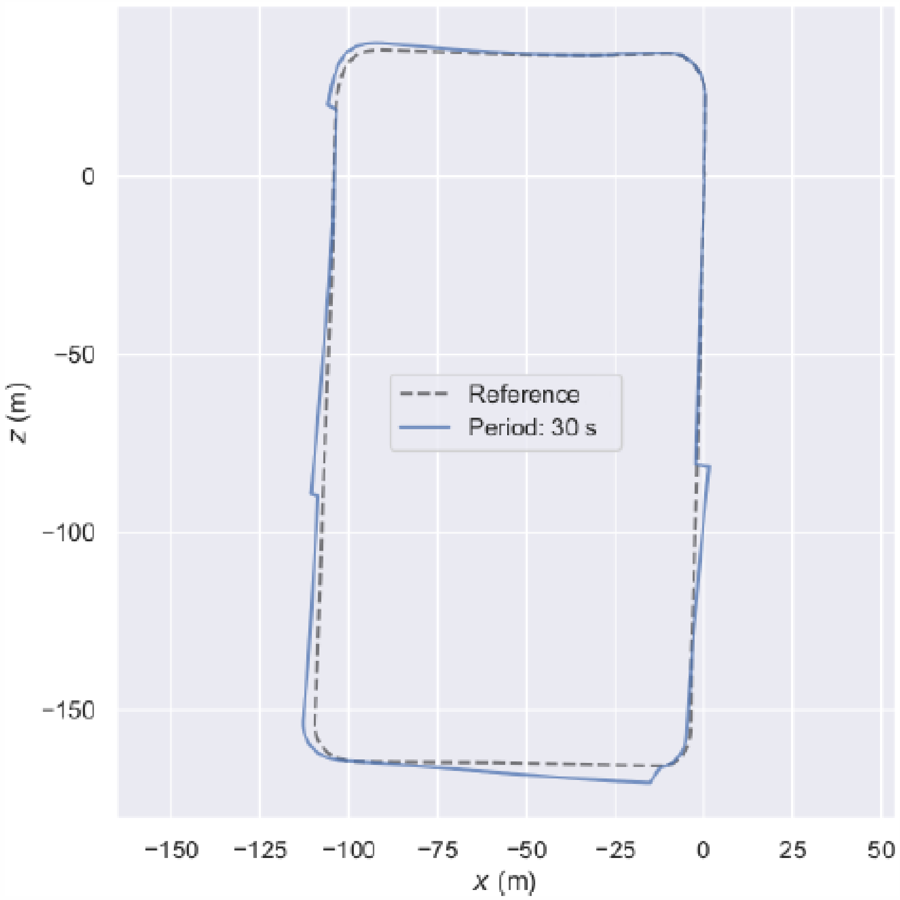
Rectangular Trace



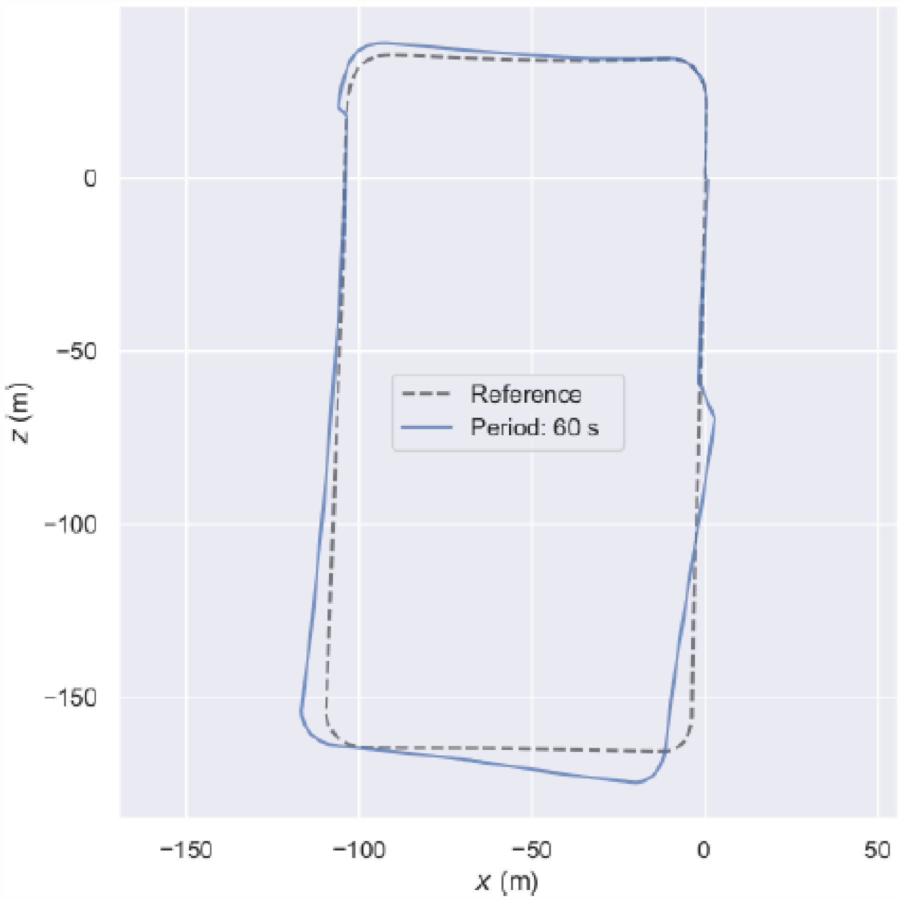
Circular Trace

CloudSLAM Output Using Periodic Strategy

- Error metric is root-mean-square error (RMSE)



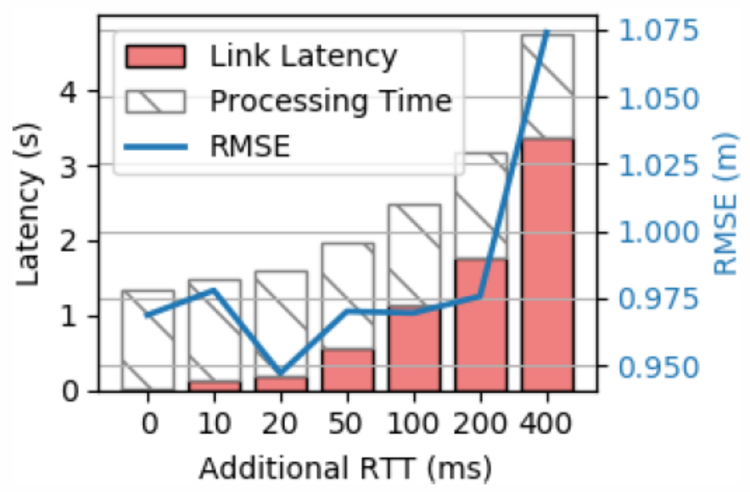
Period: 30s, RMSE: 2.12m



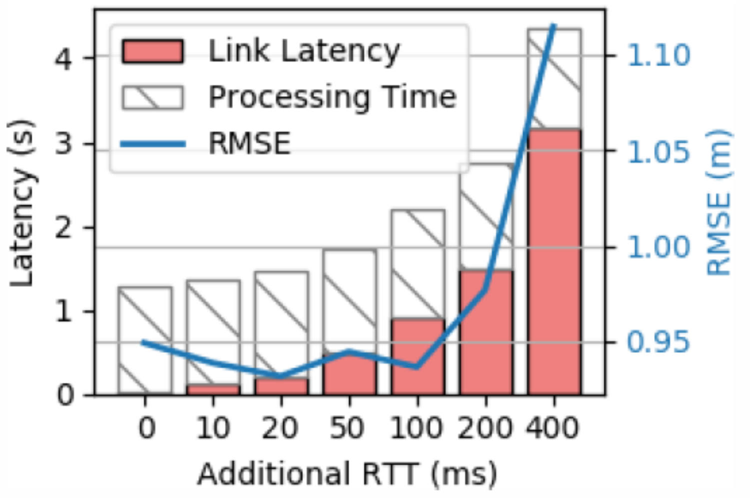
Period: 60s, RMSE: 6.22m

Impact of Link Latency

- CloudSLAM accuracy degrades as link latency becomes dominant portion of response time
- Need for low latency edge computing as opposed cloud computing



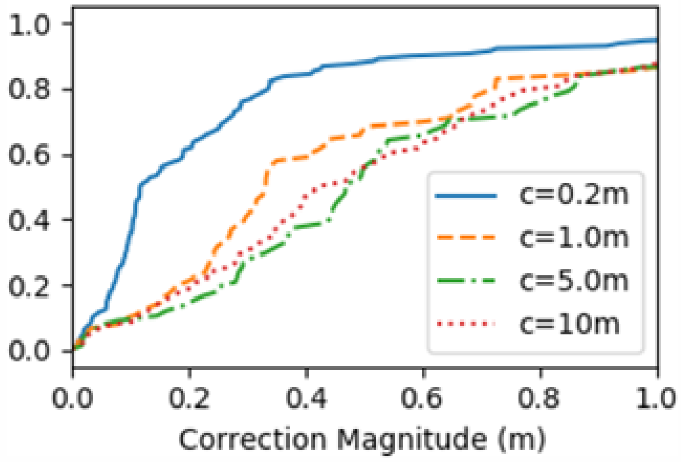
RTT impact on Periodic Strategy (10s)



RTT impact on Distance Strategy (20m)

Adaptive Strategy Performance

- If a pose correction's magnitude is above the pose correction threshold, then keyframe rate is increased. Otherwise, it is decreased.
- Sending more keyframes addresses drift more quickly, resulting in smaller pose corrections



CDF of pose correction sizes

c = pose correction threshold

Related Work

- Partition-based Offloading
  - M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan, "**Odessa: Enabling interactive perception applications on mobile devices**," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, (New York, NY, USA), p. 43–56, Association for Computing Machinery, 2011.
  - A. Ashok, P. Steenkiste, and F. Bai, "**Enabling vehicular applications using cloud services through adaptive computation offloading**," in *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*, MCS '15, (New York, NY, USA), p. 1–7, Association for Computing Machinery, 2015.
  - A. Ashok, P. Steenkiste, and F. Bai, "**Adaptive cloud offloading for vehicular applications**," in *Proceedings of IEEE Vehicular Networking Conference (VNC)*, (Piscataway, NJ), pp. 1–8, IEEE, December 2016.
- Edge-assisted SLAM
  - A. J. B. Ali, Z. S. Hashemifar, and K. Dantu, "**Edge-slam: Edge-assisted visual simultaneous localization and mapping**," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, MobiSys '20, (New York, NY, USA), p. 325–337, Association for Computing Machinery, 2020.
  - J. Xu, H. Cao, D. Li, K. Huang, C. Qian, L. Shangguan, and Z. Yang, "**Edge assisted mobile semantic visual slam**," in *IEEE Conference on Computer Communications (INFOCOM '20)*, IEEE, April 2020.

Conclusion

- CloudSLAM, an offloading architecture for stateful, latency-sensitive applications
- Output-driven Consistency, a mechanism for maintaining consistency between replicas that focuses on output instead of state
- Highlighted the need for access to edge computing resources with low link latency

Thank you!

kwame@cmu.edu