# CloudSLAM : Edge Offloading of stateful vehicular applications

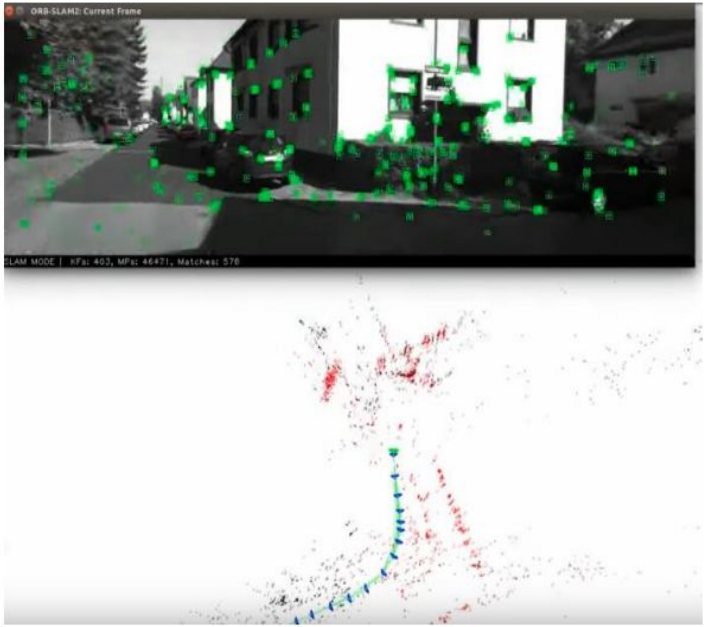
K Wright, Ashiwan Sivakumar,Peter Steenkiste,Bo Yu,Fan Bai
CMU, AT&T Research, General Motors

# Background

- Vehicular applications are becoming increasingly complex and resource hungry (e.g. autonomous driving)
- Running these applications entirely on vehicles is not feasible with increasing compute requirements of these applications.
- Complete offloading is also not feasible for all applications: Stringent latency requirements.
- This paper deals with one such application : SLAM (Simultaneous localisation and mapping).

# 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**.

# Challenges with SLAM on edge

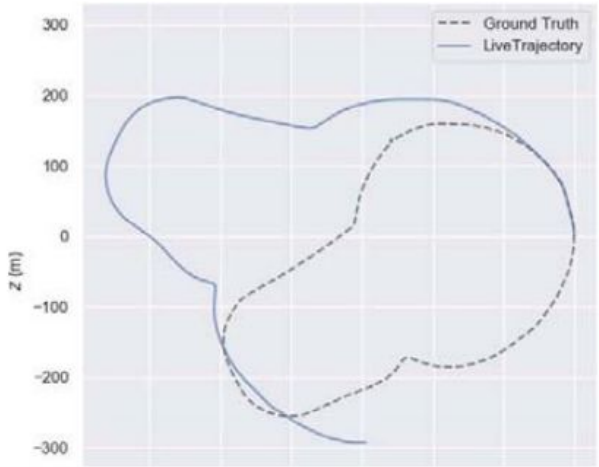
- High performance compute on vehicle can be costly
- Storage does not scale well with SLAM. (Every 1 mile of travel generates approx 200 MB of map data)
- Simplifying SLAM implementation reduces accuracy.

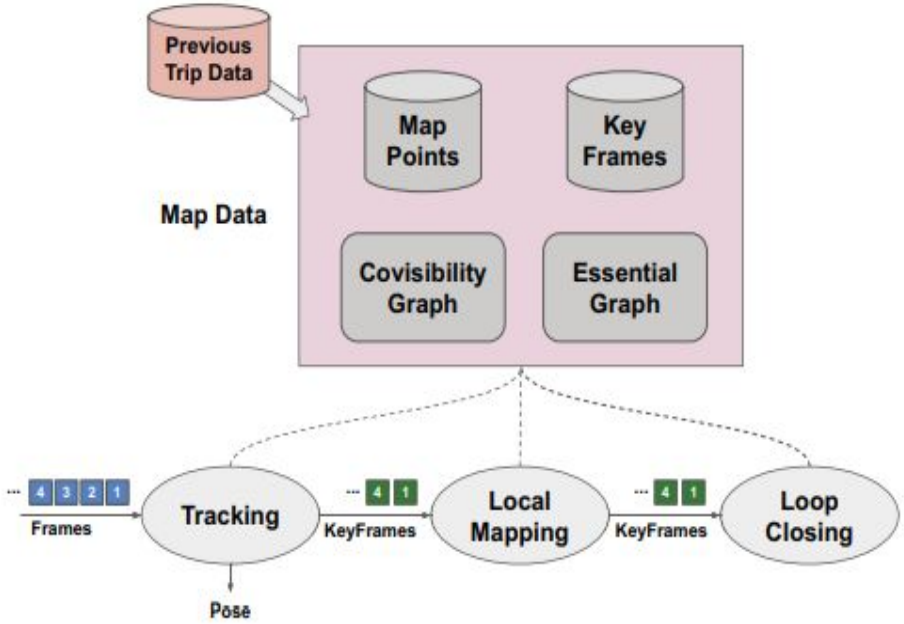
# CloudSLAM Goals

- Develop an offloading architecture for stateful, latency-sensitive applications.
- Utilize edge cloud resources to reduce CPU and memory load.
- Maintain accuracy similar to ORB-SLAM
- Minimize network usage

# ORB-Slam

- State of the art SLAM technique.
- 3 modules
- Previous trip data critical to achieving high accuracy



# Possible approaches

- Offloading completely
  - Simple but not very practical
  - Requires too much bandwidth
  - Highly susceptible to n/w delay

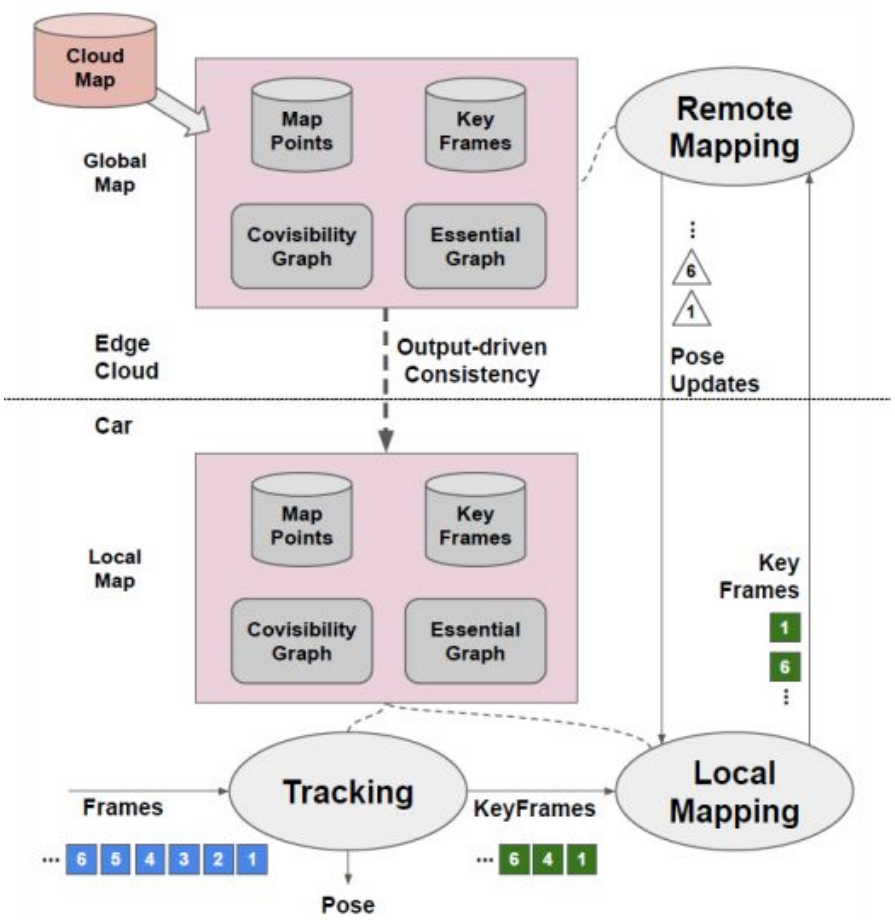
ORB-SLAM's average performance on KITTI-05

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

- Partitioning
  - Low latency tasks may be executed on vehicle
  - Slow but infrequent tasks may be executed on edge cloud
  - Use bandwidth more effectively
  - Tolerant of n/w delay

# CloudSLAM - 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
- Challenges
    - Map state management
    - Limiting bandwidth usage
    - Maintaining accuracy



# Map state management

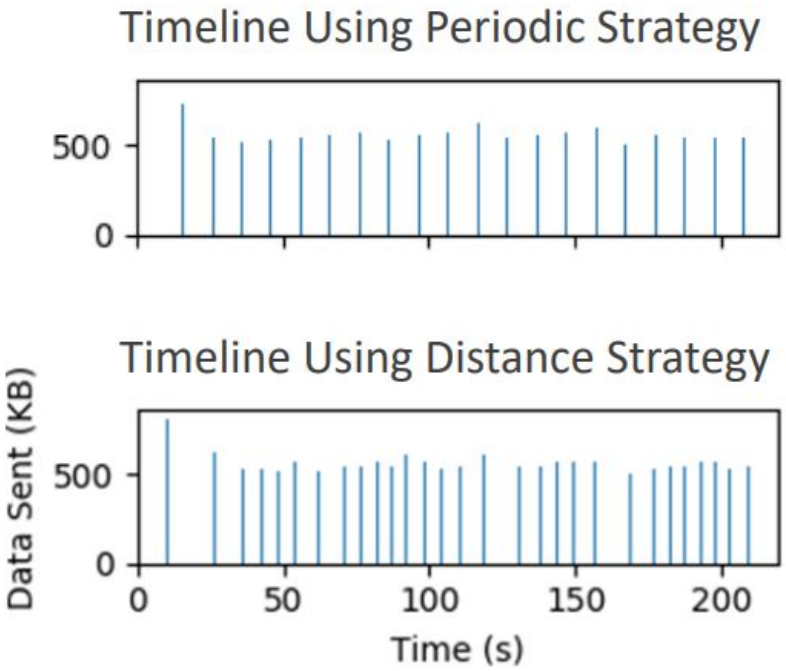
- ORB-SLAM’s modules all read and write to the same complex data structures Traditional consistency models not suitable because of bandwidth usage and/or delays.

- 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

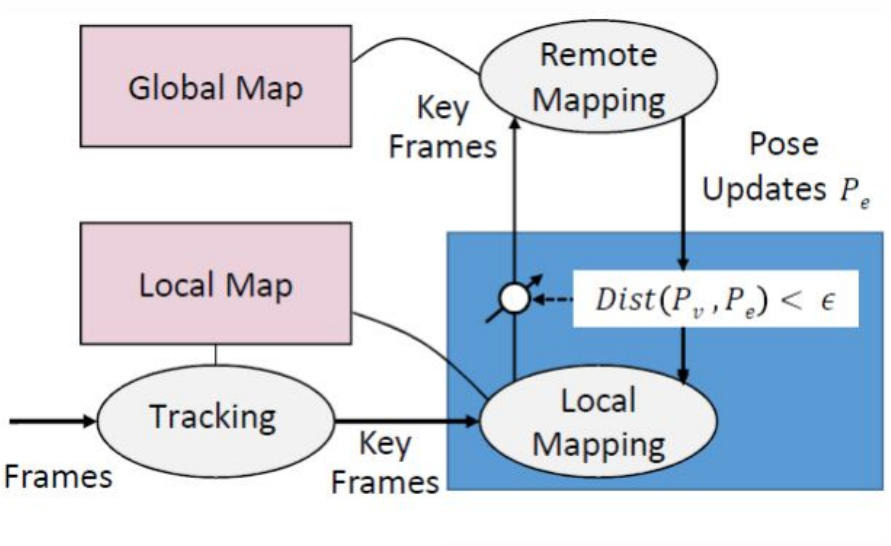
# Limiting Bandwidth usage

- 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
  - Distance Strategy - send keyframes at a fixed distance interval • For example, send keyframe once every 10 meters

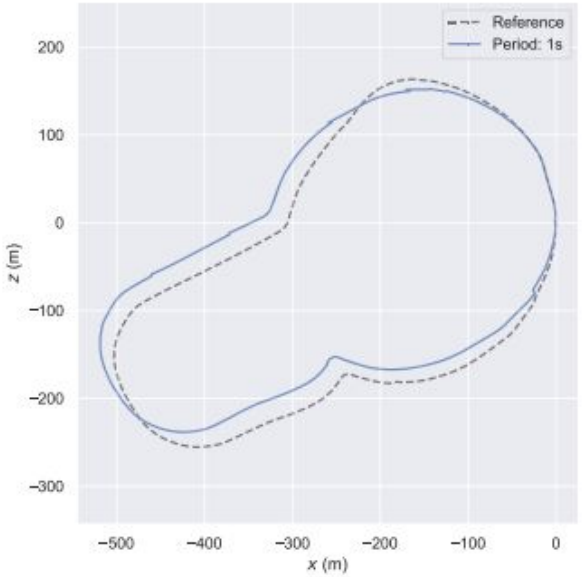


# Maintaining accuracy

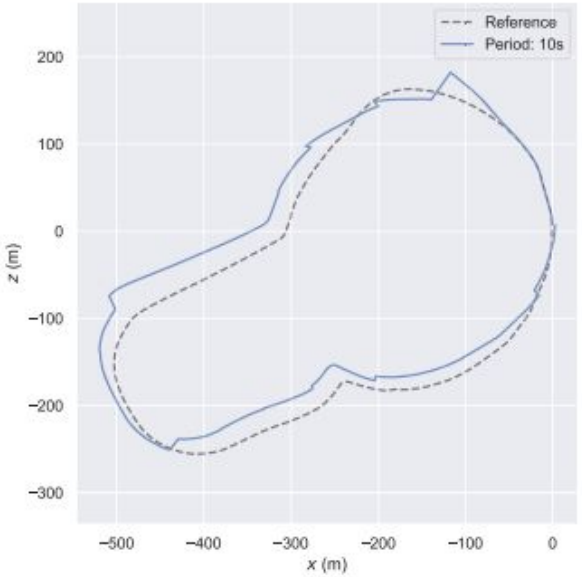
- Adaptive Strategy uses magnitude of pose correctness as an indicator of error in the pose o/p
- If pose corrections are large, more keyframes are sent to improve consistency
- Implemented as an extension of Distance Strategy
  - Dynamically tunes distance threshold based on pose correction magnitude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



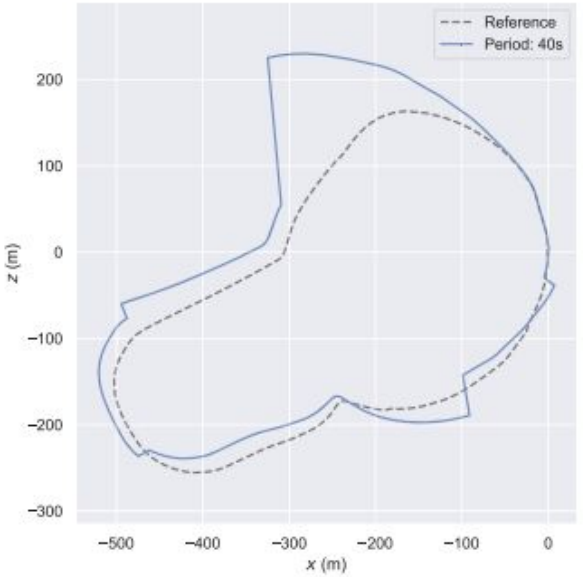
# Evaluation - Update freq vs rmse




(a) Period: 1s, RMSE: 17.88m

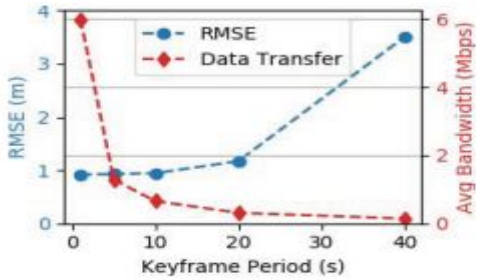



(b) Period: 10s, RMSE: 19.13m

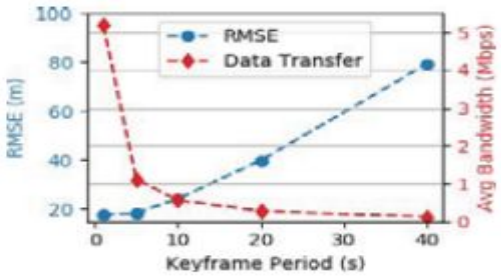



(c) Period: 40s, RMSE: 48.42m

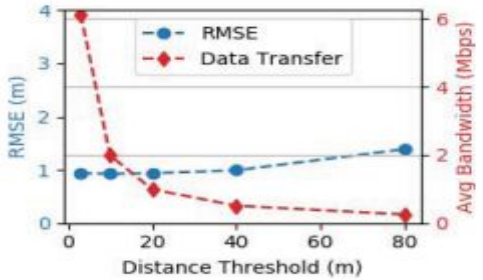
# Evaluation - Contd.



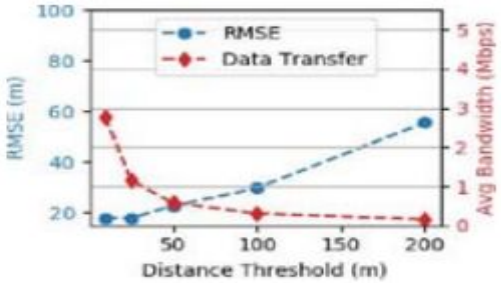

(a) Periodic Strategy on Rectangle Trace




(b) Periodic Strategy on Circular Trace




(c) Distance Strategy on Rectangle Trace

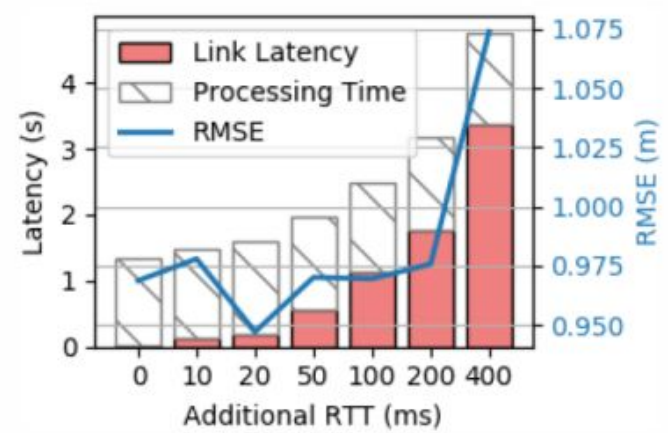



(d) Distance Strategy on Circular Trace

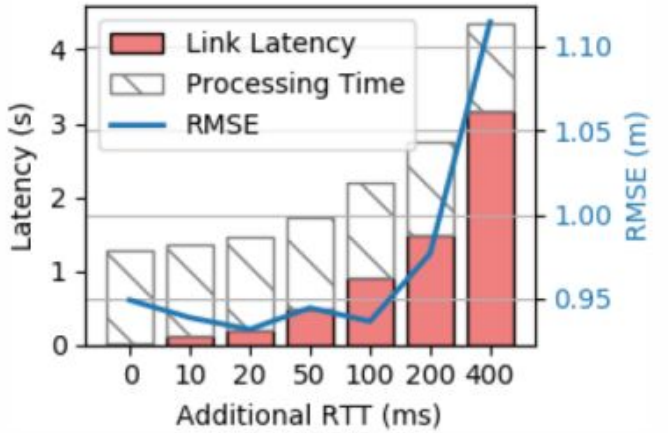
Fig. 11. RMSE and data transfer plots for each strategy and trace, averaged over five runs.

# 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)

# Discussion

- Sudden change in environment, can cause significant drift.
- Mechanism for identifying key frames is naive.
- Does not consider state management across edge cloud for long distance trips