SwarmMap: Scaling Up Real-time Collaborative Visual SLAM at the Edge

Jingao Xu*, Hao Cao*, Zheng Yang, Longfei Shangguan
Jialin Zhang, Xiaowu He, Yunhao Liu

* Co-primary author

USENIX NSDI 2022
Background

- **Visual SLAM (Simultaneous Localization And Mapping)**
  - Environment Reconstruction
  - Construction Inspection
  - Robot/Human Navigation

- **Collaborative Visual SLAM**
  - Search and Rescue
  - Inventory Automation
  - Multi-user AR
Background

• Typical Collaborative SLAM architecture
Background

- Scalability issues

Excessive bandwidth consumption

Large data storage overhead

Severe localization errors

As the number of agents growing, the overheads dramatically increase.
Scalability issues

- **C1: Map synchronization stresses the network bandwidth**
  - Map synchronization has two sessions

Diagram:
- Frame
- Map Point
- Path

- Cold-start
- Maintenance
- Deletion event
- Position adjustment
- The 2nd Trace
Scalability issues

- **C1: Map synchronization stresses the network bandwidth**
  - A minor modification on a single map element will spread to many other elements
Scalability issues

• **C1: Map synchronization stresses the network bandwidth**
  – A minor modification on a single map element will spread to many other elements

Previous works design compact map representations and transfer differences caused by map calibration, get only limited volume reduction in maintenance session.
Scalability issues

- C2: Map expansion exacerbates the memory footprint
  - Global map on edge server contains large redundancy
  - Device heterogeneity

Re-visit the same area  Different agents share the co-visible area
Scalability issues

- C3: FCFS-based job scheduling impairs the localization accuracy
  - Agents need the timely optimized map from the server
  - First-come, first-served (FCFS) scheduling exacerbates the localization error on task-sensitive agents
Key Insight

- Transferring operation logs but not modified data itself significantly reduces the bandwidth overhead
  - Map change on one side can be reproduced on the other side by solely transferring the map change operations

![Diagram showing map synchronization](Image)

Maps are Synchronized
Key Insight

• Data quality among different agents’ maps can be balanced by elements in co-visible areas
  – Leverage redundant elements to improve low-quality map segments
  – Compress the global map without compromising mapping quality
Key Insight

- Prioritize requests to offer a timely optimized map and earn higher information gain and
  - Prioritize agents with unstable tracking state
  - Prioritize agents with higher information gain for global map construction and optimization

Server processing queue

- Process firstly to prevent losing track
Our work

• **SwarmMap** is a framework design to **scale up** collaborative visual SLAM service in edge offloading setting.
S1: Mapit – Map Information Tracker

- A light-weight map information tracker to automate the operation tracking and reproducing on mobile and edge.
S1: Mapit – Map Information Tracker

- **Mapit add**
  - Generate an operation log entry

<table>
<thead>
<tr>
<th>Element Type</th>
<th>Element ID</th>
<th>Function Name</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>MapPoint</td>
<td>00003</td>
<td>SetBadFlag</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Operation log record
**S1: Mapit – Map Information Tracker**

- **Mapit aggregate**
  - Overwritten operation logs

<table>
<thead>
<tr>
<th>Element Type</th>
<th>Element ID</th>
<th>Function Name</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>MapPoint</td>
<td>00003</td>
<td>SetPosition</td>
<td>(0, 0, 0)</td>
</tr>
<tr>
<td>MapPoint</td>
<td>00003</td>
<td>SetPosition</td>
<td>(1, 2, 3)</td>
</tr>
</tbody>
</table>

Overwritten entries

Single entry
S1: Mapit – Map Information Tracker

- **Mapit aggregate**
- Stackable operation logs

<table>
<thead>
<tr>
<th>Element Type</th>
<th>Element ID</th>
<th>Function Name</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>MapPoint</td>
<td>00003</td>
<td>IncreaseVisible</td>
<td>1</td>
</tr>
<tr>
<td>MapPoint</td>
<td>00003</td>
<td>IncreaseVisible</td>
<td>1</td>
</tr>
</tbody>
</table>

N Stackable entries

Single entry
S2: MBP - Map Backbone Profiling

• Profiling-based Redundancy Elimination
  – Profile the quality of the map elements and evaluate each segment
  – Synthesize re-localizable keyframes in poor-evaluated areas
  – Compress without sacrificing the map with lower quality
S3: STS - SLAM-Specific Task Scheduling

- Guide the edge to strategically prioritize requests
  - Lost Handling Queue: Agent’s tracking state is marked as LOST
  - Lost Prevention Queue: Agent has a velocity burst and merely tracks few map-points
  - Map Enrichment Queue: Agents with stable running status
Experiments

- **Field studies**
  - a 22,927 sqft shopping mall
  - 12 agents

- **Trace-driven evaluations**
  - Public datasets: KITTI, EuRoC, TUM

- **Setup**
  - Edge: Intel(R) Xeon(R) CPU E5 64GB RAM
  - Wi-Fi 2.4 GHz and 5 GHz

- **Comparative systems**
  - CCM-SLAM[1]
  - Multi-UAV[2]
  - CarMap[3]

---


(4 smartphones, 4 drones, 4 mobile robots)
Experiments

• Overall Performance
  – Achieve the best tracking and localization performance in all scenarios
  – Degrade location errors are also significantly by > 40% when serving more than 10 agents, bound ATE and location error within 40cm and 1.4m even serving 20 agents
  – Total latency of SwarmMap is around 95ms and 105ms for 5 and 15 agents respectively, outperforming baselines by > 40% and 65%

(a) ATE comparison on public datasets.  (b) Location error comparison on our dataset.  (c) Map updating latency comparison
Experiments

• Resource Overhead Comparison
  – Reduce > 35%, 20%, 30%, 25%, 20% of network bandwidth requirement when serving 3, 5, 10, 15, 20 mobile agents compared with existing works
  – Save an average memory overhead of 2GB and 6GB when serving 5 and 15 agents
Experiments

• Ablation Study
  – Mapit saves nearly two times the bandwidth compared to CCM-SLAM and benchmark on all datasets
  – MBP reduces the original map size by almost half. MBP barely sacrifices the accuracy of the global map
  – STS maintains a stable service quality, and the lost percentage is within 4% in all scenarios
Conclusion

• We propose **SwarmMap**:  
  – A framework to scale up the real-time collaborative visual SLAM services at resource-constrained edge devices  
  – Provide additional system services to enhance system scalability which can be utilized by other collaborative visual SLAM systems  

• We implement SwarmMap compatible with the robotic operating system (ROS) and open-source it*

• We conduct extensive evaluations and a three-month pilot study demonstrate SwarmMap’s superior performance.

* Code is available at [https://github.com/MobiSense/SwarmMap](https://github.com/MobiSense/SwarmMap)
Thanks!

Q&A

Hao Cao
Tsinghua University
i.haocao@gmail.com