From Single-Agent to Decentralized Multi-Agent SLAM

Titus Cieslewski and Davide Scaramuzza
Our Research Areas

Visual-Inertial State Estimation (SVO)
[IJCV’11, PAMI’13, RSS’15, TRO’16]

Vision-based Navigation of Flying Robots
[AURO’12, RAM’14, JFR’15]

Deep Learning for End-to-End Navigation
[RAL’16]

Event-based Vision for Aggressive Flight
[IROS’3, ICRA’14, RSS’15]
Outline

- From single-robot SLAM to **centralized** multi-robot SLAM
  - [Forster 2013]

- **Decentralized** multi-robot SLAM: Place Recognition
  - [Cieslewski 2017]

- Decentralized collaboration with **version control**
  - [Cieslewski 2015]
From single-robot to multi-robot SLAM

T. Cieslewski – University of Zurich – From Single-Agent to Decentralized Multi-Agent SLAM
Single-robot SLAM

[Mur-Artal 2015 ORB-SLAM]
Single-robot SLAM components

- Our focus is on visual / visual-inertial SLAM
Multi-robot SLAM

T. Cieslewski – University of Zurich – From Single-Agent to Decentralized Multi-Agent SLAM
Centralized multi-robot SLAM
C. Forster, S. Lynen, L. Kneip, D. Scaramuzza: *Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles* IROS 2013
C. Forster, S. Lynen, L. Kneip, D. Scaramuzza: *Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles* IROS 2013
Distributed processing:

- Each MAV runs an onboard visual odometry and streams point features and relative poses (1 Mbit/s instead of 90 Mbit/s for full frames at 30 Hz)
- The ground station computes local maps for each MAV, detects overlaps, and merges different maps into a global map
Distributed processing:

- Each MAV runs an **onboard visual odometry** and streams point features and relative poses (1 Mbit/s instead of 90 Mbit/s for full frames at 30 Hz)
- The **ground station** computes local maps for each MAV, detects overlaps, and **merges different maps into global map**
System Overview

Distributed processing:

- Each MAV runs an **onboard visual odometry** and streams point features and relative poses (1 Mbit/s instead of 90 Mbit/s for full frames at 30 Hz)

- The **ground station** computes local maps for each MAV, detects overlaps, and merges different maps into global map

C. Forster, S. Lynen, L. Kneip, D. Scaramuzza: *Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles* IROS 2013
System Overview

Distributed processing:

- Each MAV runs an onboard visual odometry and streams point features and relative poses (1 Mbit/s instead of 90 Mbit/s for full frames at 30 Hz)

- The ground station computes local maps for each MAV, detects overlaps, and merges different maps into global map

C. Forster, S. Lynen, L. Kneip, D. Scaramuzza: Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles IROS 2013
Mapping on the Groundstation

C. Forster, S. Lynen, L. Kneip, D. Scaramuzza: *Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles* IROS 2013
Mapping on the Groundstation

C. Forster, S. Lynen, L. Kneip, D. Scaramuzza: Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles IROS 2013
C. Forster, S. Lynen, L. Kneip, D. Scaramuzza: *Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles* IROS 2013

Mapping on the Groundstation

MAV

Groundstation

Use motion estimate from Visual Odometry as prior
Mapping on the Groundstation

MAV

Refine pose w.r.t map with Bundle Adjustment

Groundstation

\[ C(x) = \frac{1}{2} \sum_i e_i(x)^T W_i e_i(x) \]

\[ \hat{x}^{LS} = \arg\min_x C(x), \]

\[ g2o \ [Kümmerle et al., ICRA'11] \]

C. Forster, S. Lynen, L. Kneip, D. Scaramuzza: Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles IROS 2013
1. Appearance-based Detection
   - Bag of Words image retrieval [Sivic et al., 2005]

2. Geometric Verification
   - 3-point RANSAC for point-cloud alignment

3-point algorithm [Kneip & Scaramuzza, CVPR’11]

C. Forster, S. Lynen, L. Kneip, D. Scaramuzza: Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles IROS 2013
Map Merging (multiple robots)

Outdoor flight
Summary: Centralized multi-robot SLAM

- Visual **odometry on-board** the individual robots
- **Ground station**
  - **Optimization** with bundle adjustment
  - Loop closure and map merging with bag-of-words **place recognition**
- At the heart of **active research**:  
  - [Morrison 2016 MOARSLAM]
  - [Schmuck 2017 Multi]
  - ...
Decentralized multi-robot SLAM

Sensor data → Odometry → Place recognition → Optimization → Trajectory and map

T. Cieslewski – University of Zurich – From Single-Agent to Decentralized Multi-Agent SLAM
Decentralized multi-robot SLAM

T. Cieslewski – University of Zurich – From Single-Agent to Decentralized Multi-Agent SLAM
Why decentralize?

- Scalability
- More practical field deployment
- Robustness to failure
- Privacy / militaristic considerations
How to decentralize?

T. Cieslewski – University of Zurich – From Single-Agent to Decentralized Multi-Agent SLAM
How to decentralize map optimization?

T. Cieslewski – University of Zurich – From Single-Agent to Decentralized Multi-Agent SLAM
Decentralized trajectory optimization

- **Filter- based:** [Grime 1994 Data], [Roumeliotis 2002 Distributed], [Nettleton 2003 Decentralised], [Carlone 2010 Rao], [Leung 2011 Distributed]

- **Graph- based:** [Kim 2010 Multiple], [Cunningham 2010/2013 DDF], [Paull 2015 Communication], [Choudhary 2016 Distributed]

- **Approach:** Each robot optimizes its own map, **exchange of condensed / marginalized information**

From [Choudhary 2016 Distributed]
How to decentralize place recognition?

T. Cieslewski, D. Scaramuzza: Efficient Decentralized Visual Place Recognition Using a Distributed Inverted Index RA-L 2017
Place recognition from other robot’s maps

- Relative localization with **visual place recognition** instead of direct observations

- **Advantages**
  - **More recall** → less redundancy e.g. in exploration
  - **No** special **hardware** needed

- **Disadvantages**
  - Relies on **connectivity**
  - More **bandwidth** required
  - Can be prone to **Perceptual Aliasing**

T. Cieslewski, D. Scaramuzza: *Efficient Decentralized Visual Place Recognition Using a Distributed Inverted Index* RA-L 2017
Decentralized visual PR: Query everyone?

T. Cieslewski, D. Scaramuzza: *Efficient Decentralized Visual Place Recognition Using a Distributed Inverted Index* RA-L 2017
We can do better
Distributed Hash Tables (DHTs)

- Developed by the distributed computing community in the early 2000s (e.g. [Stoica 2001])
- Efficient **Key-Value lookup** in a *distributed* map
- Key insight: **Deterministic assignment** of keys to peers: Report new data to and query only one peer

T. Cieslewski, D. Scaramuzza: *Efficient Decentralized Visual Place Recognition Using a Distributed Inverted Index* RA-L 2017
Distributed Hash Tables (DHTs)

T. Cieslewski, D. Scaramuzza: Efficient Decentralized Visual Place Recognition Using a Distributed Inverted Index RA-L 2017
Distributed Bag-of-Word Place Recognition

- Deterministically **assign Visual Words** to Robots using a DHT!
- A place query is now **split** into several partial queries

A visual representation of the process is shown, where visual words are assigned deterministically to robots, and place queries are split into partial queries and sent to different robots.
Results

- The amount of data exchanged is significantly reduced.
- In the paper, we discuss consequences in different network types.
- Because of a simplification in aggregation, recall is slightly affected.

![Graph showing amount of data exchanged vs number of robots](image1)

![Graph showing recall relative to centralized evaluation](image2)

T. Cieslewski, D. Scaramuzza: Efficient Decentralized Visual Place Recognition Using a Distributed Inverted Index RA-L 2017
What about full image descriptors?

- Using **NetVLAD** [Arandjelović 2016]
- Simpler: Not visual words, but *clusters of full image descriptors* assigned to robots

Image source: Yi Cao, Mathworks file exchange

T. Cieslewski, D. Scaramuzza: *Efficient Decentralized Visual Place Recognition From Full-Image Descriptors* Arxiv 2017
Full image descriptors: Preliminary results

- Performance relative to centralized place recognition similar to the Bag-of-Words approach
- Much **smaller queries** (0.5 VS 16kB) and **better absolute performance** with our Bag-of-Words implementation
- However: **Bad load balancing**: Inherent difference between training and testing distribution.

T. Cieslewski, D. Scaramuzza: *Efficient Decentralized Visual Place Recognition From Full-Image Descriptors* Arxiv 2017
Summary: Decentralized Place Recognition

- Place recognition from other robot’s maps
- Up to n-fold bandwidth reduction VS querying all robots
- Based on Bag-of-Words or full image descriptors

![Graph showing amount of data exchanged versus number of robots](image1)

![Graph showing recall relative to centralized evaluation](image2)

T. Cieslewski, D. Scaramuzza: Efficient Decentralized Visual Place Recognition Using a Distributed Inverted Index RA-L 2017
T. Cieslewski, D. Scaramuzza: Efficient Decentralized Visual Place Recognition From Full-Image Descriptors Arxiv 2017
Decentralized collaboration

- SLAM is not the only multi-robot task
- Can we make a **general framework** for multi-robot collaboration?
- **Shared state** with different levels of ownership
- How do humans collaborate on a shared state?
  - Version control

---

Decentralized version control for robots

- Like typical version control:
  - Optimistic Concurrency Control (checkout/commit)
  - Conflict handling (rule-based)

- Unlike typical version control:
  - Partial participation in the distributed state
  - Fully decentralized
  - Changes are implicitly pushed
  - Deals with time delays

- More features:
  - Decentralized lookup
Why we developed it

- We originally wanted to use it in decentralized multi-session SLAM
- Problem: Constraints propagate through the **entire graph**
- Map API better suited for **locally restricted** tasks
Do you see a use case?

- Decentralized version control for **your** robot teams
- C++
- We **open source** it!
- [https://github.com/ethz-asl/map_api](https://github.com/ethz-asl/map_api)

```cpp
// Example code

MAP_API_REVISION_PROTOBUF(proto::DataType);
enum Fields { kField }

descriptor.setTableName("my_table");
descriptor.addField<proto::DataType>(kField);
NetTable* my_table = map_api::NetTableManager::addTable(descriptor);

my_table->getChunk(chunk_id); // Assumed given.
map_api::Transaction transaction;
map_api::Revision revision = transaction.getById(
    my_table, item_id); // Assumed given.
proto::DataType data;
revision.get(kField, &data);
data.set_some_subvalue(data.some_subvalue() + 1);
revision.set(kField, data);
transaction.update(my_table, revision);
if (transaction.commit()) {
    LOG(INFO) << "Commit succeeded!";
}
```
Summary

- Overview of multi-robot SLAM
  - How a centralized system works
  - Centralized VS Decentralized
- Decentralized Place Recognition
  - Recognize from maps for higher recall
  - A DHT-based method for less bandwidth
- Decentralized general collaboration framework
  - Version control for robots.
  - Available as open-source code!