Minimizing Data Exchange in Decentralized Visual SLAM

Titus Cieslewski[1], Siddharth Choudhary[2] and Davide Scaramuzza[1]

Abstract— **Decentralized visual simultaneous localization and mapping (SLAM) is a powerful tool for multi-robot applications in environments where absolute positioning is not available. Being visual, it relies on cheap, lightweight and versatile cameras, and, being decentralized, it does not rely on communication to a central entity. In this workshop presentation, we present a decentralized visual SLAM system that is based on three core components: 1) Decentralized Visual Place Recognition (DVPR), which, using minimal data exchange, determines whether any robot observes a scene previously visited by any other robot. 2) Visual relative pose estimation, which provides the relative pose between robots based on a DVPR match. And 3) Decentralized Optimization, which resolves a globally consistent pose graph without transmitting the full map to a central place. With these components, it needs to transmit far less data than previously proposed decentralized visual SLAM systems. We characterize the resulting system and identify the main data transmission bottlenecks in its components.**

I. Introduction

Multi-robot simultaneous localization and mapping (SLAM) is an essential component of any team of robots operating in an absolute positioning system denied environment, as it relates the current state estimate of each robot to all present and past state estimates of all other robots. Because cameras are cheap, light-weight and versatile sensors, we seek to implement a visual SLAM system. Visual Multi-Robot SLAM can be solved in a centralized manner, where a single entity collects all data and solves SLAM for all robots, but that relies on a central entity to always be reachable, to never fail and to scale to the size of the robot team, in computation, memory, and bandwidth. Decentralized systems do not have this bottleneck, but are more challenging to implement. Visual SLAM systems typically consist of three components: 1) a visual odometry algorithm which provides an initial state estimate, 2) a place recognition system which is able to relate the currently observed scene to previously observed scenes, and 3) an optimization back-end which consistently integrates the place matches from the place recognition system with the full stream of state estimates. The end product is a map, and that map feeds back to place recognition and optimization, see Fig. 1. It is this feedback which makes decentralized SLAM challenging, especially if one is concerned about communication bandwidth: in state-of-the-art decentralized visual SLAM systems, all robots share all of their data with every other robot, resulting in a data transmission complexity that is quadratic with respect to the robot count. In contrast, our method has a lower complexity: Place recognition is linear in the robot count while relative pose estimation and optimization are linear in the size of the scene overlap between the robots.

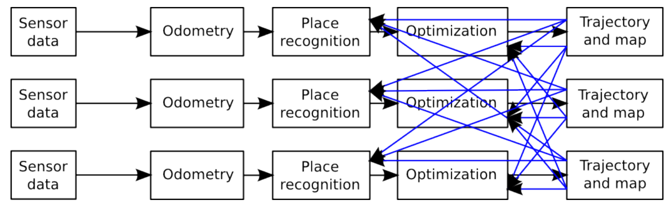

Fig. 1. High-level data flow in decentralized SLAM, illustrated for the case of three robots. Place recognition and optimization of each robot depend on trajectories and maps of every other robot. The main challenge we address is minimizing the data exchange due to this.

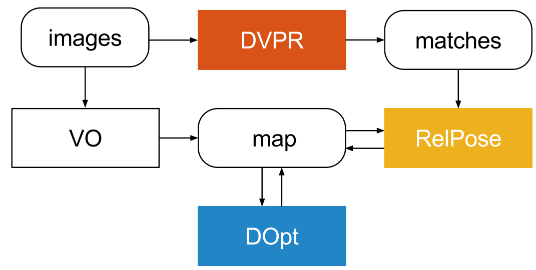

Fig. 2. Components and interactions in our system. The diagram shows the components that run on each robot. The colored blocks indicate components that communicate with other robots. Sharp corners indicate software modules, rounded corners indicate data. *DVPR*: Decentralized visual place recognition, *VO*: visual odometry, *RelPose*: Geometric verification and relative pose estimation, *DOpt*: Decentralized optimization.

II. Approach

The main components of our decentralized visual SLAM system [1] are visualized in Fig. 2 and are as follows:

1) **Visual Odometry** does the local mapping on each robot. Just like in single-robot visual SLAM, it is responsible for creating the initial guess for trajectory and map. It does not exchange any data with any other robots. We use the Visual Odometry provided by [2].
2) **Decentralized Visual Place Recognition** (DVPR) tells a robot whether a place it sees has been previously seen by another robot [3]. Rather than sending queries to all other robots or to a central entity, the workload is deterministically distributed among all robots. Since

[1]T. Cieslewski and D. Scaramuzza are with the Robotics and Perception Group, Dep. of Informatics, University of Zurich, and Dep. of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland—http://rpg.ifi.uzh.ch.

[2]S. Choudhary is with the College of Computing, Georgia Institute of Technology, Atlanta, GA, USA, siddharth.choudhary@gatech.edu

This research was funded by the DARPA FLA Program, the National Center of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation, the SNSF-ERC Starting Grant and the U.S. Army Research Lab (ARL) through the MAST-CTA project 329420.

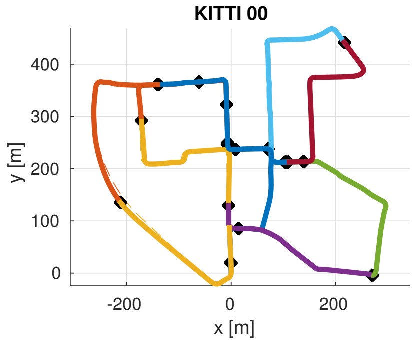

Fig. 3. Ten sub-trajectories of KITTI 00 after running our method. Each color represents an individual trajectory, place matches are marked in black, dashed lines indicate the ground truth.

we are using a single descriptor to describe the scene observed at a given time [4], this distribution can be achieved by clustering that descriptor space and assigning each cluster to a different robot.

3) **Relative Pose Estimation** provides a relative pose estimate between two robots for every DVPR match. This is achieved with P3P RANSAC and requires one robot to send 2d image features to the other robot. Also here, bandwidth can be saved by clustering the descriptor space of the 2d features [5]. Rather than sending full feature descriptors, features can be associated between robots based on the resulting Voronoi cell identifiers.
4) **Decentralized Pose Graph Optimization** optimizes the trajectory estimates of all robots such that the implicit (!) global map is consistent [6]. In order to achieve this, robots need to exchange only the estimates of poses that are connected to other robots through relative pose estimates. Hence, data transmission is linear with respect to the scene overlap between robots.

Since the bandwidth of both relative pose estimation and optimization directly depend on the amount of extracted relative poses, data usage of the overall system can be reduced by intentionally skipping relative pose estimations. This is justified under the assumption that the visual odometry is sufficiently consistent for short distances.

III. Results

To evaluate our system, we split KITTI 00 [7] into ten parts, one for each robot that we simulate. Fig. 3 shows the final global map obtained by the method when skipping relative pose estimations within $60m$ of each other. Its evolution over time is captured in Fig. 4. Here, *connected components* stand for connected components in the implicit global pose graph. At the very beginning, there are ten connected components, one for each individual robot map. As relative poses between robot maps are established, connected components merge. Initially, noisy relative pose estimates can increase the overall trajectory error of the global map, but as more relative poses are incorporated into the pose graph optimization, the error is reduced again. Fig. 5 shows the overall data transmission of the system over time. As we

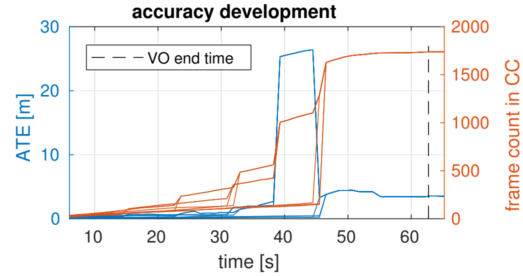

Fig. 4. Average trajectory error (ATE) and size of the connected component (CC) each of the ten robots is in. As the CCs of two robots converge, so does the line representing the frame count and ATE within these CC. Note that accuracy does not change significantly in the optimizations executed after visual odometry (VO) has ended – the state estimate reaches its final accuracy already before all the data is considered for optimization.

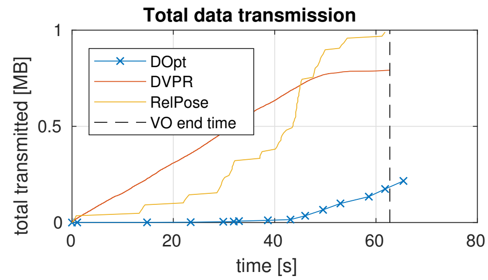

Fig. 5. Data transmission over time for the system components: decentralized optimization (DOpt), decentralized visual place recognition (DVPR) and relative pose estimation (RelPose).

can see, decentralized SLAM executed by ten robots on the KITTI 00 dataset only requires an overall data trasmission of two MB! The majority of data exchange of the system is due to place recognition and relative pose estimation. We are currently working on reducing data transmission due to relative pose estimation using machine learning, and will present any new results in the workshop.

References

[1] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-efficient decentralized visual SLAM," *IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018.

[2] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[3] T. Cieslewski and D. Scaramuzza, "Efficient decentralized visual place recognition from full-image descriptors," *IEEE Int. Symp. on Multi-Robot and Multi-Agent Sys.*, pp. 78–82, Dec. 2017.

[4] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Int. Conf. Comput. Vis. Pattern Recog. (CVPR)*, June 2016, pp. 5297–5307.

[5] D. Tardioli, E. Montijano, and A. R. Mosteo, "Visual data association in narrow-bandwidth networks," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Sept. 2015, pp. 2572–2577.

[6] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed trajectory estimation with privacy and communication constraints: A two-stage distributed Gauss-Seidel approach," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 5261–5268.

[7] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Research*, vol. 32, no. 11, pp. 1231–1237, 2013.