Collaborative Visual SLAM with Multiple MAVs

Christian Forster
Artificial Intelligence Laboratory
University of Zürich, Switzerland
Email: forster@ifi.uzh.ch

Simon Lynen, Laurent Kneip, and Roland Siegwart
Autonomous Systems Lab
ETH Zürich, Switzerland
Email: {slynen, kneipl, rsiegwart}@ethz.ch

The following work has been carried out within sFly, a European project that aims to implement swarms of autonomous Micro Aerial Vehicles (MAVs) collaboratively monitoring a disaster area. We present a framework for solving a core problem in this scenario, namely the collaborative localization and mapping with multiple MAVs in unknown environments. While solving the SLAM problem with multiple robots in parallel promises more efficient and extensive exploration of the environment, it also increases the computational and inter-robot communication load. Similar to [8] and [7], a straightforward solution to the illustrated problem consists of adding a centralized node to the system, which receives combined sensorial information from all swarm members in order to perform multi-robot SLAM. Concentrating all information at a central position allows optimal coordination of the MAVs. However, it also increases the communication load and, thus, reduces the MAVs’ work space. As presented in [11], the alternative consists of performing SLAM on each MAV individually and then share and merge the maps once the helicopters are within the communication range. This reduces the communication load, however, it leads to a substantial increase of the computational demand on the robots. In this paper, we present an intermediate solution to the multi-robot SLAM problem: multiple MAVs continuously stream sparse and preprocessed data to a central ground station where the Collaborative Structure from Motion (CSfM) system creates a 3D map of the environment in real-time and tracks the robots’ position in it.

As presented by Blösch et al. [11] and Weiss et al. [12], we use a single downlooking camera onboard our helicopters to perform incremental egomotion estimation. The overall structure of our multi-robot SLAM extension is presented in Figure 1. Each MAV tracks its position using an onboard monocular Visual Odometry (VO) algorithm. The VO algorithms operate as distributed preprocessors which stream only key-frames and corresponding incremental relative-pose estimates to the CSfM system on the ground station. The CSfM algorithm merges all the information into potentially common maps. By transmitting only features extracted from key-frames, the required bandwidth is kept at a considerably low level (~1 Mbit/s for 200 BRISK [6] features and 10 Hz key-frame rate) compared to streaming entire raw images (~86.6 Mbit/s for monochrome WVGA and 30 Hz framerate). This algorithmic layout leads to a system where mapping and MAV-motion estimation are clearly separated.

We use the minimal structure-from-motion-like VO presented in our previous work [3]. It is boosted in terms of robustness and efficiency by including incremental relative rotation priors obtained from the onboard IMU. This results in less complicated geometric algorithms that compute translation only (based on [4]). Furthermore, it also increases robustness against less favorable feature distributions in the image plane and dynamically moving scene objects.

On the ground station, the CSfM system gathers all VO’s key-frames and initially creates an individual map for each MAV. Locally, the map is constantly refined by using bundle adjustment [5]. The system’s external PlaceRecognizer module is based on the bag-of-words approach [9] [2], and detects, based upon the transmitted features, when a MAV observes a place that has already been visited by itself or by another MAV. The result is verified by efficient geometric comparison of the 3d-map structure at both places. Map overlaps trigger loop closures and map merges, respectively. Loop closures are optimized by a 7-DoF pose-graph relaxation according to [10]. On the other hand, if the maps of two MAVs are merged, their relative position is implicitly obtained. The corresponding MAVs then provide information to extend the same map. The key to real-time performance at this stage is the design of data-structures and processes that allow multiple threads to concurrently read and modify the same map. More specifically, the CSfM system assigns a separate thread for each MAV in order to asynchronously process their respective key-frames. Furthermore, the proposed internal environment representation allows additional maintenance threads to simultaneously process and improve the map in regions that are not currently relevant for the MAVs that operate on the map. To obtain a consistent map, a crucial point showed to be the estimation and correction of the scale-divergence between the drifting VO and the map on the ground station which is maintained by the mapping pipeline of the CSfM system.

The system is implemented in C++ and successfully tested on multiple datasets showing real-time performance in processing the data of two MAVs on a laptop with an Intel i7 dual core processor (2.8 GHz). The datasets are recorded with multiple sFly hexacopters navigating indoors with ground truth provided by a Vicon motion capture system. The CSfM system is able to reconstruct relative key-frame positions with an accuracy of 6 cm over summed MAV trajectories of 30 m (see Figure 2). Please refer to www.cforster.ch/csfm for videos of the experiments.
The CSfM system creates a separate thread for each Visual Odometry (VO) input. Initially, each thread creates its own map. However, the maps are merged when the PlaceRecognizer detects overlap between the two maps. Both threads then read and update the same map simultaneously.

Fig. 2: Examples of the map created by two MAVs and the CSfM system. The top image shows the state shortly after an overlap was detected by the PlaceRecognizer (red line). The maps are then merged (middle image). The bottom image illustrates the final result from the side.

References