Collaborating Low Cost Micro Aerial Vehicles: a Demonstration

Richard Williams, Boris Konev, and Frans Coenen

Department of Computer Science
University of Liverpool, United Kingdom

Abstract. In this paper we demonstrate our Distributed Collaborative Tracking and Mapping (DCTAM) system for collaborative localisation and mapping with teams of Micro-Aerial Vehicle's MAVs. DCTAM uses a distributed architecture which allows us to run both image capture and frame-to-frame tracking on-board the MAV while offloading the more computationally demanding tasks of map creation/refinement to an off-board computer. The low computational cost of the localisation components of our system allow us to run additional software on-board such as an Extended Kalman Filter (EKF) for full state estimation and a PID-based Position Controller. This allows us to demonstrate complete cooperative autonomous operation.

1 Introduction and Motivation

Autonomous aerial vehicles are becoming pervasive in many diverse application domains from search and rescue to aerial transportation. The small size and robust nature of Micro Aerial Vehicles (MAVs) mean they have numerous applications, particularly in indoor environments (for exploration or remote inspection tasks) where it becomes difficult to rely on external positions systems such as GPS and Motion Capture systems or carry heavy sensor payloads (e.g. Laser Rangefinders). Monocular vision-based localisation systems offer advantages as they provide a very light-weight, low power sensor solution. Much recent work has addressed the issue of localisation using monocular vision [**?**, **?**, **?**, 1, 2, 4]. However fewer works address the problem from a multi-robot perspective. In this paper we demonstrate a distributed framework for collaborate multi-robot localisation and mapping for teams of low-cost, light-weight (≈ 500 grams) MAV platforms. We demonstrate a complete state estimation and position control solution that allows teams of MAVs to perform autonomous, collaborative localisation and mapping tasks.

1.1 Related Work

The multi-robot SLAM problem has been previously explored for ground-based robots with range sensors (such as laser range-finders, stereo vision) [5–7]. There is comparatively less work on the use of monocular vision as the only extrospective sensor, or involving agents capable of omni-directional (6DOF) motion such as flying robots or hand-held devices (e.g. mobile phones). Foster et al. [8] introduce a centralised system in which each agent tracks their local position using a Visual Odometry (VO) algorithm and sends image features from selected keyframes to a centralised mapping server. Foster et al. report real-time performance with up to 3 MAVs. Riazuelo et al. [11] is the closest to the proposed approach in terms of components and architecture. Specifically they also build on Parallel Tracking and Mapping (PTAM) and use a distributed architecture, however they focus on multiple map merging using a RGB-D camera-based solution. In our work we focus more on cooperative navigation where robots start from the same location (a common assumption in most practical deployments). We assume all robots localise themselves within the global map before proceeding which allows them to perform cooperative tasks like exploration and robot-to-robot collision avoidance immediately without waiting for a map merge/rendezvous to occur. Additionally our work focuses on using RGB cameras only (we only use grey-scale images) which are more lightweight and consume less bandwidth than the RGB-D cameras. This allows our system to operate on low power ARM and ATOM processor-based MAV clients. Our previous work [12] featured a highly centralised approach with both tracking and mapping for each MAV running on a single ground-station computer and only image and sensor capture running on-board the MAVs. The sensitivity our previous approach to wireless interference and its limitations in terms of scalability (a maximum of 4 MAVs) motivated the development of the distributed approach we demonstrate in this paper.

2 System overview

Our goal is to enable cooperative multi-robot navigation tasks using light MAVs with very low on-board computing resources. Our DCTAM system is based on the PTAM system developed by Klien and Murray [13]. While Klien and Murray separate the tasks of real-time motion estimation and map creation/refinement into separate threads running on the same computer for tracking a hand-held camera we split these components into a distributed system where the tracking component operates on-board several MAVs in parallel and the map creation/refinement component runs on a more powerful ground-station computer.

An overview of the system is shown in Figure 1. Each MAV has a low-level flight controller responsible for attitude estimation, stabilization and motor control. The flight controller provides sensor data via a Serial link to the companion computer (Odroid U3/C1). The companion computer runs the main state estimation and control components of the system. The DCTAM Tracker is responsible for real-time camera pose estimation and selecting the keyframes to be used for global map construction. Pose estimates are fused with orientation and acceleration data from the flight controller to generate a complete state estimate. This state estimate is used by the High-level controller for real-time velocity and position control. The position controller sends raw control (pitch, roll, yaw and throttle) commands to the flight controller via the Serial link. The ground-station computer communicates with each MAV via a WiFi link and runs the DCTAM Mapper component which is responsible for map creation and optimization.Our framework has been implemented in C++ and integrated into the Robot Operating System (ROS) [15].

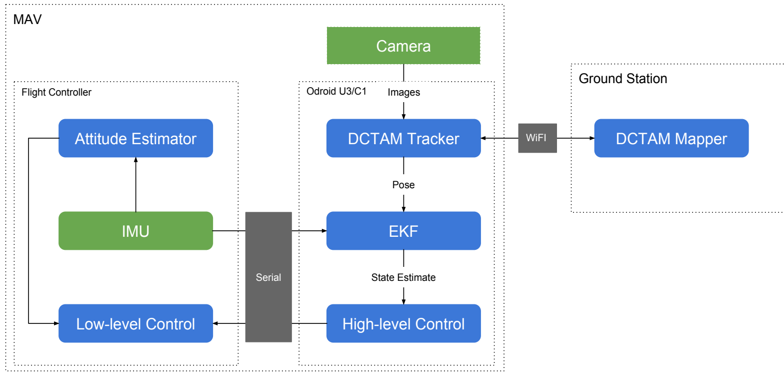

Fig. 1. System overview showing the components of the system(blue), sensors (green) and communication links (grey)

Fig. 2. AR. Drone/PX4-based MAV Platform

3 Hardware Platforms

We have developed two separate hardware platforms to demonstrate our system. While comprised of different hardware each has the required components to run our system i.e. flight controller, on-board computer with WiFi link and a single camera. The first demonstration platform is based on the popular AR.Drone[1] frame. We replaced the AR.Drone electronics with a PX4-based flight-control system consisting of a PX4 Flight Management Unit (PX4FMU) and AR.Drone adapter board (PX4IOAR) which interfaces with the AR.Drone motors. The on-board companion computer is an Odroid U3, an ARM-based single board computer with a 1.7 Ghz Quad-core processor with 2 GB of RAM. The MAV has a single MatrixVision mvBlueFOX-MLC200w 752x480 pixel monochrome camera fitted with a 100° wide-angle lens. The total cost of this platform is £750 and the average flight time is 10 minutes. Our second platform is based

[1] `http://ardrone2.parrot.com/`

around an open source 3D-printed quadcopter frame (The T4 Mini[2]). It uses low cost flight controller (Ardupilot APM2.6) and a low cost on-board computer Odroid C1. The C1 is an ARM-based single-board computer with a 1.5 Ghz Quad-core processor and 1 GB of RAM. The MAV also features the same MatrixVision monochrome camera. The total cost of this platform is £400 and the average flight time is 12 minutes.

4 Experimental Evaluation

We conducted experiments to verify the performance of our DCTAM system on the demonstration platforms. Figure 4 (left) shows the tracking times plotted against map size for the ODROID U3. The very short spikes in runtime coinciding with the arrival of a map update (resulting in an increase in map size). The average tracking time for the U3 was 0.01 milliseconds(ms) and 0.04 ms for the C1. Both platforms show good performance and the trackers demonstrate near constant-time performance on map sizes from small (5mx5m, $\approx$ 2000 map-points) indoor environments to large (20mx20m, $\approx$ 20000 map-points) outdoor environments. This is as a result of running the costly bundle adjustment procedure on the ground-station.

We show in Figure 4 the bandwidth requirements of a single MAV exploring an indoor environment; the final map for this experiment was 52 keyframes and 7240 map-points. We show that even with a very large(7240 points) map the required bandwidth remains low (42 Kb/s) for our system. Significantly lower than streaming colour video directly (28 MB/s) to the ground-station as in [12] and even lower than the 1 MB/s required by [11] who use the same library as our system but who send the full colour image captured by the camera. We instead send only a compressed grey-scale image (we use lossless PNG compression with a low compression rate to limit computation time) and are able to achieve a requirement of only 9 Kb/s for a single MAV and 42 Kb/s for a single MAV operating as part of a team. As stated previously, the additional bandwidth is required when sending keyframes to the other trackers in the team. To verify tracking

[2] `https://www.thingiverse.com/thing:408363`

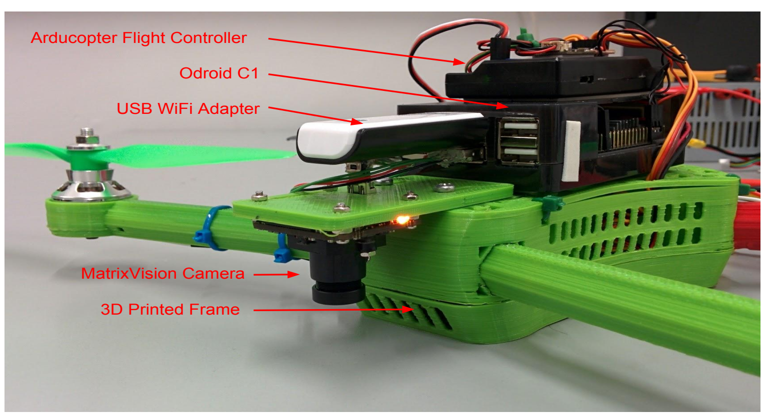

Fig. 3. 3D Printed MAV Platform

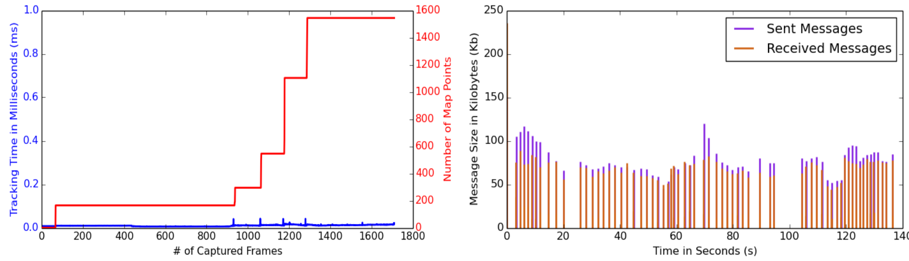

Fig. 4. Results showing tracking performance plotted against map size for the ODROID U3 (right). Bandwidth requirements for a single drone operating as part of a team. Recieved messages consist of keyframes and map-points generated by other MAV's (Left).

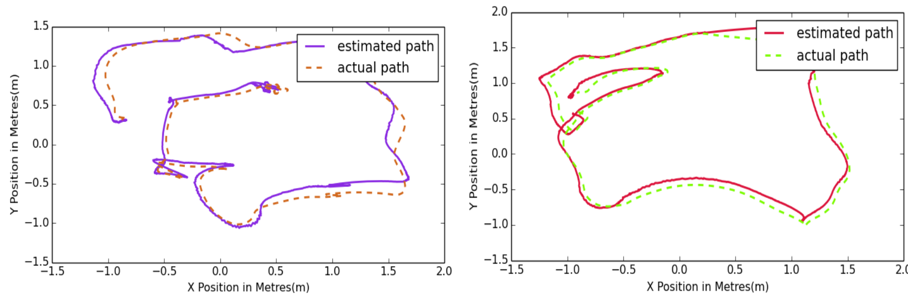

Fig. 5. Results of a physical experiment with a two MAVs navigating a 2mx2m area under manual control. The RMS Error for both MAVs is 0.13 metres(reft) and 0.15 metres (right).

performance we performed an experiment with two MAVs navigating in the same area simultaneously. Ground truth data for this experiment was captured using an OptiTrack[3] motion capture system. Position set-points for both MAVs we were adjusted manually using joystick controllers. Each MAV was flown on a (roughly) square path around a 2 metre by 2 metre area. Figure 5 shows the results of this experiment. For clarity we have separated the position plots for both MAVs and do not include vertical position. The RMS Error for both MAVs was 13cm (left) and 15cm(right) (including Z position error).

5 Demonstration

In this demonstration we show multiple MAVs performing a cooperative localisation task. Each MAV estimates its global position (using the DCTAM Tracker and EKF) with respect to a shared global map produced by the DCTAM Mapper. Each MAV is given a series of position goals, to explore their environment and expand the existing map. Examples of the expected output of the system are show in Fig. 5 and Fig 6 Additional details including videos and source code can be found at the following webpage: `http://cgi.csc.liv.ac.uk/~rmw/DCTAM.html`

[3] `http://www.optitrack.com/`

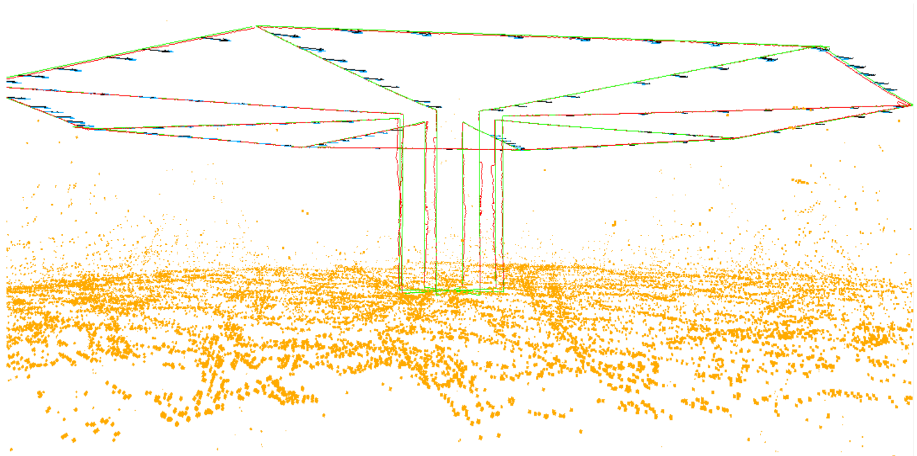

Fig. 6. The output of a large-scale simulated exploration experiment. Each MAV's estimated path is shown in red, the actual path in green and each 3D mappoint is shown in orange.

References

1. Weiss, S., Scaramuzza, D., Siegwart, R.: Monocular-slam–based navigation for autonomous micro helicopters in gps-denied environments. J. Field Robot. **28**(6) (2011) 854–874
2. Engel, J., Sturm, J., Cremers, D.: Camera-based navigation of a low-cost quadrocopter. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE (2012) 2815–2821
3. Wendel, A., Maurer, M., Graber, G., Pock, T., Bischof, H.: Dense reconstruction on-the-fly. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE
4. Forster, C., Pizzoli, M., Scaramuzza, D.: Svo: Fast semi-direct monocular visual odometry. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE (2014)
5. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: Computer Vision–ECCV 2014. Springer (2014) 834–849
6. Mur-Artal, R., Tardós, J.D.: Fast relocalisation and loop closing in keyframe-based slam. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE (2014)
7. Fox, D., Burgard, W., Kruppa, H., Thrun, S.: A probabilistic approach to collaborative multi-robot localization. Autonomous robots **8**(3) (2000) 325–344
8. Özkucur, N.E., Akın, H.L.: Cooperative multi-robot map merging using fast-slam. In: RoboCup 2009: Robot Soccer World Cup XIII. Springer (2010) 449–460
9. Carlone, L., Ng, M.K., Du, J., Bona, B., Indri, M.: Rao-blackwellized particle filters multi robot slam with unknown initial correspondences and limited communication. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE (2010) 243–249
10. Forster, C., Lynen, S., Kneip, L., Scaramuzza, D.: Collaborative monocular slam with multiple micro aerial vehicles. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, IEEE (2013) 3962–3970
11. Riazuelo, L., Civera, J., Montiel, J.: C 2 tam: A cloud framework for cooperative tracking and mapping. Robotics and Autonomous Systems **62**(4) (2014) 401–413
12. Williams, R., Konev, B., Coenen, F.: Multi-agent environment exploration with ar. drones. In: Advances in Autonomous Robotics Systems. Springer (2014) 60–71

13. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on, IEEE (2007) 225–234
14. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA workshop on open source software. Volume 3. (2009)