Graz Griffins' Solution to the European Robotics Challenges 2014

J. Pestana[1,3], R. Prettenthaler[1], T. Holzmann[1], D. Muschick[2]*, C. Mostegel[1], F. Fraundorfer[1] and H. Bischof[1]

***Abstract*— An important focus of current research in the field of Micro Aerial Vehicles (MAVs) is to increase the safety of their operation in general unstructured environments. An example of a real-world application is visual inspection of industry infrastructure, which can be greatly facilitated by autonomous multicopters. Currently, active research is pursued to improve real-time vision-based localization and navigation algorithms. In this context, the goal of Challenge 3 of the EuRoC 2014[4] Simulation Contest was a fair comparison of algorithms in a realistic setup which also respected the computational restrictions onboard an MAV. The evaluation separated the problem of autonomous navigation into four tasks: visual-inertial localization, visual-inertial mapping, control and state estimation, and trajectory planning. This EuRoC challenge attracted the participation of 21 important European institutions. This paper describes the solution of our team, the Graz Griffins, to all tasks of the challenge and presents the achieved results.**

I. Vision-Based Localization and Mapping

The first track of the simulation contest was split into the tasks of localization and mapping. A robust solution for both tasks is essential for a safe navigation in GPS-denied environments as they form the basis for controlling and trajectory planning respectively.

A. Localization

In this task, the goal was to localize the MAV using stereo images and synchronized IMU data only. The stereo images had a resolution of 752x480 pixels each and were acquired with a baseline of 11 cm and a framerate of 20 Hz. The implemented solution had to run on a low-end CPU (similar to a CPU onboard an MAV) in real-time. The results were evaluated on three datasets with varying difficulty (see Fig. 1) in terms of speed and local accuracy.

We used a sparse, keypoint-based approach which uses a combination of blob and corner detectors for keypoint extraction. First, feature points uniformly distributed over the whole image are selected. Next, quad matching is performed, where feature points of the current and previous stereo pair are matched. Finally, egomotion estimation is done by minimizing the reprojection error using Gauss-Newton optimization. We used *libviso2* [3] for our solution, a highly optimized visual odometry library.

*The first four authors contributed equally to this work.

[1]Institute for Computer Graphics and Vision and [2]Institute of Automation and Control, Graz University of Technology; and [3]CVG, Centro de Automática y Robótica (CAR, CSIC-UPM)
{pestana,holzmann,mostegel}@icg.tugraz.at
{daniel.muschick,rudolf.prettenthaler}@tugraz.at
{fraundorfer,bischof}@icg.tugraz.at

This project has partially been supported by the Austrian Science Fund (FWF) in the project V-MAV (I-1537), and a JAEPre (CSIC) scholarship.

[4]http://www.euroc-project.eu/

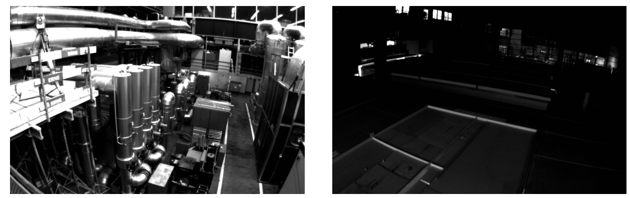

Fig. 1. Input data for the localization task. *Left:* Image from the simple dataset. *Right:* Image from the difficult dataset. In comparison to the left image, the right image includes poorly textured parts, reflecting surfaces, over- and underexposed regions and more motion blur.

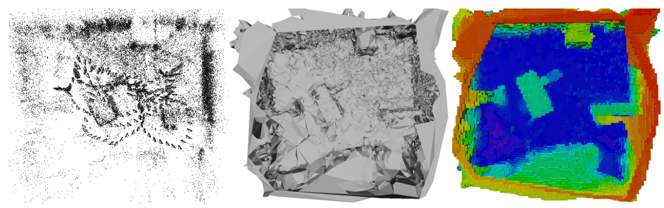

Fig. 2. Mapping process. *Left:* 3D points and their keyframe camera poses. *Middle:* Constructed mesh. *Right:* Evaluated occupancy grid (color coded by scene height).

B. Mapping

To successfully detect obstacles and circumnavigate them, an accurate reconstruction of the environment is needed. The goal of this task was to generate an occupancy grid of high accuracy in a limited time frame.

For our solution we only process frames from the stereo stream whose pose change to the previously selected keyframe exceeds a given threshold. From these keyframes we collect the sparse features (100 to 120) that are extracted and matched using *libviso2* [3]. For these features we triangulate 3D points and store them in a global point cloud with visibility information. After receiving the last frame, we put all stored data into a multi-view meshing algorithm based on [5]. The generated mesh is then smoothed and converted to an occupancy grid for evaluation. An example mapping process can be seen in Fig. 2.

C. Results

The final evaluation for all participants was performed on a computer with a Dual Core i7 @ 1.73 GHz using three different datasets for each task.

For the localization task, the local accuracy is evaluated by computing the translational error as defined in the KITTI vision benchmark suite [1]. Over all datasets, we reach a mean translational relative error of 2.5 % and a mean runtime of 48 ms per frame.

For the mapping task, the datasets contained a stereo stream and the full 6DoF poses captured by a Vicon system.

With increasing difficulty, the motion of the sensor changed from smooth motion to a jerky up and down movement with a lot of rotational change only. In addition, the illumination changed frequently and the captured elements consisted of fine parts that were challenging to reconstruct (e.g. a ladder). For scoring, the accuracy is calculated using the Matthews correlation coefficient (MCC). Our solution obtains an average MCC score of 0.85 on the final evaluation datasets. An MCC score of 1.0 would indicate a perfect reconstruction.

II. State Estimation, Control and Navigation

The second track aimed at the development of a control framework to enable the MAV to navigate through the environment fast and safely. For this purpose, a simulation environment was provided by the EuRoC organizers where the hexacopter MAV dynamics were simulated in ROS/Gazebo.

The tasks' difficulty increased gradually from simple hovering to collision-free point-to-point navigation in a simulated industry environment (see Fig. 3). The evaluation included the performance under influence of constant wind, wind gusts as well as switching sensors.

A. State Estimation and Control

For state estimation, the available sensor data is a 6DoF pose estimate from an onboard virtual vision system (the data is provided at $10\,\mathrm{Hz}$ and with $100\,\mathrm{ms}$ delay), as well as IMU data (accelerations and angular velocities) at $100\,\mathrm{Hz}$ and with negligible delay, but slowly time-varying bias.

During flight, the position and orientation are tracked using a KALMAN-filter–like procedure based on a discretized version of [7]: the IMU sensor data are integrated using EULER discretization (*prediction* step); when an (outdated) pose information arrives, it is merged with an old pose estimate (*correction* step) and all interim IMU data is reapplied to obtain a current estimate. Orientation estimates are merged by turning partly around the relative rotation axis. The corresponding weights are established *a priori* as the steady-state solution of an Extended Kalman Filter simulation.

For control, a quasi-static feedback linearization controller with feedforward control similar to [2] was implemented. First, the vertical dynamics are used to parametrize the thrust; then, the planar dynamics are linearized using the torques as input. With this controller, the dynamics around a given trajectory in space can be stabilized via pole placement using linear state feedback; an additional PI-controller is necessary to compensate for external influences like wind.

The trajectory is calculated online and consists of a point list together with timing information. A quintic spline is fitted to this list to obtain smooth derivatives up to the fourth order, guaranteeing jerk and snap free trajectories.

B. Trajectory planning

Whenever a new goal position is received, a new path is delivered to the controller. In order to allow fast and safe navigation the calculated path should stay away from obstacles, be smooth and incorporate a speed plan.

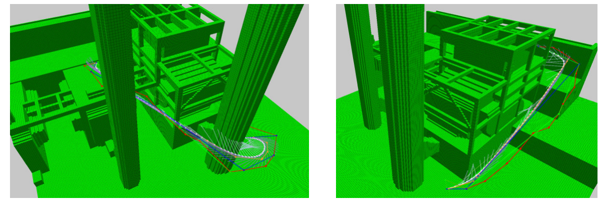

Fig. 3. Industrial environment of size $50\,\mathrm{m}\times50\,\mathrm{m}\times40\,\mathrm{m}$. A typical planned trajectory is shown: The output of the *PRMStar* algorithm (red) is consecutively shortened (green-blue-orange-white).

First, the path that minimizes a cost function is planned. This function penalizes proximity to obstacles, length and unnecessary changes in altitude. Limiting the cost increase, the raw output path from the planning algorithm (shown in red in Fig. 3) is shortened (white). Finally, a speed plan is calculated based on the path curvature.

The map is static and provided as an Octomap [4]. In order to take advantage of the environment's staticity a Probabilistic Roadmap (PRM) based algorithm was selected, the *PRMStar* implementation from the *OMPL* library [8]. The roadmap and an obstacle proximity map are precalculated prior to the mission. For the latter the *dynamicEDT3D* library [6] is used.

C. Results

The developed control framework achieves a position RMS error of $0.055\,\mathrm{m}$ and an angular velocity RMS error of $0.087\,\mathrm{rad/s}$ in stationary hovering. The simulated sensor uncertainties are typical of a multicopter such as the Asctec Firefly. The controlled MAV is able to reject constant and variable wind disturbances in under four seconds.

Paths of $35\,\mathrm{m}$ are planned in $0.75\,\mathrm{s}$ and can be safely executed in $7.55\,\mathrm{s}$ to $8.8\,\mathrm{s}$ with average speeds of $4.2\,\mathrm{m/s}$ and peak speeds of $7.70\,\mathrm{m/s}$.

III. Conclusions

Our solution to EuRoC 2014 Challenge 3 Simulation Contest earned the 6$^{\text{th}}$ position out of 21 teams. Although the developed algorithms are a combination of existing techniques, this work demonstrates their applicability to MAVs and their suitability to run on low-end on-board computers.

References

[1] The kitti vision benchmark suite, 2015. `http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php`, 26.03.2015.

[2] O. Fritsch, P. D. Monte, M. Buhl, and B. Lohmann. Quasi-static feedback linearization for the translational dynamics of a quadrotor helicopter. In *American Control Conferenc*, June 2012.

[3] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposion*, 2011.

[4] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[5] P. Labatut, J.-P. Pons, and R. Keriven. Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts. In *ICCV*, pages 1–8. IEEE, 2007.

[6] B. Lau, C. Sprunk, and W. Burgard. Efficient grid-based spatial representations for robot navigation in dynamic environments. *Elsevier RAS2013*.

[7] R. Mahony, T. Hamel, and J.-M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE TAC*, 2008.

[8] I. A. Şucan, M. Moll, and L. E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, December 2012.