Vision-Controlled Micro Flying Robots: from System Design to Autonomous Navigation and Mapping in GPS-denied Environments

D. Scaramuzza, M.C. Achtelik, L. Doitsidis, F. Fraundorfer, E.B. Kosmatopoulos, A. Martinelli, M.W. Achtelik, M. Chli, S.A. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G.H. Lee, S. Lynen, L. Meier, M. Pollefeys, A. Renzaglia, Roland Siegwart, J.C. Stumpf, P. Tanskanen, C. Troiani, S. Weiss

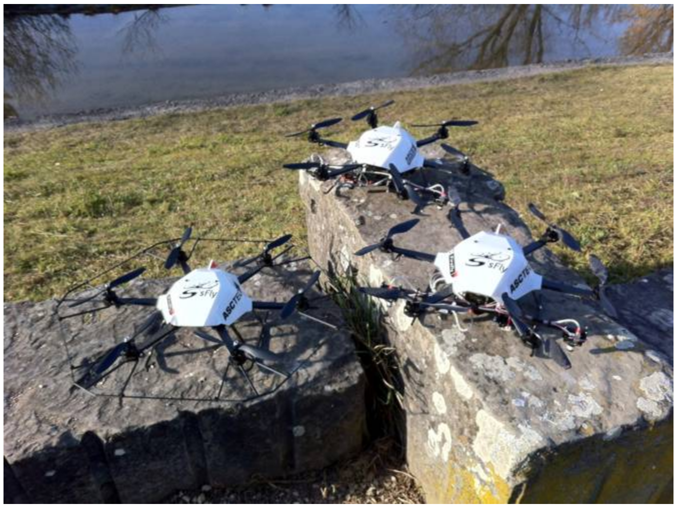

Fig. 1. The three SFLY hexacopters designed for inertial-visual navigation in GPS-denied environments.

I. Introduction

A. Motivation

Autonomous navigation of micro helicopters (where "micro" means up to the size of a few decimeters and less tan 2kg) has progressed significantly in the last decade thanks to the miniaturization of exteroceptive sensors (e.g., laser rangefinders and digital cameras), and to the recent advances in micro-electromechanical systems, power supply, and vehicle design.

Micro helicopters—and notably multi-rotor helicopters—have several advantages compared to fixed-wing micro aerial vehicles: they are able to take off and land vertically, hover on a spot, and even dock to a surface. This capability allows them easily to work in small indoor environments, pass through windows, traverse narrow corridors, and even grasp small objects [1].

A key problem in aerial-vehicle navigation is the stabilization and control in six degrees of freedom (DOF), that is, attitude and position control. Today's systems handle well the attitude control. However, without a position control, they are prone to drift over time. In GPS-denied environments, this can be solved using offboard sensors (such as motion-capture systems or total stations) or onboard sensors (such as cameras and laser rangefinders). The use of offboard sensors allows research to focus on control issues without dealing with the challenges of onboard perception. Today's popular MAV testbeds are given by Vicon or OptiTrack motion-capture systems, which consist of multiple infrared static cameras tracking the position of a few highly-reflective markers attached to the vehicle with millimeter accuracy and at a very high frame rate (several hundred Hz). These systems are very appropriate for testing and evaluation purposes [2] —such as multi-robot control strategies or fast maneuvers—and serve as a ground-truth reference for other localization approaches. Using this infrastructure, several groups have demonstrated aggressive maneuvers and impressive acrobatics [3], [4].

In the works mentioned above, the MAVs are actually "blind." To navigate, they rely on the highly-precise position measurement provided by the external motion-tracking system. As a matter of fact, what is really autonomous is not the single MAV itself but the system comprising the MAVs plus the external cameras. Furthermore, these systems are limited to small, confined spaces and require manual installation and calibration of the cameras, making it impossible to navigate autonomously in unknown, yet-unexplored environments. Therefore, for a MAV to be fully autonomous, sensors should be installed onboard.

B. Paper Overview

This paper describes the technical challenges and results of a three-year European project—named SFLY (Swarm of Micro Flying Robots[1]—devoted to the implementation of a system of multiple micro flying robots capable of autonomous navigation, 3D mapping, and optimal coverage in GPS-denied environments. The SFLY MAVs do not rely on remote control, radio beacons, or motion-capture systems but can fly all by themselves using only an onboard camera and an IMU. This paper describes the major contributions of the SFLY, from hardware design and embedded programming to vision-based navigation and mapping. The first contribution is the development of a new hexacopter equipped with enough processing power for onboard computer vision. The second

The authors are with ETH Zurich, University of Zurich, INRIA Rhone-Alpes, CERTH, Technological Educational Institute of Crete, Ascending Technologies.

[1] www.sfly.org

contribution is the development of a local-navigation module based on monocular SLAM that runs in real time onboard the MAV. The output of the monocular SLAM is fused with inertial measurements and is used to stabilize and control the MAV locally without any link to a ground station. The third contribution is an offline dense-mapping process that merges the individual maps of each MAV into a single, global map that serves as input to the global navigation module. Finally, the fourth contribution is a cognitive, adaptive optimization algorithm to compute the positions of the MAVs, which allows the optimal surveillance coverage of the explored area.

The structure of the paper is the following. After reviewing the related work (Section II), the paper starts with the description of the design concept, the electronic architecture, and the mechanical concept of the aerial platform (Section III). Then, it describes the inertial-aided vision-controlled navigation, 3D mapping, and optimal-coverage approaches (Section IV, V, and VI, respectively). Finally, it presents the experimental results with three MAVs (Section VII).

II. Related Work

A. System Design

Extensive work has been carried out on quadrotor systems. The function principle of quadrotors can be found in [5], [6]. A review of the state of the art on modeling, perception, and control of quadrotors can be found in [7]. The pitch angle of the propellers is typically fixed; an evaluation of variable-pitch propellers is presented [8]. The platform described in this paper—the Asctec FireFly—is the improvement of the previous and popular model known as AscTec Pelican. While other groups often run the computation offboard—by transmitting image data to a powerful ground-station computer—the SFLY platform runs most computer-vision algorithms fully onboard. This demands high onboard-computation capabilities. In the first SFLY vehicle [9], a 1.6 GHz Intel Atom computer was used; however, in the latest platform, this was replaced with a Core-2-Duo onboard computer able to process all flight critical data on-board.

B. Autonomus Navigation

Autonomous navigation based on onboard 2D laser-rangefinders has been largely explored for ground mobile robots. Similar strategies have been extended to MAVs to cope with their inability to "see" outside the scan plane. This is usually done by varying the height and/or the pitch and roll of the helicopter, and by incorporating readings from air-pressure and gyroscopic sensors [10]. Although laser scanners are very reliable and robust, they are still too heavy and consume too much power for lightweight MAVs. Therefore, vision sensors are very appealing; however, they require external illumination and a certain computing power to extract meaningful information for navigation.

Most of the research on vision-based control of MAVs has focused on optical flow [11]. However, since optical flow can only measure the relative velocity of features, the position estimate of the MAV will inevitably drift over time. In order to avoid drift over long time, the system should be able to relocalize whenever it comes back to a previously-visited location. One possibility is offered by SLAM (Simultaneous Localization and Mapping) approaches.

Preliminary experiments for MAV localization using a visual EKF-based SLAM technique were described in [12]. However, the first use of visual SLAM to enable autonomous basic maneuvers—such as take-off and landing, point-to-point navigation, and drift-free hovering on the spot—was done right within the framework of the SFLY project [13], [14]. Due to the use of a single camera, the absolute scale was initially determined manually or using a known-size object [15]. Later, the system was extended [16] to incorporate data from an Inertial Measurement Unit (IMU) and, thus, estimate the absolute scale automatically while self-calibrating all the sensors (this approach will be outlined in Section IV).

C. Optimal Coverage

Optimal coverage is the problem of computing the poses of a team of robots, which guarantee the optimal visibility of an area under the constraints that:

- The part of terrain monitored by each robot is maximized;
- For every point in the terrain, the closest robot is as close as possible to that point.

The optimal visibility problem is also related to the Art-Gallery Problem, where the goal is to find the optimum number of guards in a non-convex environment such that each point of the environment is visible by at least one guard [17].

Most approaches for multi-robot surveillance coverage concentrate on the second objective and tackle 2D surfaces [18]. A method for non-planar surfaces embedded in 3D was presented in [19], while a study for multiple flying robots equipped with a downlooking camera observing a planar 2D environment was proposed in [20]. Conversely, the approach described in this paper is based on a new stochastic optimization method, called Cognitive-based Adaptive Optimization (CAO). This method addresses 3D environments and tackles the two aforementioned objectives simultaneously.

III. Micro Helicopter Platform

A. Design Concept

One goal of the SFLY project was to have a vehicle as small, lightweight (less than 1.5kg), and safe as possible, while being capable of carrying and powering an onboard computer and cameras. Since the SFLY helicopter was envisaged to operate in urban environments, the impact energy had to be reduced to a minimum. To limit the risk of injuries, studies were made to evaluate the effects of having more than four (but smaller and safer) rotors on efficiency achievable dynamics and redundancy. These studies are presented in detail in [21]. Summarized, the smaller the numbers of rotors, the better the efficiency of the vehicle. On the other hand, the achievable dynamics and, therefore, the maneuverability of the vehicle can be enhanced by a larger number of propellers and a smaller ratio between rotor surface and total weight. However, for safe operation, the most important aspect is redundancy against at least a single-rotor failure. In [21] it was shown that the minimum number of rotors with redundancy against a single failure

TABLE I

THE TABLE SHOWS THE THEORETICAL MAXIMUM THRUST IN REDUNDANCY SITUATIONS FOR DIFFERENT CONFIGURATIONS.

System configuration	Thrust in failure situation
Triangle hex	50 %
Hexagon hex	66 %
V-Shape octo	62 %
Octagon octo	70-73 %

could be reduced to six due to a new redundancy concept. To do so, different shapes of redundant multi-rotor vehicles were analyzed and the maximum thrust in a redundancy situation was calculated. The results are shown in Table I (neglecting the additional margin needed to control the other axes). The hexagon-shaped six-rotor design was chosen as the best trade-off. This can be built with propellers as small as the known safe propellers of the AscTec Hummingbird [9]. Additionally, it can carry the demanded payload and is redundant against single-rotor failures, thus, enabling safe operations in urban areas. Compared to an octocopter design, the thrust in a redundancy situation is smaller but the overall efficiency is higher due to the use of six rotors instead of eight.

B. Electronic Architecture

Except for the two additional motors, the electronic components and the software architecture are about the same as the Asctec Pelican described in [9]. A distribution of the Flight-Control-Unit's (FCU) main task between two microprocessors is illustrated in Fig. 2. The so-called Low Level Processor (LLP) handles all hardware interfaces; it is connected to the sensors and computes the attitude-data-fusion and flight-control algorithms at an update rate of 1 kHz. The High Level Processor (HLP) is open for customized or experimental code. In the SFLY project, the HLP is used for state estimation and control. It has proven to be helpful to have the LLP as a safety backup while performing experiments in flight.

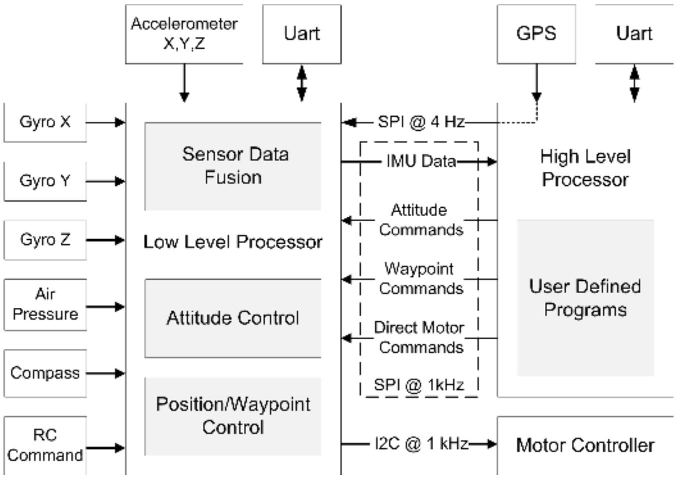

Fig. 2. Electronic architecture: All sensors, except the GPS, are connected to the LLP which communicates via I2C with the motor controllers and via SPI with the HLP.

C. Onboard Computer

To integrate all computational intense parts onboard the vehicle, the initial Atom computer board of the Pelican platform was not sufficient. Therefore, the ongoing development of a new motherboard supporting the COM express standard was pushed forward to support the integration of a Dual Core Atom, a Core 2 Duo, or a Core i7 CPU unit. These computer boards provide enough computational power to run all onboard software. Furthermore, additional interfaces like Firewire and hardware serial ports are supported. Especially, the hardware serial ports are another step towards precise and fast state estimation on the onboard computer as the latency is reduced to a minimum.

Fig. 3. On-board computer AscTec Mastermind featuring a Core 2 Duo CPU

D. Mechanical Concept and Vibration Decoupling

One important requirement, raised from test flights of the previous vehicles is a vibration decoupling. Just decoupling the IMU has proven not to be sufficient. Instead, payloads such as cameras should be decoupled as well, and ideally fixed to the IMU. Vibration damping is necessary to improve state estimation for position control as well as image quality. The damping system has to be designed so that there is a rigid connection between cameras and IMU in order to avoid any dynamic misalignment. These requirements led us to a completely new concept. A so-called ”frame-in-frame” concept was built: the outer frame holds the motors, the landing gear, the canopy, and the propeller protection, while the inner frame carries the IMU, the battery, and the payload. As shown in Fig. 4, both frames are connected using special silicon dampers, distributed in a pattern to intentionally influence the dynamics between both frames. This is necessary because the frame-in-frame concept leads to additional dynamics between both parts. The eigenmodes of this new dynamic system had to be adjusted so that no resonance oscillations between both frames occurred for a variety of payload configurations. Flight tests show an improvement of image and state-estimation quality and all resonance oscillations are eliminated. Due to this new damping concept, the whole mechanical structure had to be redesigned.

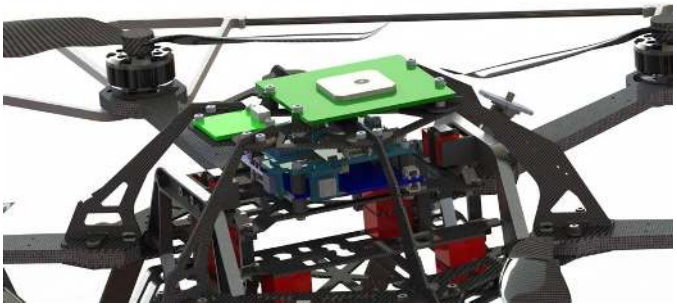

Fig. 4. CAD model illustrating the vibration damping between the two parts of the frame: The motors and the landing gear are connected to the outer frame and the inner frame to the IMU, battery and payload. The silicon dampers are highlighted red

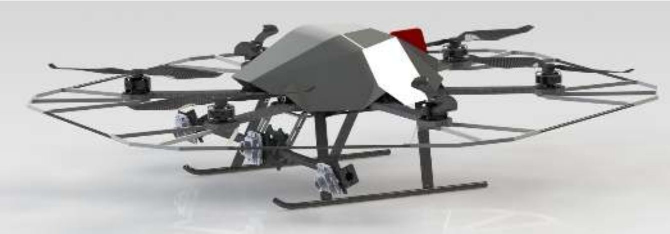

Fig. 5. Complete CAD model including three cameras on the SFLY hexacopter.

To reduce the overall height and to concentrate the mass closer to the center of gravity, the battery was moved to the center of the frame. Furthermore, a landing gear was added to protect the payload which is connected to the dampened frame. A roll-over bar protecting the electronic components and supporting the cover was added as well.

Besides these additional features, another requirement was to enable fast component changes in case of a crash or modification during integration and testing. To put all these requirements and features together, a new combination of carbon fiber, carbon fiber sandwich, and aluminum was chosen. Details of this concept can also be seen in Fig. 4 and a complete CAD model including a camera mount is shown in Fig. 5. [2] Table II summarizes the main technical data.

E. Flight-Time Estimation and Payloads

Based on test-bench data of the consequently improved motors and propellers, as well as a final empty weight of 640g, the flight time can be calculated for different payloads and batteries (Fig. 6). The weight of the different batteries is taken into account and the plots are limited to the maximum take-off weight. The flight time is calculated for 85% of the battery capacity because lithium-polymer batteries must not be completely discharged. For the SFLY requirements, the 4900 mAh battery was selected, resulting in approximately 16-minutes flight time at 400 g payload (neglecting the onboard-computers power consumption).

TABLE II
MAIN TECHNICAL DATA

Empty weight	0.64kg
$I_{xx} \approx I_{yy}$	$0.013 kg \cdot m^2$
I_{zz}	$0.021 kg \cdot m^2$
Total thrust(@10.5V)	$24N$
max take off weight	1.45kg
max. Payload	450g
max. Flight Time	up to $30min$

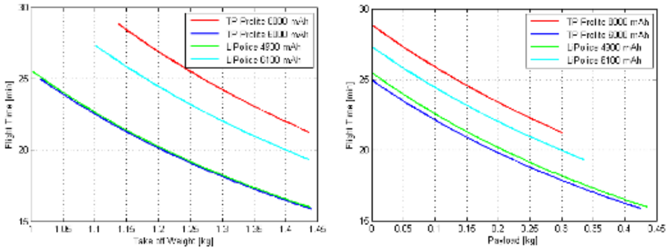

Fig. 6. Calculated flight time vs. payload and take-off weight. The figure shows the estimated flight time for a given payload with different batteries.

IV. Inertial-Aided Visual Navigation

The navigation of the MAVs is handled by two different modules that are named Local-Navigation and Global-Navigation. The Local-Navigation module is responsible for flight stabilization, state estimation (including absolute-scale estimation), and way-point–based navigation of each MAV. The Local-Navigation module runs onboard each platform and estimates the pose of each MAV with respect to its starting position. The relative positions of the MAVs at start are unknown. The task of the Global-Navigation module (running offboard the MAVs, on a ground-station computer) is to express the poses of all MAVs in a common, global coordinate frame and, possibly, to reduce both motion and map drifts. This is done by identifying both loop closures by the same MAV and path intersections between multiple MAVs (Fig. 7).

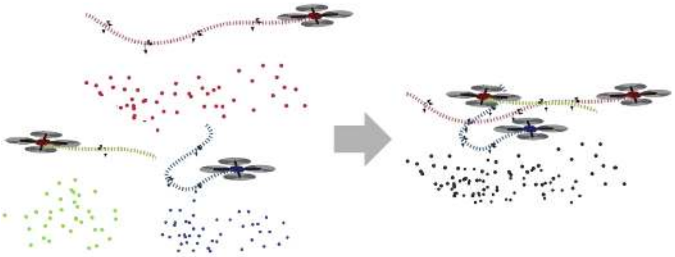

Fig. 7. (Left) The Local-Navigation module (running onboard) estimates the pose of each MAV independently for each platform. (Right) The Global-Navigation module (offboard) recognizes path intersections and uses them to express the MAVs poses in the same, global coordinate frame and to reduce drift.

[2] Note that only one camera (down looking)is used for navigation, while the other two—in stereo configuration—are used for obstacle avoidance (not described here) and dense matching (Section V).

A. Local Navigation

5DOF[3] single-camera–based visual odometry has made significant progress in the recent years (a tutorial on monocular and stereo VO can be found in [22] and [23]). Filter-based and keyframe-based off-the-shelf algorithms are publicly available. Because of robustness, real-time performance, and position accuracy, the keyframe-based solution proposed in [24] was selected and tailored to the general needs of our computationally-limited platform. Our framework uses the ROS[4] middleware and runs on a standard Ubuntu operating system, facilitating the development of new algorithms. The current implementation uses only 60% of one core of the Core 2 Duo processor at 30 Hz, leaving enough resources for future higher-level tasks. As a reference, the same implementation on an Atom 1.6 GHz single-core computer runs at 20 Hz using 100% of the processing power.

The 5DOF pose of the MAV camera output by the visual-odometry algorithm was fused with the inertial measurements of an IMU using an Extended Kalman Filter (EKF). More details are given in [25]. An EKF framework consists of a prediction and an update step. The computational load required by these two steps is distributed among the different units of the MAV as described in [26]. The state of the filter is composed of the position p_w^i, the attitude quaternion q_w^i, and the velocity v_w^i of the IMU in the world frame. The gyroscope and accelerometer biases b_ω and b_a as well as the missing scale factor λ are also included in the state vector. For completeness, the extrinsic calibration parameters describing the relative rotation q_i^s and position p_i^s between the IMU and the camera frames were also added. Notice that the calibration parameters could be omitted from the state vector and be set to a pre-measured constant. This yields a 24-element state vector X:

$$X = \{p_w^{i^T}\ v_w^{i^T}\ q_w^{i^T}\ b_\omega^T\ b_a^T\ \lambda\ p_i^s\ q_i^s\}. \tag{1}$$

Fig. 8 depicts the setup with the IMU and camera coordinate frames and the state variables introduced above.

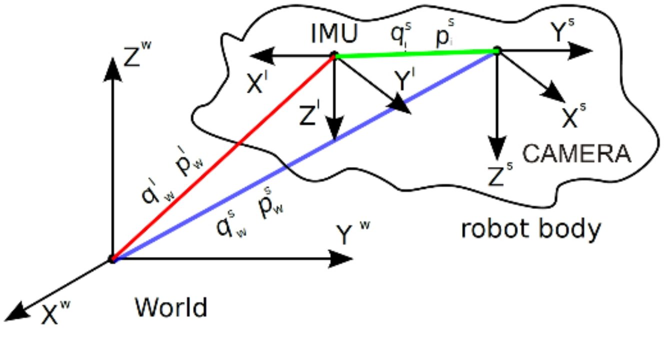

Fig. 8. Setup depicting the robot body with its sensors with respect to a world reference frame. The system state vector is $X = \{p_w^i\ v_w^i\ q_w^i\ b_\omega\ b_a\ \lambda\ p_i^s\ q_i^s\}$ whereas p_w^s and q_w^s denote the robot's sensor measurements in (a possibly scaled) position and attitude respectively in a world frame.

The equations of the EKF prediction step for the considered IMU-camera fusion are given in [25]. The equations of the update step are derived by computing the transformation from the world reference frame to the camera frame as follows. For the position z_p, we can write:

$$z_p = p_w^s = (p_w^i + C^T_{(q_w^i)} p_i^s)\lambda + n_p \tag{2}$$

where $C_{(q_w^i)} \in SO(3)$ is the rotation matrix associated to the IMU attitude quaternion q_w^i in the world frame, z_p denotes the observed position (the output of the visual odometry), λ is the scale factor, and n_p the measurement noise. For the rotation measurement z_q, we apply the notion of error quaternion. Since the visual-odometry algorithm yields the rotation q_w^s from the world frame to the camera frame, we can write:

$$z_q = q_w^s = q_i^s \otimes q_w^i \tag{3}$$

A non-linear observability analysis [27] reveals that all state variables are observable, including the inter-sensor calibration parameters p_i^s and q_i^s. Note that the visual pose estimates are prone to drift in position, attitude, and scale with respect to the world-fixed reference frame. Since these quantities are observable (and notably roll, pitch, and scale), gravity-aligned metric navigation becomes possible even in long-term missions. This is true as long as the robot excites the IMU accelerometer and gyroscopes sufficiently as discussed in [28]. Note that the estimated attitude and position of the MAV in the world frame is subject to drift over time. However, since the gravity vector measured by the IMU is always vertically aligned during hovering, this prevents the MAV from crashing even during long-term operations.

B. Global Navigation

The task of the Global-Navigation module (running on the ground station) is to express the poses of all MAVs in a common, global coordinate frame and, possibly, to reduce both motion and map drifts. This is done by matching the current camera image to a 3D environment map. The 3D map consists of landmarks (3D points and corresponding descriptors in each image) and the corresponding camera poses. The 3D map is computed offline as described in section V and combines the maps of the individual MAVs into a single merged map. Map merging works by identifying both loop closures by the same MAV and path intersections between multiple MAVs (Fig. 7). To reduce the computational load of the onboard computer, the Global-Navigation module runs on a ground station that constantly receives the images of the MAVs via WiFi and sends back the updated global poses.

Matching the current camera view to the 3D map is done by vocabulary-tree–based image search as described in [29], [30]. For every frame, SURF features [31] are extracted and then quantized into visual words using a vocabulary tree that was pre-trained on a general image dataset. The image IDs and the corresponding visual words are stored in a database that is organized as an inverted file for efficient data access. Additional meta data (pose estimates from the Local-Navigation module and IMU data) are stored with each image in the database. Whenever a new image is processed, it is ranked

[3] We refer to 5DOF instead of 6DOF because the absolute scale is not observable with a single camera. However, the scale factor can be estimated by adding an IMU, as explained in this section.

[4] www.ros.org

with all images in the database according to the similarity of the visual words. Geometric verification is performed on the top-N most similar frames using $P3P$-based RANSAC [32]. A match is accepted if the inlier count exceeds a certain threshold. The initial pose from RANSAC gets refined using non-linear optimization and is sent back as global pose update. This approach allows for efficient localization and also scales to large maps.

V. 3D Mapping

For the 3D mapping of the environment, an offboard ground station takes images from all MAVs and fuses them into a detailed map. The mapper is based on the *g2o* framework [33]; it uses a pose-graph optimizer for pre-alignment of the data, and then, runs a bundle adjustment to get optimal results.

The maximum-likelihood estimates of the poses are computed by minimizing the Euclidean distances between the transformations in a pose graph. The non-linear optimization is done by sparse Cholesky decomposition using the g2o framework. To improve the accuracy of the map, a bundle adjustment is run. The bundle adjustment optimizes the poses and the 3D positions of all features at the same time by minimizing the image reprojection error. The corresponding graph of this problem consists of the MAV poses and the 3D feature points as nodes. They are connected by edges that represent the projection of the 3D feature point to images where the feature was detected. During the loop-detection phase, for every new frame all image projections of the inlier features are added to the bundle-adjustment graph.

A dense map is built using the poses of the MAV computed from the bundle adjustment process, and the corresponding stereo images. For each pose and corresponding stereo frame, a 3D point cloud in global coordinates is computed via stereo triangulation, and used to update a 3D occupancy map. After all the data has been processed, a terrain map is extracted from the 3D occupancy map by thresholding the occupancy value in each cell in the occupancy map. The terrain map is triangulated to create a dense mesh. Furthermore, a dense textured map is created by projecting all triangular faces in the mesh onto the images from which the faces are entirely visible, and texturing each face with the image that has the smallest incident angle relative to the face normal. This image selection heuristic helps to minimize perspective distortion. A textured visualization of a 3D map is shown in Fig. 19. More details can also be found in [34].

VI. Optimal Surveillance Coverage

The problem of deploying a team of flying robots to perform surveillance coverage missions over an unknown terrain of complex and non-convex morphology was tackled using a novel Cognitive-based Adaptive Optimization (CAO) algorithm. The CAO algorithm was originally developed and analyzed for the optimization of functions for which an explicit form is unknown but their measurements are available, as well as for the adaptive fine-tuning of large-scale nonlinear-control systems [35]. Within SFLY, CAO was implemented for surveillance tasks in unknown 3D terrains of complex and non-convex morphology with obstacles using only onboard monocular vision. CAO possesses several advantages compared to the previous works described in Section II-C: it is computationally simple to implement, scalable, and can easily embed any kind of physical constraints and limitations (e.g., obstacle avoidance, nonlinear sensor-noise models, etc). CAO does not create an approximation or estimation of the obstacles' location and geometry; conversely, it produces an online local approximation of the cost function to be optimized. A detailed description of the CAO algorithm and its functionality for the case of a team of aerial robots can be found in [36], [37].

In the context of the SFLY project, CAO algorithm tackles two objectives simultaneously to assure that the robot team will perform optimal surveillance coverage:

- Maximize the part of terrain monitored by each robot;
- For every point in the terrain, the closest robot has to be as close as possible to that point

If only the first objective were considered, the robots would fly as high as their visibility threshold allows for (which is defined as the maximum distance the robots sensor can measure). Therefore, the second objective ensures that, among all possible configurations that maximize the visible area V, the robot team converges to the one that keeps as small as possible the average distance between each robot and the part of the terrain that that particular robot is responsible for. The second objective is also necessary for two practical reasons: first, the closer is the robot to a point in the terrain the better is, in general, its sensing ability to monitor this point; second, in many multi-robot coverage applications it is necessary to intervene as fast as possible in any of the points of the terrain with at least one robot.

The two aforementioned objectives are combined in an objective function that the robot team has to minimize [36], that is:

$$J(P) = \int_{q \in V} \min_{i \in \{1,\ldots,M\}} \|x^{(i)} - q\|^2 dq + K \int_{q \in T-V} dq, \qquad (4)$$

where M is the number of robots that are deployed to monitor a terrain T, $x^{(i)}$ is the position of the i-th robot, $P = \{x^{(i)}\}_{i=1}^{M}$ denotes the configuration of the robot team, q is a point in the terrain T, V consists of all points $q \in T$ that are visible from the robots, and K is a user-defined positive constant.

The first term in Eq. (4) addresses the second objective. The second term addresses the first objective and relates to the invisible area of the terrain (i.e., $\int_{q \in T-V}$, which is the total part of the terrain that is not visible to any of the robots). The positive constant K serves as a weight to give more or less priority to one or the other objective. A detailed analysis of the effect of K is presented in [36].

The implementation of CAO within the SFLY framework ensures that the physical constraints are also met throughout the entire multi-robot coverage application. Such physical constraints include—but are not limited to—the following ones:

- The robots remain within the terrain's limits;
- The robots satisfy a maximum-height requirement while not hitting the terrain;

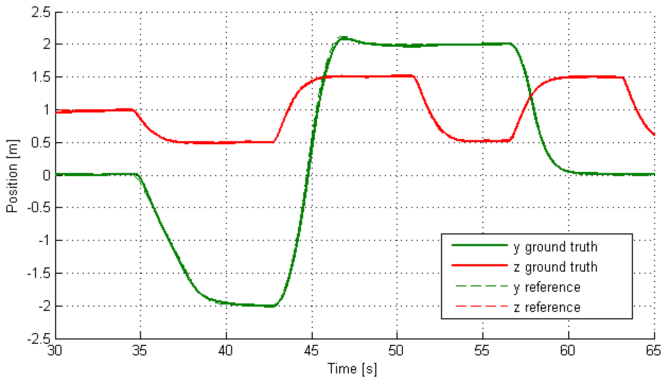

Fig. 9. The plot shows the commanded reference trajectory and the measured ground truth.

- The robots do not come closer to each other than a minimum allowable safety distance.

The above constraints can be easily formulated and incorporated in the optimization problem. CAO uses function approximators for the estimation of the objective function at each time instant; therefore, a crucial factor for the successful implementation is the choice of the regressor vector, as described in [36]. Once the regressor vector has been set and the values of the cost function are available for measurement, it is possible to find at each time step the vector of parameter estimates and, thus, the approximation of the cost function.

VII. Experimental Results

A. Flying Platform

The achievable dynamics and maneuverability are demonstrated by the accurate trajectory following and position control shown in Fig. 9.

To evaluate the redundancy capabilities, a switch disabling one motor was implemented to be operated by the radio controller. There was no measurable deviation in the roll and pitch axes, but the maximum thrust is obviously limited during these redundancy situations. Figure 10 shows the motor commands input to the four propellers during such a redundancy test. The motor commands are in the range [-100,200]. As observed, at about 14s, one motor is deactivated (the yellow plot drops to 0) and one motor command starts compensating for the yaw moment by slowly oscillating around zero (red plot). The other four motors are set feed forward to a higher thrust to compensate for the loss caused by the other two motors. The middle plot shows the pilots stick inputs. This plot looks absolutely normal for a manual flight like the bottom one, showing the attitude measurement.

B. Vision Based Navigation

Fig. 11 and Fig. 12 show the evolution of the position and attitude of one MAV estimated by the EKF framework described in Section IV-A. The position plot (Fig. 11) shows that the visual scale has been estimated correctly by the filter; as observed, the position and attitude drifts of the vision system are very low. For a rapidly-drifting vision system, one would observe an increased difference between GPS data and filter estimates. The attitude plot shows that, although each MAV was initially aligned manually to the correct yaw angle, the filter converges in less than 20 seconds to the correct attitude values. Notice that GPS measurements were used as additional input in the EKF only to allow ground-truth comparison.

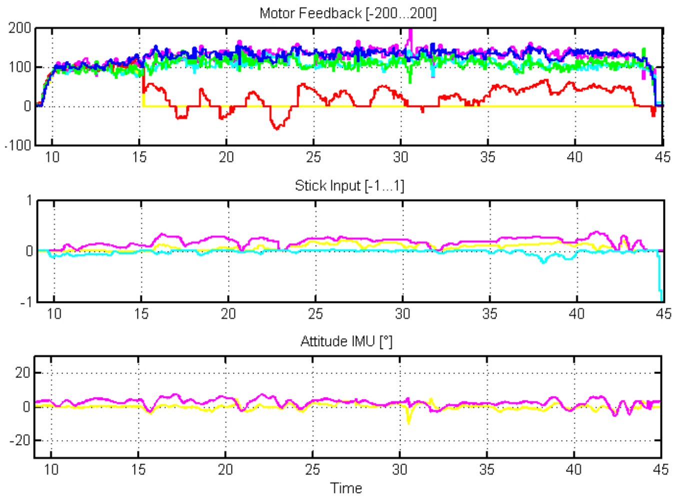

Fig. 10. The top plot shows the motor commands in a range [-100,200]. At about 14s, the yellow motor is disabled so that the redundancy controller can be activated. As observed, the red motor command slowly oscillates around zero to compensate the yaw moment. The middle and lower plot show that there is nearly no influence of the failing motor to the pilots commands or measured attitude.

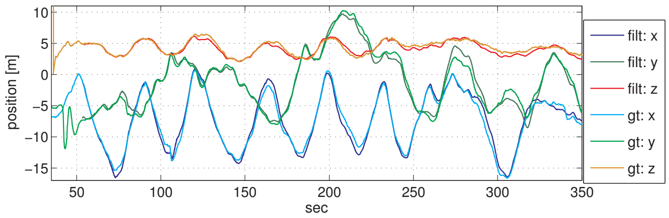

Fig. 11. Comparison between EKF-based position estimate (filt: x,y,z)and raw GPS measurements (gt: x,y,z) during a 5-minute interval of time. The plot suggests that the absolute scale is estimated correctly.

A 350m trajectory estimated using this framework, resulting in an overall position drift of only 1.5m, is shown in Fig. 13.

The presented framework was tested under a variety of challenging conditions, exhibiting robustness in the presence of wind gusts, strong light conditions causing saturated images, and large scale changes in flight altitude. More details are given in [38].

C. 3D Mapping and Optimal Coverage

The platforms and the algorithms described in the previous sections were used to implement an autonomous-navigation scenario that was publicly demonstrated at the firefighters' training area of the city of Zurich (Fig. 14). As described in Section IV, a visual odometry algorithm ran onboard each MAV and served for local stabilization as well as for trajectory estimation. At the same time, each MAV built a sparse 3D map that was incrementally transmitted—together with images

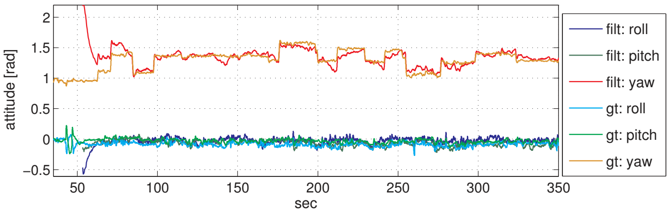

Fig. 12. Comparison between EKF-based attitude estimate (filt: *roll*, *pitch*, *yaw*)and GPS-IMU based estimates from the AscTec internal state estimator (gt: *roll*, *pitch*, *yaw*) during a 5-minute interval of time.

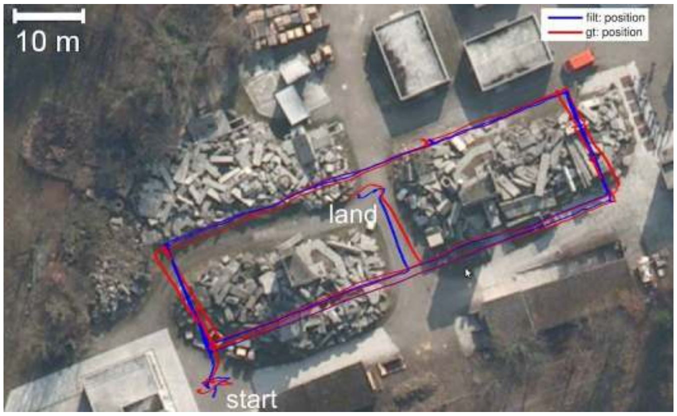

Fig. 13. After a short initialization phase at the start, vision-based navigation (blue) was switched on for successful completion of a more than 350 m-long trajectory until battery limitations necessitated landing. The comparison of the estimated trajectory with the GPS ground truth (red) indicates a very low position and yaw drift of the real-time onboard visual odometry.

and pose estimates—over a WiFi network to a ground station computer. The ground station—a quadcore Lenovo W520 laptop—was in charge of combining all the received data to compute real-time global position estimates of the three MAVs as well as a dense 3D map.

Fig. 15 shows the pose graphs built by the three MAVs during a flight over the area. These graphs are generated after visual odometry. Drift is visible especially in the blue trajectory. There, start and end points are marked with red arrows. Start and end points should overlap in this case, but do not due to drift. Loop detection, however, recognized the loop closure.

Finally, the three individual submaps are merged into a single global map: first, loop closures are detected between the submaps; then, global bundle adjustment is run over the whole map. Fig. 16 shows the pose graph of the final map. The black lines between the cameras of different submaps show the detected loop closures. The global bundle adjustment is able to remove the drift in the individual submaps; thus, the resulting global map is drift-free and in the correct absolute (metric) scale.

A 3D occupancy map was built as described in Section V. Out of the 3D occupancy grid, a height map was generated (Fig. 17) and fed to the CAO algorithm to compute the optimal-coverage poses. The produced map covers a 42m×32m area with maximum height of 8.3m. The final poses for the optimal surveillance coverage of the area by the three MAVs are shown in Fig. 18.

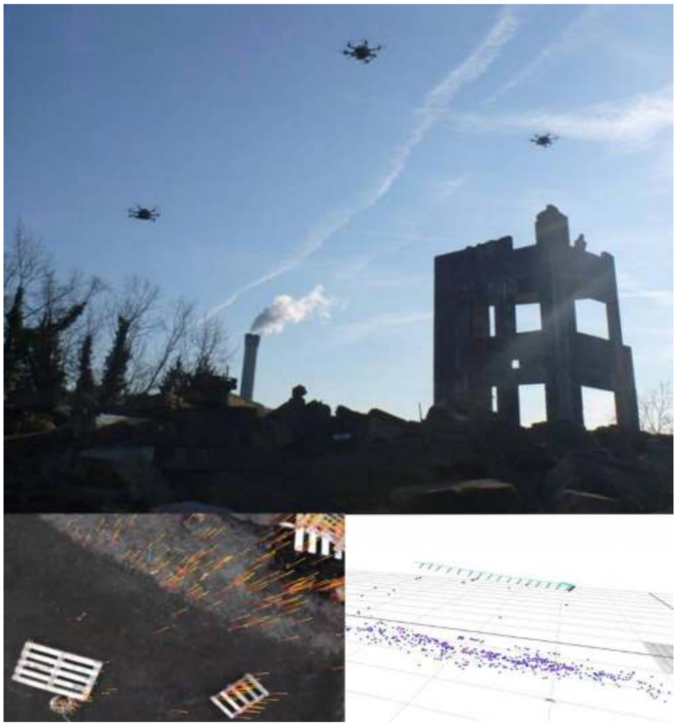

Fig. 14. SFLY helicopters during a demonstration of autonomous exploration at the firefighters training area in Zurich. (Bottom left) Feature tracks. (Bottom right) Online-built 3D sparse map used for local navigation.

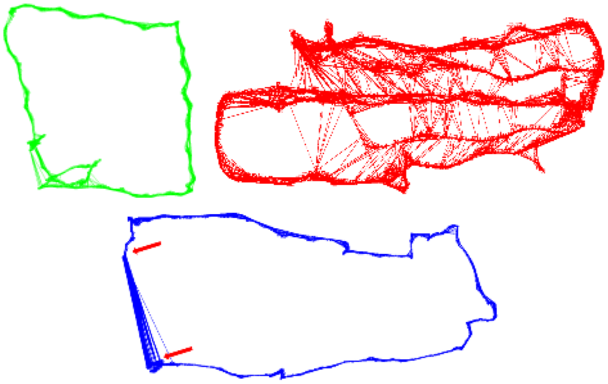

Fig. 15. The pose graphs of three flight trajectories that were used for 3D mapping. The camera poses are plotted after visual odometry and windowed bundle adjustment. The connecting lines between the cameras show loop closures. As no global optimization is run, pose drift is visible. In the blue trajectory, start and end points are marked with red arrows. Start and end points should overlap in this case, but do not due to drift. Loop detection, however, recognized the loop closure and pose graph optimization will remove the drift 16.

Fig. 19 shows a textured visualization of the 3D environment map of the firefigher area created from 3 MAV's.

VIII. Videos, Code, Datasets publicly available

Please notice that this paper is accompanied by multimedia material. Videos showing the SFLY MAVs' capabilities can be watched on the SFLY YouTube channel:
https://www.youtube.com/sflyteam
Some highlights are:

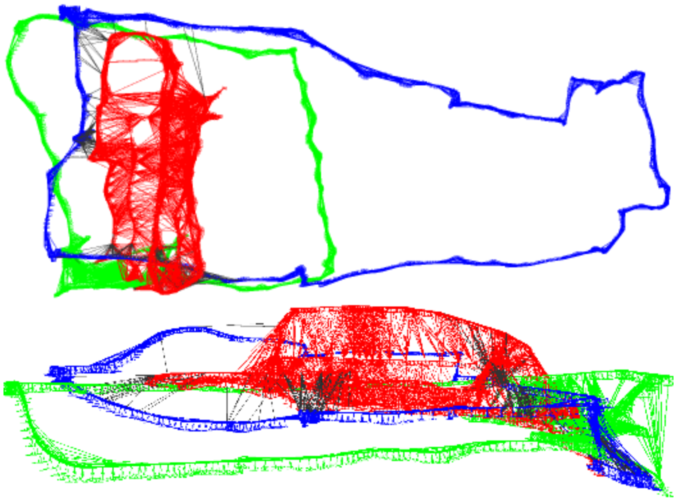

Fig. 16. Top and front view of the pose graphs of the three flight trajectories in Fig. 15 after map merging and global bundle adjustment. The black lines show the loop closures between the three sub maps.

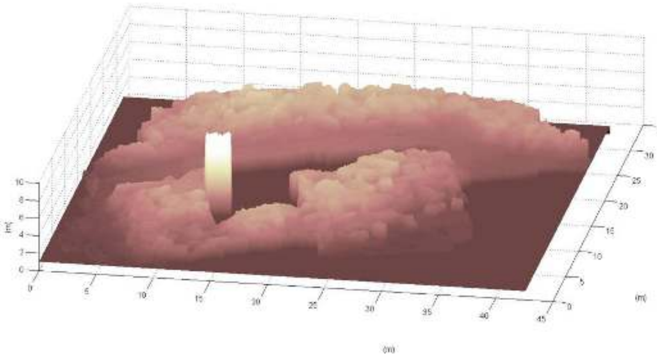

Fig. 17. Height map of the Zurich's firefighters training area.

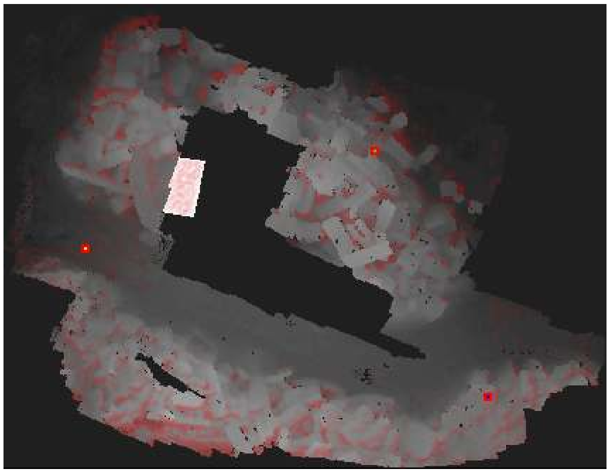

Fig. 18. Final configuration of a robot team performing surveillance coverage. The red squares represent the final positions of the MAVs, while the red areas represent the invisible part of the map.

- http://youtu.be/_-p08o_oTO4
- http://youtu.be/vHpw8zc7-JQ

Also, note that the code for visual SLAM, vision-based control, sensor fusion, and self-calibration is publicly available on the ROS webpage:

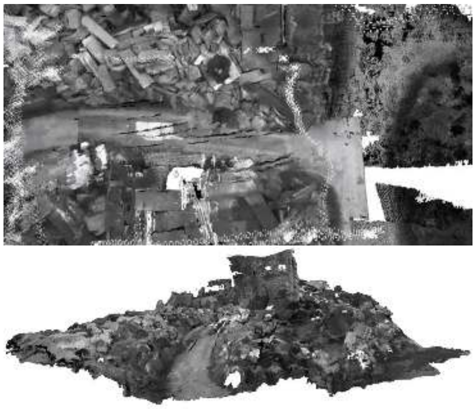

Fig. 19. Textured visualization of the 3D map of the firefighter area. (top and side view).

- Modified version of the well known monocular SLAM framework PTAM
  http://www.ros.org/wiki/ethzasl_ptam
- Sensor fusion:
  http://www.ros.org/wiki/ethzasl_sensor_fusion
- Communication, state estimation and position control of AscTec helicopters:
  http://www.ros.org/wiki/asctec_mav_framework

A demo version of the optimal coverage approach is also publicly available at:
http://www.convcao.com/?page_id=492.

Finally, datasets (images, IMU) with ground truth are available from the SFLY webpage at:
http://www.sfly.org/mav-datasets.

IX. Conclusions

This paper described a framework that allows small-size helicopters to navigate all by themselves using only a single onboard camera and an IMU, without the aid of GPS or active range finders. This framework allows unprecedented MAV navigation autonomy, with flights of more than 350m length, in previously unexplored environments.

This paper shared the experience earned during the three-year European project SFLY about visual-inertial real-time onboard MAV navigation, multi-robot 3D mapping, and optimal surveillance coverage of unknown 3D terrains. Particular focus was devoted to the technical challenges that have been faced and the results achieved, with a detailed insight of how all the modules work and how they have been integrated into the final system. Code, datasets, and videos were made publicly available to the Robotics community.

This paper highlighted four major contributions of SFLY. The first one is the development of a new a six-rotor–based platform robust to single-rotor failures, equipped with enough processing power for onboard computer vision. The second contribution is the development of a local-navigation module based on monocular SLAM that runs in real time onboard the MAV. The output of the monocular SLAM is fused with inertial measurements and is used to stabilize and control the

MAV locally without any link to a ground station. The third contribution is an offline dense-mapping process that merges the individual maps of each MAV into a single, global map that serves as input to the global navigation module. Finally, the fourth contribution is a cognitive, adaptive optimization algorithm to compute the positions of the MAVs, which allows the optimal surveillance coverage of the explored area.

To the best of our knowledge, this paper describes the first, working vision-only–based system of multiple MAVs in real-world scenarios able to autonomously navigate while collaboratively building a rich 3D map of the environment and performing optimal surveillance coverage. It is believed that the presented system constitutes a milestone for vision-based MAV navigation in large, unknown, and GPS-denied environments, providing a reliable basis for further research towards complete missions of search-and-rescue or inspection scenarios with multiple MAVs.

References

[1] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2010.

[2] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro uav testbed," *IEEE Robotics and Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[3] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *IEEE International Conference on Robotics and Automation*, May 2010.

[4] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2520–2525.

[5] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and control of an indoor micro quadrotor," in *IEEE International Conference on Robotics and Automation*, 2004.

[6] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles-modeling, estimation, and control of quadrotor," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[7] N. Michael, D. Scaramuzza, and V. Kumar, "Special issue on micro-uav perception and control," *Autonomous Robots, special issue, editorial*, vol. 23, no. 1–2, 2012.

[8] M. Cutler, N. Ure, B. Michini, and J. P. How, "Comparison of fixed and variable pitch actuators for agile quadrotors," in *AIAA Guidance, Navigation, and Control Conference (GNC)*, Portland, OR, August 2011. [Online]. Available: http://acl.mit.edu/papers/GNC11_Cutler_uber.pdf

[9] M. C. Achtelik, J. Stumpf, D. Gurdan, , and K.-M. Doth, "Design of a Flexible High Performance Quadcopter Platform Breaking the MAV Endurance Record with Laser Power Beaming," in *Proc. of the IEEE International Conference on Intelligent Robots and Systems*, 2011.

[10] S. Shen, N. Michael, and V. Kumar, "Autonomous indoor 3d exploration with a micro-aerial vehicle," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 20–25.

[11] J. Zufferey and D. Floreano, "Fly-inspired visual steering of an ultralight indoor aircraft," *IEEE Transactions of Robotics*, vol. 22, no. 1, pp. 137–146, 2006.

[12] S. Ahrens, D. Levine, G. Andrews, and J. How, "Vision-based guidance and control of a hovering vehicle in unknown, gps-denied environments," in *In International Conference on Robotics and Automation*, 2009, pp. 2643–2648.

[13] M. Bloesch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based mav navigation in unknown and unstructured environments," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 21–28.

[14] S. Weiss., D. Scaramuzza, and R. Siegwart, "Monocular-slam-based navigation for autonomous micro helicopters in gps-denied environments," *Journal of Field Robotics*, vol. 28, no. 6, 2011.

[15] D. Eberli, D. Scaramuzza, S. Weiss, and R. Siegwart, "Vision based position control for mavs using one single circular landmark," *Journal of Intelligent and Robotic Systems*, vol. 61, no. 1–4, pp. 495–512, 2011.

[16] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 957–964.

[17] P. Agarwal and M. Sharir, "Efficient algorithms for geometric optimization," *ACM Computing Surveys*, vol. 30, no. 4, pp. 412–458, 1998.

[18] A. Breitenmoser, M. Schwager, J. Metzger, R. Siegwart, and D. Rus, "Voronoi coverage of non-convex environments with a group of networked robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, USA, 2010, pp. 4982–4989.

[19] A. Breitenmoser, J. Metzger, R. Siegwart, and D. Rus, "Distributed coverage control on surfaces in 3d space," in *Proceedings of the IEEE International Conference on Robotics and Intelligent System (IROS)*, 2010.

[20] M. Schwager, B. Julian, and D. Rus, "Optimal coverage for multiple hovering robots with downward facing camera," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009, pp. 3515–3522.

[21] M. Achtelik, K.-M. Doth, D. Gurdan, and J. Stumpf, "Multi-rotor-mavs - a trade off between efficiency, dynamics and redundancy," in *Guidance, Navigation, and Control*, August 2012.

[22] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part i - the first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[23] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii - matching, robustness, and applications," *IEEE Robotics and Automation Magazine, Volume 19, issue 2*, vol. 19, no. 2, pp. 78–90, 2012.

[24] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *International Symposium on Mixed and Augmented Reality*, 2007.

[25] S. Weiss and R. Siegwart, "Real-time metric state estimation for modular vision-inertial systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[26] S. Weiss, M. W. Achtelik, M. Chli, and R. Siegwart, "Versatile distributed pose estimation and sensor self-calibration for an autonomous mav," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[27] A. Martinelli, "Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2012.

[28] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.

[29] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition, New York City, New York*, 2006, pp. 2161–2168.

[30] F. Fraundorfer, C. Engels, and D. Nister, "Topological mapping, localization and navigation using image collections," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 3872–3877.

[31] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *ECCV*, 2006, pp. 404–417.

[32] L. Kneip, D. Scaramuzza, , and R. Siegwart, "A novel parameterization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[33] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.

[34] L. Heng, G. Lee, F. Fraundorfer, and M. Pollefeys, "Real-time photo-realistic 3d mapping for micro aerial vehicles," in *Proceedings of the IEEE International Conference on Robotics and Intelligent System (IROS)*, 2011, pp. 4012–4019.

[35] E. Kosmatopoulos and A. Kouvelas, "Large-scale nonlinear control system fine-tuning through learning," *IEEE Transactions Neural Networks*, vol. 20, no. 6, pp. 1009–1023, 2009.

[36] A. Renzaglia, L. Doitsidis, A. Martinelli, and E. Kosmatopoulos, "Multi-robot three-dimensional coverage of unknown areas," *The International Journal of Robotics Research*, vol. 31, no. 6, pp. 738–752, 2012.

[37] L. Doitsidis, S. Weiss, A. Renzaglia, M. W. Achtelik, E. Kosmatopoulos, R. Siegwart, and D. Scaramuzza, "Optimal surveillance coverage for teams of micro aerial vehicles in gps-denied environments using onboad vision," *Autonomous Robots*, vol. 33, no. 1–2, pp. 173–188, 2012.

[38] M. Achtelik, S. Lynen, S. Weiss, L. Kneip, M. Chli, and R. Siegwart, "Visual-inertial slam for a small helicopter in large outdoor environments," in *Video Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.