Auton Robot (2012) 33:173–188
DOI 10.1007/s10514-012-9292-1

Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboard vision

Lefteris Doitsidis · Stephan Weiss · Alessandro Renzaglia · Markus W. Achtelik · Elias Kosmatopoulos · Roland Siegwart · Davide Scaramuzza

Received: 27 September 2011 / Accepted: 19 March 2012 / Published online: 31 March 2012
© Springer Science+Business Media, LLC 2012

Abstract This paper deals with the problem of deploying a team of flying robots to perform surveillance-coverage missions over a terrain of arbitrary morphology. In such missions, a key factor for the successful completion of the mission is the knowledge of the terrain's morphology. The focus of this paper is on the implementation of a two-step procedure that allows us to optimally align a team of flying vehicles for the aforementioned task. Initially, a single robot constructs a map of the area using a novel monocular-vision-based approach. A state-of-the-art visual-SLAM algorithm tracks the pose of the camera while, simultaneously, autonomously, building an incremental map of the environment. The map generated is processed and serves as an input to an optimization procedure using the cognitive, adaptive methodology initially introduced in Renzaglia et al. (Proceedings of the IEEE international conference on robotics and intelligent system (IROS), Taipei, Taiwan, pp. 3314–3320, 2010). The output of this procedure is the optimal arrangement of the robots team, which maximizes the monitored area. The efficiency of our approach is demonstrated using real data collected from aerial robots in different outdoor areas.

Keywords Mesh map · Mapping · Multi robot coverage · Autonomous micro aerial vehicles

L. Doitsidis (✉)
Department of Electronics, Technological Educational Institute of Crete, Chania, 73133, Greece
e-mail: ldoitsidis@chania.teicrete.gr

L. Doitsidis · E. Kosmatopoulos
Informatics & Telematics Institute, (ITI-CERTH), 57001, Thessaloniki, Greece

S. Weiss · M.W. Achtelik · R. Siegwart
Autonomous Systems Lab, ETH Zurich, Zurich, Switzerland

S. Weiss
e-mail: stephan.weiss@mavt.ethz.ch

M.W. Achtelik
e-mail: markus.achtelik@mavt.ethz.ch

R. Siegwart
e-mail: rsiegwart@ethz.ch

A. Renzaglia
INRIA Rhône-Alpes, Grenoble, France
e-mail: alessandro.renzaglia@inria.fr

E. Kosmatopoulos
Dept. of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi 67100, Greece
e-mail: kosmatop@ee.duth.gr

D. Scaramuzza
AI Lab, University of Zurich, Zurich, Switzerland
e-mail: davide.scaramuzza@ieee.org

1 Introduction

The use of multi-robot teams has gained a lot of attention in the past years. This is due to the extended capabilities that multiple robots offer with respect to a single robot for the same task. Robot teams can be used in a variety of missions, such as surveillance in hostile environments (i.e. areas contaminated with biological, chemical or even nuclear wastes), environmental monitoring (i.e. air quality monitoring, forest monitoring) and law enforcement missions (i.e. border patrol), etc. In all aforementioned tasks, there are several crucial factors that affect the overall behavior of the robot teams. These include the sensors, the size of the robot team and the type of robots used. In this paper, we introduce a methodology for optimal surveillance coverage using a team of Micro Aerial Vehicles (MAVs), based on maps created by a monocular-vision approach, which works onboard and in

real time. A state-of-the-art visual-SLAM algorithm tracks the pose of the camera while simultaneously building an incremental map of the surrounding environment. The generated map is processed and serves as an input to an optimization procedure that uses the cognitive, adaptive methodology initially introduced in Renzaglia et al. (2010, 2011) and analyzed—in terms of convergence, scalability and applicability to non-convex 3D environments—in Renzaglia et al. (2012). To define the surveillance-coverage problem we can identify two main optimization objectives that can be expressed as follows:

(O1) The part of the terrain that is "visible"—i.e. that is monitored—by the robots is maximized;
(O2) The team members are arranged so that for every point in the terrain the closest robot is as close as possible to that point.

The majority of existing approaches for multi-robot surveillance-coverage concentrate mostly on the 2D case of ground robots and deal only with one of the objectives (O1) or (O2). In most of them, the terrain morphology is assumed to be convex and/or known. In such cases, the problem of multi-robot surveillance coverage is equivalent to a standard optimization problem where the robot trajectories are generated according to a gradient-descent-like methodology.

However, in the case where it is required that both the objectives (O1) and (O2) are simultaneously addressed and the terrain's morphology is non-convex and/or unknown, standard optimization tools are no longer applicable since these tools require full knowledge of an objective function that depends on the unknown terrain's morphology. To overcome the above-mentioned shortcomings of the existing approaches for multi-robot surveillance coverage, we propose a new solution based on the recently introduced Cognitive-based Adaptive Optimization (CAO) algorithm (Kosmatopoulos 2009; Kosmatopoulos and Kouvelas 2009). The main advantage of CAO, as compared to standard optimization tools, is that it does not require the explicit knowledge of the objective function to optimize; conversely, CAO requires that at each time instant a value (measurement) of this objective function is available. As a result, if it is possible to define an appropriate objective function—whose analytical form may be unknown but is available for measurement for every given team configuration—the CAO methodology will be directly applicable to the problem of surveillance coverage treated in this paper.

Our intent is to define this objective function such that it simultaneously takes into account both the criteria (O1) and (O2) by trying to obtain a compromise between maximizing visible area and minimizing the distance of the robots to points in the environment. It has to be emphasized that, apart from rendering the optimization problem solvable, the CAO-based approach preserves additional attributes that make it particularly tractable: it can easily handle a variety of physical constraints and limitations and it is fast and scalable. These further attributes of the proposed CAO-based approach are detailed in the next section. It is finally mentioned that CAO does not create an approximation or estimation of the obstacles location and geometry; instead, it produces online a local approximation of the cost function the robots are called to optimize. For this reason, it requires simple and thus scalable approximation schemes to be employed.

The rest of the paper is organized as follows. In Sect. 2 we provide an extended literature review. In Sect. 3 we describe in detail the hardware and the software of the platform used, as long as the mono-vision framework. In Sect. 4, we give an extended description of the on line elevation mesh map generation. In Sect. 5, we present a detailed description of the CAO approach, while in Sect. 6 we provide experimental results using data obtained by real aerial robots. Finally in Sect. 7 we raise issues for discussion and future work.

2 Related work

The majority of approaches for multi-robot surveillance coverage concentrate on objective (O2) described in the previous section. In Cortés et al. (2004), the authors present a gradient-descent algorithm for the coverage of a convex region, i.e., without obstacles, with a team of mobile robots. This solution is based on the concept of centroidal Voronoi partition and adopts the Lloyd algorithm to lead the robots to the final positions. A similar approach is proposed in Schwager et al. (2006), where additionally the robots estimate a function indicating the relative importance of different areas in the environment, using information from the sensors. The same problem in a non-convex environment is more complex but also more interesting for practical applications. A possible solution to this problem is proposed in Pimenta et al. (2008). Also in this case, the solution is based on Voronoi partition, but it is obtained using the geodesic distance instead of the Euclidean one. This choice allows taking into account the particular geometry of the environment. In Howard et al. (2002), the same problem is approached by using the artificial potential field method: each robot feels a repulsive force from the obstacle and from the other robots. In this way the algorithm assures at the same time the spreading out of the team and the collision avoidance during the mission. Another possible solution for environments which include obstacles is proposed in Breitenmoser et al. (2010): the main idea is to combine the classical Voronoi coverage with the Lloyd algorithm and the local path planning algorithm TangentBug. In all the aforementioned works the regions to cover are in 2D. In Breitenmoser et al. (2010) the authors study also the problem of deploying a team of ground robots on a non-planar surface in 3D space. As far as it concerns objective (O1) described in the previous section, different solutions have been

proposed in the literature. In Ganguli et al. (2005) the authors propose a gradient-based algorithm for the case of a single robot case and they prove that the visible area is almost everywhere a locally Lipschitz function of the observer location. In Ganguli et al. (2007), an approach for the multi-robot problem is presented based on the assumption that the environment is simply connected. The visibility problem is also related with the Art Gallery Problem where the goal is to find the optimum number of guards in a non-convex environment so that each point of the environment is visible by at least one guard (O'Rourke 1987; Urrutia 2000). All the aforementioned solutions are based on the hypothesis that a given point can be monitored regardless of its distance from the robot. An incremental algorithm which takes into consideration also a maximum monitoring distance is presented in Howard et al. (2002). In Schwager et al. (2009), the authors consider the coverage of a 2D region by using a team of hovering robots. In this case, information per pixel is proposed as optimization criterion. To the best of our knowledge, the problem of considering the two objectives simultaneously to cover an arbitrary region by using a team of flying robots has never been investigated so far.

A key issue for the successful implementation of the CAO proposed methodology in the case of a team of MAVs, is the accuracy of the input it will have, which in this case is an elevation map of the environment. In this work, we consider an elevation map as a tradeoff between complex environmental mapping versus online availability of the environment shape for real-time coverage. A more sophisticated, yet much more costly approach in terms of computational complexity is presented in Triebel et al. (2006). There, the authors reconstruct the 3D environment aid of Multi Level Surface maps on a ground robot. Since MAVs generally fly at a reasonable altitude, the area is well approximated by a computationally much less expensive elevation map not considering tunnel- or cave-like structures. Since we deal with MAVs, the choice of sensors to perceive the environment to be monitored and therefore to construct the elevation maps is limited. For GPS-denied navigation and mapping, vision sensors and laser range finders might be the only options. In So and Kanade (1990) combine range images with a digital elevation model for accurate environment modelling. A computationally less complex approach was chosen by Thrun et al. (2000) using a multi-resolution approach adopted from the computer graphics literature. This approach shows real-time capabilities on a ground robot. However, on aerial vehicles, we are even more constraint in the computation power budget. Furthermore, laser scanners are too heavy for MAVs and have a limited field of view. Therefore, cameras and inertial sensors might be the only viable solutions for such limited weight and calculation power budgets. For ground vehicles (cars), 3D occupancy grids built from stereo vision and GPS data have been shown to be a valid solution (Chen and Xu 2006). However, occupancy grids are not a good option for MAVs because of their limited calculation power. Lacroix (2001) presented an off-line method to map a large outdoor scenario in fine resolution using low-altitude aerial stereo-vision images. However, stereo vision loses its advantage when the baseline is too small compared to the scene depth. Considering the limited weight, power and computation budget on MAVs we rely on a monocular solution in which the appropriate baseline is provided by a keyframe-based visual SLAM framework (Strasdat et al. 2011).

3 Platform

3.1 Hardware

The MAV we use is a so-called quadrocopter, a helicopter driven by four rotors, symmetric to the center of mass. The control of the quadrocopter is performed solely by changing the rotation speed of the propellers and is described in more detail in Gurdan et al. (2007). For our experiments, we use the "AscTec Pelican" quadrocopter (Ascending Technologies GmbH) presented in Fig. 1, which is a further development of the one described in Gurdan et al. (2007). The quadrocopter is equipped with rotors with 10″ diameter which allow to carry a payload of about 500 g. Depending on battery size and payload, flight times between 10 and 20 minutes can be achieved. Further key features are the Flight Control Unit (FCU) "AscTec Autopilot" as well as the flexible design enabling one to easily mount different payloads like computer boards or cameras. The FCU features a complete Inertial Measurement Unit (IMU) as well as two 32 Bit, 60 MHz ARM-7 microcontrollers used for data fusion and flight control. One of these microcontrollers, the Low Level Processor (LLP) is responsible for the hardware management and IMU sensor data fusion. An attitude and GPS-based position controller is implemented as well on this processor. The LLP is delivered as a black box with defined interfaces to additional components and to the High Level Processor (HLP). To operate the quadrocopter, only the LLP is necessary. Therefore, the HLP is dedicated for custom code. All relevant and fused IMU data is provided at an update rate of 1 kHz via a highspeed serial interface. In particular, this comprises body accelerations, body angular velocities, magnetic compass, height measured by an air pressure sensor and the estimated attitude of the vehicle.

For the computationally more expensive onboard processing tasks, we outfitted the helicopter with a 1.6 GHz Intel Atom Based embedded computer, available from Ascending Technologies GmbH. This computer is equipped with 1 GB RAM, a MicroSD card slot for the operating system, a 802.11n based miniPCI Express WiFi card and a Compact Flash slot. The miniPCIE WiFi card is preferred

Fig. 1 Overview of the Pelican quadrocopter

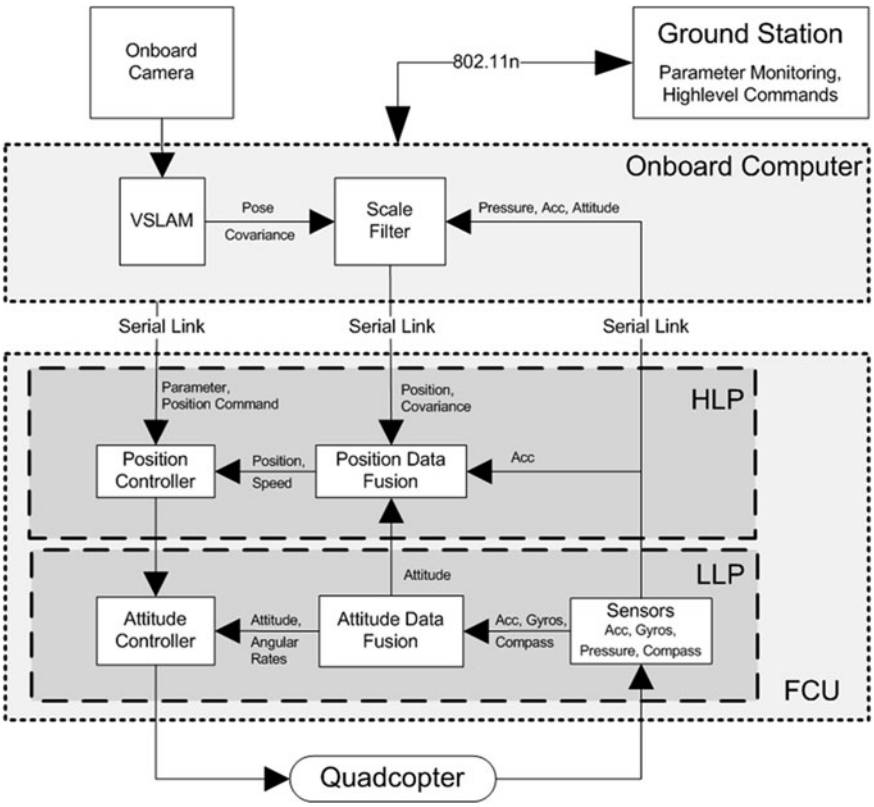


Fig. 2 Overview of the onboard schematics and interfaces

over USB to keep the USB bus free for devices like the cameras we use. We furthermore use a high speed CF-card that allows us data logging with up to 40 MByte/s.

As camera, we use a MatrixVision BluFox camera with a resolution of 752 × 480 px and a global shutter. The camera faces the ground with a 150° field-of-view lens since we are expecting the most stable features trackable over longer time in this configuration.

The configuration of our system is schematically depicted in Fig. 2.

3.2 Software

To provide a maximum portability of our code and to avoid potential (binary) driver issues, we installed Ubuntu Linux 10.04 on our onboard computer which makes tedious cross-compiling unnecessary. Since we are running a couple of different subsystems that need to communicate between each other, we use the ROS (Quigley et al. 2009) framework as a middleware. This is also used to communicate to the ground station over the WiFi datalink for monitoring

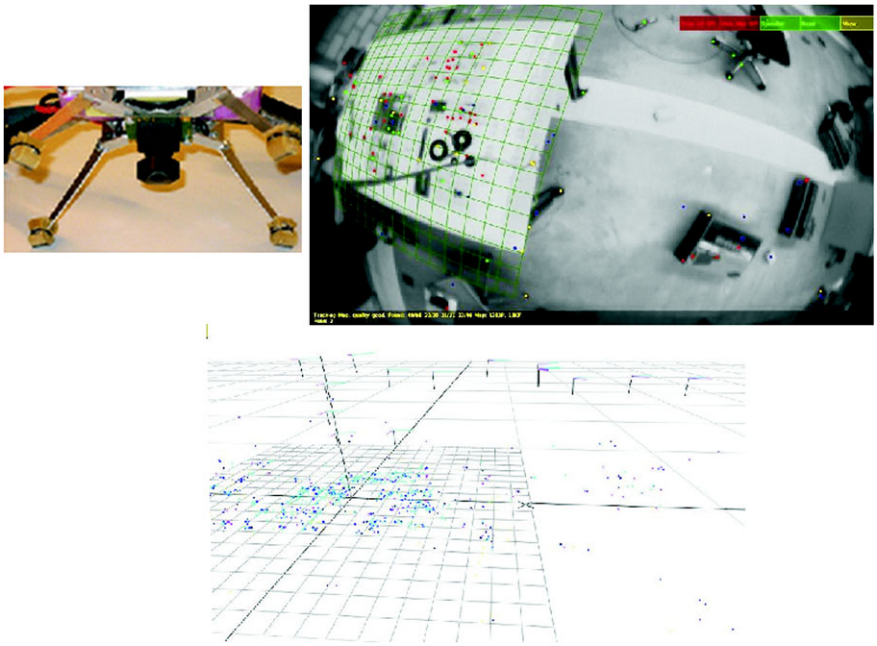


Fig. 3 The *top-left* picture depicts the onboard-mounted camera on our vehicle (the Pelican) from Ascending Technologies. The *top-right* picture is a screenshot of our visual SLAM algorithm. The tracking of features can be observed. This is used for the localization of the camera. In the *bottom* picture, the 3D point cloud map built by the mapping thread is shown. The 3-axis coordinate frames represent the location where new keyframes were added

and control purposes. The FCU is interfaced via a ROS node communicating over a serial link to the FCU's Higlevel Controller with firmware we developed for our purposes.

Software development on the HLP is done based on a SDK available for the AutoPilot FCU providing all communication routines to the LLP and a basic framework. The HLP communicates with the ROS framework on the onboard computer over a serial datalink and a ROS FCU-node handling the serial communication. This node subscribes to generic ROS pose messages with covariance, in our case from the vision framework, and forwards it to the HLP. Moreover, it allows to monitor the state of the fusion filter and the position controller, and to adjust their parameters online via the "dynamic reconfigure" functionality of ROS.

For the implementation of the position control loop and data fusion onboard the HLP, a Matlab/Simulink framework is used in combination with the Mathworks Real-Time Workshop Embedded Coder. The framework provides all necessary tools to design the control structure in Simulink, optimize it for fixed point computing, as well as compiling and flashing the HLP.

3.3 Monocular-vision framework

The approach presented here builds upon the keyframe-based visual SLAM algorithm of Klein and Murray (2007) to localize the MAV and build a dense elevation map with a single camera (see Fig. 3). This monocular SLAM algorithm and its use for MAV autonomous navigation is described in details in our previous work (Weiss et al. 2011) and (Bloesch et al. 2010).

When moving the helicopter through a region, our camera faces downwards. This increases the overlapping image portion of neighboring keyframes, so that we can even further loosen the heuristics for adding keyframes to the map. It also ensures that we can assume an elevation map later on in the meshing procedure. When exploring new areas the global bundle adjustment can be very expensive, limiting the number of keyframes to a few hundred on our platform. An intricate hurdle when using a monocular camera is the lack of any depth information. Closely linked to this problem is the unknown map scale. We tackle this issue with our approach presented in Weiss and Siegwart (2011) using an inertial sensor. We are thus able to have all distance in metric units.

3.3.1 Adaptations to the SLAM algorithm

The most evident and crucial change consists of porting the SLAM algorithm to ROS (Robot Operating System from Willow Garage). This facilitates the transfer of information to different nodes and computers. From the performance point of view, the most important change is the degeneration of the SLAM framework to a visual odometry framework: we no longer keep all keyframes in the bundle adjustment step, but only keep a constant number of them. This makes the algorithm scalable to large environments while keeping the calculation complexity linear with the number of features. If the number of keyframes exceeds a threshold, we only take the closest N keyframes to the current MAV pose. The augmentation in drift is minimal, since keyframes far away from the current MAV pose only contribute minimally in a global bundle adjustment step. Loop closure is handled passively equally to the original version of the algorithm. That is, if the loop did not drift significantly, the keyframe

which closes the loop is considered as neighbor of the current MAV pose and is taken into account in the local bundle adjustment step.

Besides the fundamental changes mentioned above, we also adapt some parameters of Klein and Murray's visual SLAM algorithm to increase its performance within our framework for optimal coverage in unknown terrain. First, we use a more conservative keyframe selecting heuristic in order to decrease the number of keyframes added during map expansion. Additionally, we reduce the number of points tracked by the tracking thread from 1000 to 300. This again increases the maximal map size and the frame rate, while keeping the accurate tracking quality. This leads to a very sparse information for the elevation mesh map, however, our tests show still very satisfying results underlining the strength of our approach for dense elevation mesh maps.

These modifications led to a framerate of max 20 Hz on an Intel ATOM 1.6 GHz processor. We demonstrate the pure navigation task (i.e. without mesh mapping the environment) in our previous work (Weiss et al. 2011).

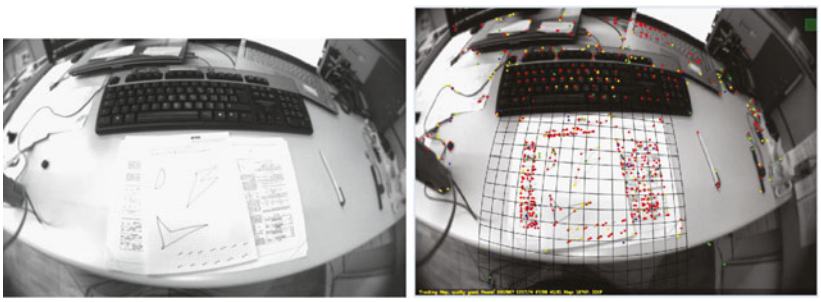
4 Online elevation-map generation

To perform optimal surveillance coverage over an arbitrary terrain, we need to reconstruct the area in an elevation map. Note that most works on optimal coverage assume an existing map. In this work, we use an approach to build an elevation map online. Thus, the MAV has to be able to fly autonomously in the yet-unknown and later-mapped area. For the vision-based autonomous navigation, we use the approach described in Sect. 3. We extended our meshing approach of our previous work in Weiss et al. (2011) to meet the needs for optimal surveillance coverage in an arbitrary terrain. In particular, we build the map iteratively while the MAVs are exploring the environment. Since we degenerated the visual framework to a visual odometry setup (c.f. Sect. 3.3) only the features triangulated with the newest keyframe are added to the meshing process. Notice that, thanks to this modification, the meshing process has constant complexity, since the number of added features per keyframe does not grow with the map size. Furthermore, the required rate of the mesh update is given by the rate of newly added keyframes. That is, it is dependent on the speed/altitude ratio the MAV moves, i.e., the rate of newly added keyframes is the same if the MAV moves fast at high altitude or slower on low altitude. It is the pixel change in the image that triggers a new keyframe. During all our experiments, we use a down-looking camera on the MAV. Thus, we can assume the point-cloud to be an elevation map.

4.1 Elevation-mesh generation from a point cloud

For the sake of completeness we summarize here the idea of our previous work in Weiss et al. (2011) for the mesh-map creation. For the sake of simplicity and for better understanding we use a sample scene throughout this section. Figure 4 depicts this scene. Note that it is a small scale scene, however, due to our monocular approach, all techniques and algorithms applied to this scene are perfectly scalable. That is, huge terrain captured from far away looks identical to a small terrain captured from very close—i.e. the images and thus the map are scale invariant. At the end of this section we show our algorithm performing in a large scale outdoor environment. Figure 4(b) shows the information available in a keyframe of the SLAM algorithm.

Assume the point cloud $\{\mathbf{p}_i\}$ with M 3D points $\mathbf{p}_i$ representing the initial map constructed by the visual SLAM algorithm in the start phase. Without any restrictions to the terrain to explore later on we assume the start area to be relatively flat and the aerial vehicle in hover mode. The main map plane H is found using a least square method on $\{\mathbf{p}_i\}$ or a RANSAC algorithm. In our case the latter one is used to be more robust against outliers. This is done in the given



(a) (b)

Fig. 4 (**a**) Sample image of the scene mapped for the following illustration of the algorithm in this section. The sheets in front of the keyboard are flat and represent the main plane H whereas the keyboard has a soft inclination in depth towards the upper part of the image. (**b**) Scene with 3D point features. This represents the data available in a keyframe of the visual SLAM algorithm. Back projecting a 3D triangle of the meshed map allows getting the texture for the triangle in question. Note that this is the distorted image while for texturing the mesh we use the undistorted one

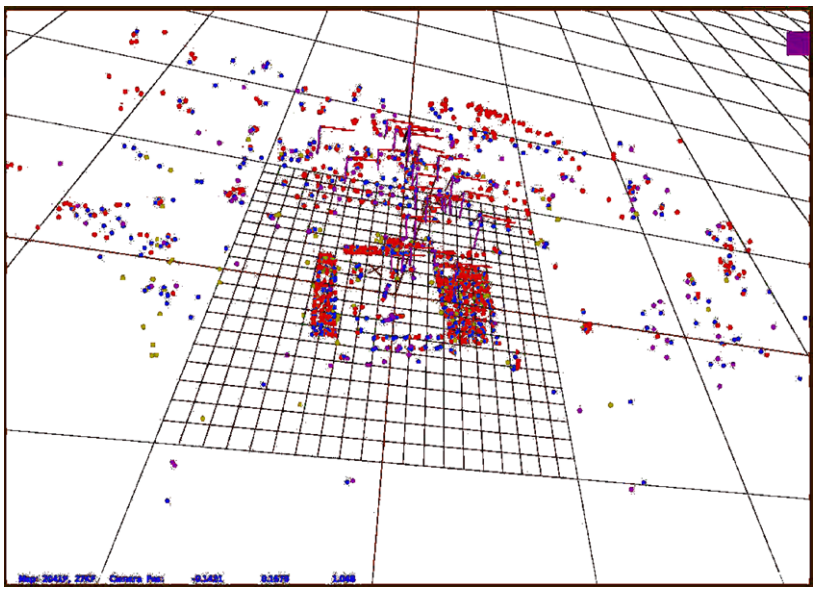


Fig. 5 The 3D point cloud of the sample scene. A trained eye can spot the papers and the keyboard. However, usually neither human users nor standard path planning and obstacle avoidance algorithms understand the point cloud

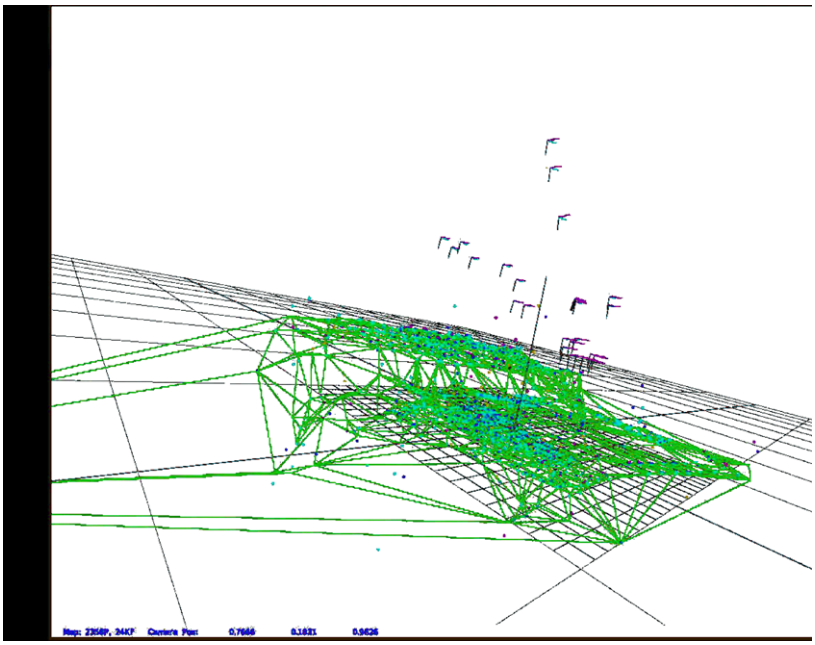


Fig. 6 Applying Delaunay Triangulation to the point cloud reveals the real topology of the scene. The 'hill' represents the keyboard in the sample scene. Note that we applied a median filter to the mesh vertices in order to eliminate outliers. Thus the 3D points may not always lie on the grid. This grid is already sufficient for path planning and obstacle avoidance

SLAM framework. All current and future map points are projected to this main plane to reduce the dimensionality:

$$\mathbf{r}_i = P * \mathbf{p}_i \tag{1}$$

where $\mathbf{p}_i$ is a three dimensional point of the current map and $\mathbf{r}_i$ is its two dimensional counterpart projected to the main map plane H using the 2×3 projection matrix P. Note that H usually corresponds to a physical plane in the scene (i.e. table or floor). Furthermore, as the camera is down looking on a helicopter this plane usually is only slightly inclined to the xy-plane in the camera frame. Thus the two dimensional positions of the features $\mathbf{r}_i$ are accurate while the third (eliminated by the projection) is very noisy due to the depth triangulation of the visual SLAM algorithm. After the projection a Delaunay Triangulation is run in 2D space to generate a 2D mesh. We use a Sweep algorithm for the triangulation to keep calculation power low. For the Sweep triangulation, calculation is in the order of $O(n \log n)$ compared to the standard algorithm with $O(n^2)$. The 3D point cloud of the scene is depicted in Fig. 5. One can note the difficulty even a trained eye has to interpret the scene. Standard path planning and obstacle avoidance algorithms cannot be used. In Fig. 6 the generated mesh is shown. After the Delaunay Triangulation in 2D space we add again the third dimension. As (1) is not invertible (P is not a square matrix and we therefore have ambiguities in the back projection)

we only use the edge information of the Delaunay Triangulation. That is if an edge in the 2D Delaunay Triangulation is defined by

$$d_{2d} = \overline{\mathbf{r}_i \mathbf{r}_j} \tag{2}$$

we map it to an edge in 3D space according to

$$d_{3d} = \overline{\mathbf{p}_i \mathbf{p}_j} \tag{3}$$

with $\mathbf{r}_k = P * \mathbf{p}_k$ and $k \in$ map. This initial elevation mesh is then median filtered in the third coordinate to remove outliers and noise. The median value is calculated using all adjacent vertices to the center vertex. That is

$$p_{zk} = \text{median}(p_{zi} \forall p_{zi} \in d_{3d} = \overline{\mathbf{p}_k \mathbf{p}_i}) \tag{4}$$

where p_{zi} denotes the third coordinate of the 3D point $\mathbf{p}_i$ previously eliminated for the Delaunay Triangulation.

At this point standard path planning and obstacle avoidance algorithms could be applied for enhanced autonomous navigation. The most simple rule for obstacle avoidance is to not traverse the mesh. That is, if the airborne vehicle always stays on the same side of the mesh it will not crash against an obstacle. Note that thanks to the sparseness of the point features this rule is highly robust, however, may be too restrictive in some particular cases. In the task of optimal coverage, we are more interested in the general shape of the landscape, rather than detailed 3D reconstruction. Thus, for the use of optimal coverage, the level of details of these elevation mesh maps is largely sufficient. Note that we can recover the absolute scale factor of the monocular SLAM by using an inertial sensor as we described in Weiss and Siegwart (2011). This way, we can reconstruct a metric mesh-map of an arbitrary terrain. Figure 7 shows the initialization of the visual SLAM algorithm and the reconstruction of our outdoor test terrain. For better visibility we added texture to the mesh map as described in Weiss et al. (2011). With the above described procedure, we are able to reconstruct metrically any environment autonomously given that sufficient (arbitrary) visual features are available. In unprepared outdoor environments, this requirement is generally fulfilled. The reconstructed mesh map of the environment can then be used by any coverage algorithm. In particular, we describe an approach using our novel CAO algorithm in the next section.

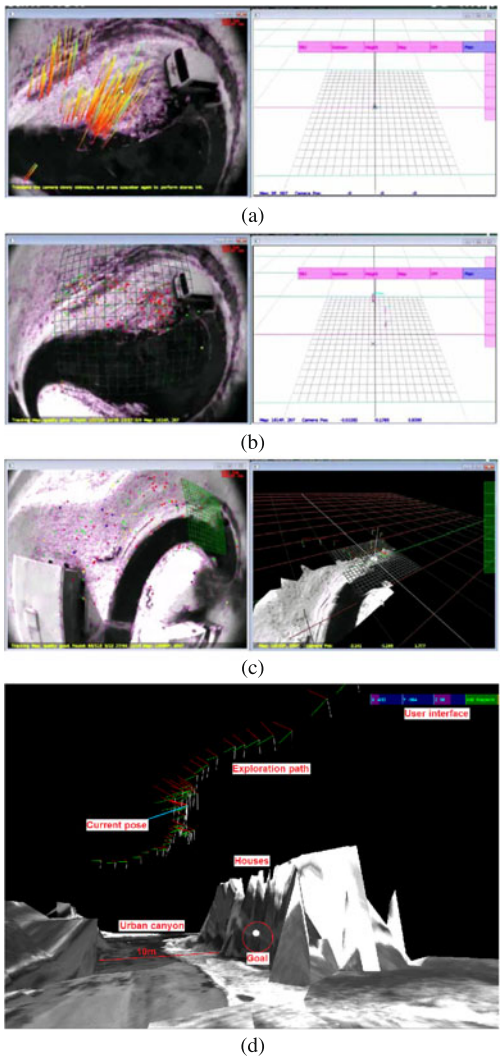


Fig. 7 (**a**) Initialization of the visual SLAM algorithm (on the *left* the tracked features used to initialize the map, on the *right* the reference frame). (**b**) The reference frame is displayed as a grid on the image (*left*). On the *right*, a few reconstructed camera poses are displayed as faint tripods. The *bold tripod* is the actual camera pose. This pose is used for the MAV position controller and yields the metric map scale by fusing it with the IMU measurements. (**c**) Generation of the textured map. (**d**) Sample of a meshed and also textured (snowy) outdoor environment. For the CAO approach the generated mesh is sufficient, however, the texture gives the user intuitive information of where the MAV is positioned at the given time instance. Even with the texturing, this approach runs in real-time. Note that the reconstruction precision is not very high. It is, however, largely sufficient for our optimal-coverage tasks. Aid of the IMU we have a metric map and estimate here the urban canyon width to be about 10 m (error is $<10\%$). The map reconstruction runs online while flying

5 The cognitive-based optimization approach

The Cognitive-based Adaptive Optimization (CAO) approach (Kosmatopoulos et al. 2007; Kosmatopoulos 2009; Kosmatopoulos and Kouvelas 2009) 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. In this section, we will describe how the CAO approach can be appropriately adapted and extended so that it is applicable to the problem of multi-robot coverage. More explicitly, let us consider the problem

where M robots are involved in a coverage task, attempting to optimize a given *coverage criterion*. Apparently, the coverage criterion is a function of the robots' positions or poses (positions and orientations), i.e.

$$J_k = \mathcal{J}\big(x_k^{(1)}, \ldots, x_k^{(M)}\big) \tag{5}$$

where $k = 0, 1, 2, \ldots$ denotes the time-index, J_k denotes the value of the coverage criterion at the kth time-step, $x_k^{(1)}, \ldots, x_k^{(M)}$ denote the position/pose vectors of robots $1, \ldots, M$, respectively, and $\mathcal{J}$ is a nonlinear function which depends, apart from the robots' positions/poses, on the particular environment where the robots live; for instance, in the 2D case the function $\mathcal{J}$ depends on the location of the various obstacles that are present, while in the 3D case with flying robots monitoring a terrain, the function $\mathcal{J}$ depends on the particular terrain morphology.

Due to the dependence of the function $\mathcal{J}$ on the particular environment characteristics, the *explicit form of the function $\mathcal{J}$ is not known* in most practical situations; as a result, standard optimization algorithms (e.g. steepest descent) are not applicable to the problem in hand. However, in most practical cases—like the one treated in this paper—the current value of the coverage criterion can be estimated from the robots' sensor measurements. In other words, at each time-step k, an estimate of J_k is available through robots' sensor measurements,

$$J_k^n = \mathcal{J}\big(x_k^{(1)}, \ldots, x_k^{(M)}\big) + \xi_k \tag{6}$$

where J_k^n denotes the estimate of J_k and ξ_k denotes the noise introduced in the estimation of J_k due to the presence of noise in the robots' sensors. Note that, although it is natural to assume that the noise sequence ξ_k is a stochastic *zero-mean* signal, it is not realistic to assume that it satisfies the typical Additive White Noise Gaussian (AWNG) property even if the robots' sensor noise is AWNG: as $\mathcal{J}$ is a nonlinear function of the robots' positions/poses (and thus of the robots' sensor measurements), the AWNG property is typically lost.

An efficient robot coverage algorithm have additionally to deal with the problem of *restricting the robots' positions* so that obstacle avoidance as well as minimum and maximum height of flight constraints are met. In other words, at each time-instant k, the vectors $x_k^{(i)}$, $i = 1, \ldots, M$ should satisfy a set of constraints which, in general, can be represented as follows:

$$\mathcal{C}\big(x_k^{(1)}, \ldots, x_k^{(M)}\big) \leq 0 \tag{7}$$

where $\mathcal{C}$ is a set of nonlinear functions of the robots' positions. As in the case of $\mathcal{J}$, the function $\mathcal{C}$ depends on the particular environment characteristics (e.g. location of obstacles, terrain morphology) and an explicit form of this function may be not known in many practical situations; however, it is natural to assume that the coverage algorithm is provided with information whether a particular selection of robots' positions/poses satisfies or violates the set of constraints (7).

Given the mathematical description presented above, the multi-robot coverage problem can be mathematically described as the problem of moving $x_k^{(1)}, \ldots, x_k^{(M)}$ to a set of positions/poses that solves the following constrained optimization problem:

$$\begin{aligned} &\text{minimize} \quad (5) \\ &\text{subject to} \quad (7). \end{aligned} \tag{8}$$

As already noticed, one of the difficulty in solving in real-time and in real-life situations the constrained optimization problem (8) lies in the fact that explicit forms for the functions $\mathcal{J}$ is not available. To circumvent this difficulty, the first step of the CAO approach is to makes use of function approximators for the estimation of the objective function $\mathcal{J}$ at each time-instant k according to

$$\hat{J}_k\big(x_k^{(1)}, \ldots, x_k^{(M)}\big) = \vartheta_k^\tau \phi\big(x_k^{(1)}, \ldots, x_k^{(M)}\big). \tag{9}$$

Here $\hat{J}_k(x_k^{(1)}, \ldots, x_k^{(M)})$ denotes the approximation of $\mathcal{J}$ generated at the kth time-step, ϕ denotes the nonlinear vector of L regressor terms, ϑ_k denotes the vector of parameter estimates calculated at the kth time-instant and L is a positive user-defined integer denoting the size of the function approximator (9). The parameter estimation vector ϑ_k is calculated according to

$$\vartheta_k = \underset{\vartheta}{\operatorname{argmin}} \frac{1}{2} \sum_{\ell=\ell_k}^{k-1} \big(J_\ell^n - \vartheta^\tau \phi\big(x_\ell^{(1)}, \ldots, x_\ell^{(M)}\big)\big)^2 \tag{10}$$

where $\ell_k = \max\{0, k - L - T_h\}$ with T_h being a user-defined nonnegative integer. Standard least-squares optimization algorithms can be used for the solution of (10). As soon as the estimator $\hat{J}_k$ is constructed according to (9), (10), the set of new robots' positions is selected as follows: firstly, a set of N candidate robots' positions is constructed according to:

$$x_k^{i,j} = x_k^{(i)} + \alpha_k \zeta_k^{i,j}, \quad i \in \{1, \ldots, M\}, \ j \in \{1, \ldots, N\}, \tag{11}$$

where $\zeta_k^{i,j}$ is a zero-mean, unity-variance random vector with dimension equal to the dimension of $x_k^{(i)}$ and α_k is a positive real sequence which satisfies the conditions:

$$\lim_{k\to\infty} \alpha_k = 0, \qquad \sum_{k=1}^{\infty} \alpha_k = \infty, \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty. \tag{12}$$

Among all N candidate new positions $x_k^{1,j}, \ldots, x_k^{M,j}$, the ones that correspond to non-feasible positions, i.e. the ones

that violate the constraints (7), are neglected and then the new robots' positions are calculated as follows:

$$\left[x_{k+1}^{(1)},\ldots,x_{k+1}^{(M)}\right]=\underset{\substack{j\in\{1,\ldots,N\}\\ x_k^{i,j}\text{ not neglected}}}{\operatorname{argmin}}\ \hat{J}_k\left(x_k^{1,j},\ldots,x_k^{M,j}\right)$$

The idea behind the above logic is simple: at each time-instant a set of many candidate new robots' positions is generated. The candidate, among the ones that provide with a feasible solution, that provides the "best" estimated value $\hat{J}_k$ of the coverage criterion is selected as the new set of robots' positions. The random choice for the candidates is essential and crucial for the efficiency of the algorithm, as such a choice guarantees that $\hat{J}_k$ is a reliable and accurate estimate for the unknown function $\mathcal{J}$; see Kosmatopoulos (2009), Kosmatopoulos and Kouvelas (2009) for more details. In the specific 3D case studied here the problem can be formulated as follows.

Consider a team of M flying robots that is deployed to monitor an unknown terrain $\mathcal{T}$. Let $z=\Phi(x,y)$ denote the unknown height of the terrain at the point (x,y) and assume for simplicity that the terrain $\mathcal{T}$ is rectangular along the (x,y)-axes, i.e. $x_{min}\le x\le x_{max}$, $y_{min}\le y\le y_{max}$. Let $\mathcal{P}=\{x^{(i)}\}_{i=1}^{M}$ denote the *configuration* of the robot team, where $x^{(i)}$ denotes the position/pose of the ith robot.

Given a particular team configuration $\mathcal{P}$, let $\mathcal{V}$ denote the *visible area* of the terrain, i.e. $\mathcal{V}$ consists of all points $(x,y,\Phi(x,y))\in\mathcal{T}$ that are visible from the robots. Given the robots' sensor capabilities, a point $(x,y,\Phi(x,y))$ of the terrain is said to be *visible* if there exists at least one robot so that

- the robot and the point $(x,y,\Phi(x,y))$ are connected by a line-of-sight;
- the robot and the point $(x,y,\Phi(x,y))$ are at a distance smaller than a given threshold value (defined as the maximum distance the robot's sensor can "see").

Apparently, the main objective for the robot team is to maximize the visible area $\mathcal{V}$. However, this cannot be the only objective for the robot team in a coverage task: trying to maximize the visible area will simply force the robots to "climb" as high as[1] possible. In parallel to maximizing the visible area, the robot team should make sure that it *minimizes the average distance between each of the robots and the terrain subarea the particular robot is responsible for*, where the terrain subarea a particular robot is responsible for, is defined as follows: given a team configuration $\mathcal{P}$, the subarea of the terrain the ith robot is responsible for is defined as the part of the terrain that (a) is visible by the ith robot and (b) each point in this subarea is closer to the ith robot than any other robot of the team. This second, and parallel to maximizing visibility, objective for the robot team is necessary for two practical reasons: (a) firstly, the closer is the robot to a point in the terrain the better is, in general, its sensing ability to monitor this point and (b) secondly, in many multi-robot coverage applications there is the necessity of being able to intervene as fast as possible in any of the points of the terrain with at least one robot. Having in mind that the robot team has to successfully meet the two above-described objectives, we define the following combined objective function the robot team has to minimize:

$$J(\mathcal{P})=\int_{q\in\mathcal{V}}\min_{i\in\{1,\ldots,M\}}\left|x^{(i)}-q\right|^2dq+K\int_{q\in\mathcal{T}-\mathcal{V}}dq \quad (13)$$

where K is a large user-defined positive constant. The first of the terms in the above equation is related to the second objective (minimize the average distance between the robots and the subarea they are responsible for) and the second term is related to the invisible area in the terrain ($\int_{q\in\mathcal{T}-\mathcal{V}}dq$ is the total part of the terrain that is not visible by any of the robots). The positive constant K is used to make sure that both objectives are met. To see this, consider the case where $K=0$, in which case we will have that the robots, in their attempt to minimize their average distance to the subarea they are responsible for, may also seek to minimize the total visible area. On the other hand, in case where the first of the terms in (13) is absent, we will have the situation mentioned above where the robots in their attempt to maximize the visible area will have to "climb" as high as they are allowed to.

It has to be emphasized that the positive constant K should be chosen sufficiently large so that the second term in (13) dominates the first term unless no or a negligible part of the terrain remains invisible. In this way, minimization of (13) is equivalent to firstly making sure that all—or almost all—of the terrain is visible and then to locate the robots so that their average distance to the subarea they are responsible for is minimized.

A large choice for the positive term K plays another crucial role for the practical implementation of the CAO algorithm in multi-robot coverage applications: the problem with the performance index defined in (13) is that its second term $\int_{q\in\mathcal{T}-\mathcal{V}}dq$ *cannot be, in general, computed in practice*; as this term involves the part of the terrain that is not currently visible, its computation requires that the geometry of this part is known or equivalently, as the invisible part changes with the evolution of the team's configuration, that the whole terrain is known. To overcome this problem, instead of minimizing (13) the following performance index is

[1]Note also that in the case where there are no limits for the robot's maximum height and the maximum sensing distance, it suffices to have a single robot at a very high position to monitor the whole terrain.

actually minimized by the CAO approach:

$$\bar{J}(\mathcal{P}) = \int_{q\in\mathcal{V}} \min_{i\in\{1,\ldots,M\}} \left|x^{(i)} - q\right|^2 dq + K \int_{(x,y,\phi(x,y))\in\mathcal{T}-\mathcal{V}} \mathcal{I}(x,y)\,dx\,dy \quad (14)$$

where $\mathcal{I}(q)$ denotes the indicator function that is equal to 1 if the point $(x, y, \phi(x, y))$ belongs to the invisible area of the terrain and is zero, otherwise. In other words, in the cost criterion $\bar{J}(\mathcal{P})$ and for the whole invisible area, the unknown terrain points $(x, y, \phi(x, y))$ are replaced by $(x, y, 1)$, i.e. $\bar{J}(\mathcal{P})$ assumes that the whole invisible area is a flat subarea.

It is not difficult for someone to see that the replacement of the cost criterion (13) by the criterion (14) has a negligible implication in the team's performance: as a large choice for K corresponds to firstly making sure that the whole terrain is visible and then to minimizing the average distance between the robots and their responsible subareas, minimizing either of criteria (13) or (14) is essentially the same.

An efficient trajectory generation algorithm for optimal coverage—i.e. for minimization of the cost criteria (13) or (14)—must make sure that the physical constraints are also met throughout the whole multi-robot coverage application. Such physical constraints include, but are not limited to, the following ones:

- The robots remain within the terrain's limits, i.e. they remain within $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$ in the x- and y-axes, respectively.
- The robots satisfy a maximum height requirement while they do not "hit" the terrain, i.e. they remain within $[\Phi(x, y) + d, z_{max}]$ along the z-axis, where d denotes the minimum safety distance (along the z-axis) the robots' should be from the terrain and z_{max} denotes the maximum allowable height for the robots.
- The robots do not come closer on to each other than a minimum allowable safety distance d_r.

It is not difficult for someone to see that all the above constraints can be easily cast in the form (7) and thus can be handled by the CAO algorithm.

6 Experimental results

To validate our approach in a realistic environment, we have used two different data sets which were collected with the use of a miniature quadrocopter specially designed for the needs of the European project sFLY (www.sfly.org). These data sets are used as input, in an optimization framework developed in C#, which runs the CAO algorithm off board and produces the optimal positions of the robot team, in terms of terrain surveillance coverage.

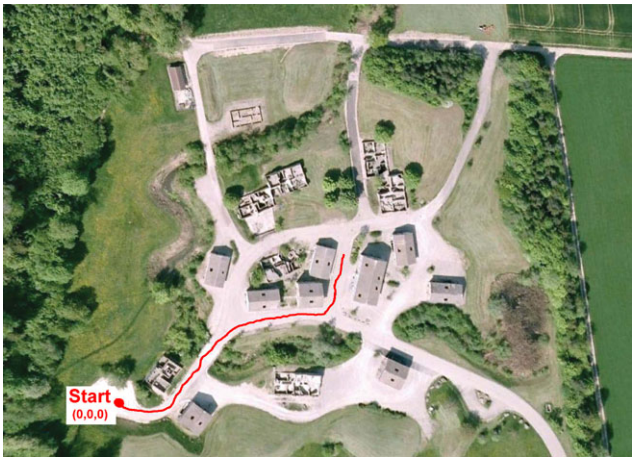


Fig. 8 Outdoor flight path through the Birmensdorf area

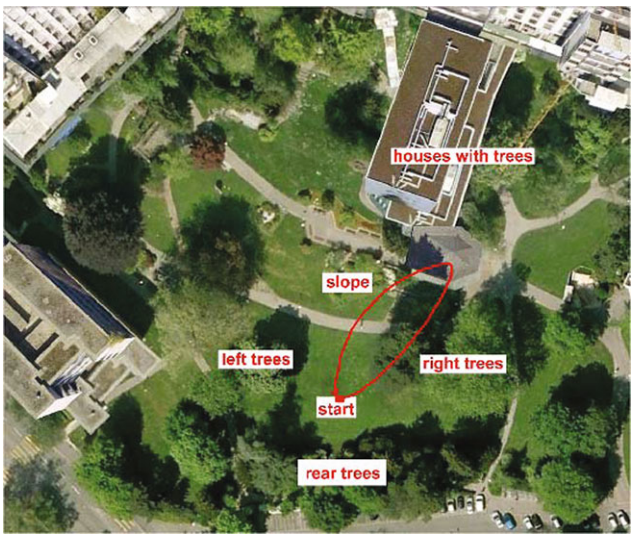


Fig. 9 Outdoor flight path through the ETHZ's hospital area

The scenarios tested consider a team of four MAVs and correspond into two different areas. The first area is Birmensdorf in Switzerland and it's presented in Fig. 8, while the second area corresponds to the ETHZ's hospital area and it's presented in Fig. 9. More details about the data and the methodology used to extract them, are presented in Bloesch et al. (2010) and Weiss et al. (2011).

The main constraints imposed to the robots are that they remain within the terrain's limits, i.e. within $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$ in the x- and y-axes, respectively. At the same time they have to satisfy a maximum height requirement while they do not "hit" the terrain, i.e. they remain within $[\Phi(x, y) + d, z_{max}]$ along the z-axis. Several initial configurations for each scenario were tested. The values of the cost function for three different configurations, in the case of the Birmensdorf area are presented in Fig. 10. Sample trajectories for a robot team are presented in Fig. 11, while in Fig. 12 the final positions of 3 robot teams starting

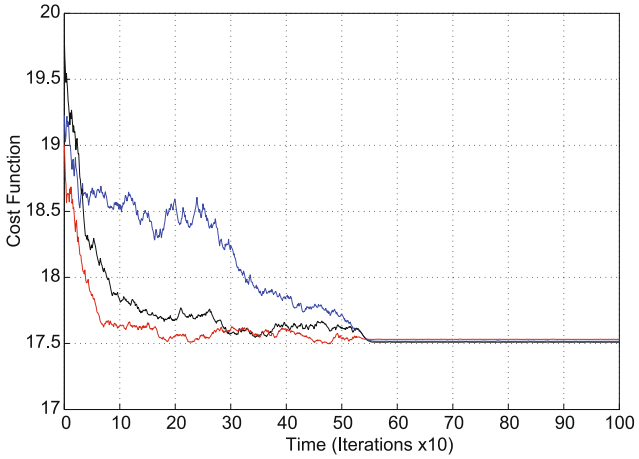


Fig. 10 Comparative cost functions for different initial robot team configurations in Birmensdorf area

from different initial positions are presented in a 3D view. Different marker type corresponds to different robots, while different color corresponds to a different team. In Table 1 the final coverage percentage for different initial configurations in the Birmensdorf area, is presented. The values of the cost function for three initial configurations in the case ETHZ's hospital area are presented in Fig. 13. Sample trajectories for a robot team are presented in Fig. 14. In Fig. 15 the final positions of 3 robot teams starting from different initial positions are presented in a 3D view.

To validate the efficiency of the proposed methodology, an incremental scenario is also presented. A single aerial robot is flying over an unknown area and incrementally is producing maps which are used as an input to the proposed CAO algorithm. Each increment is a subset of the following map. The result of the optimization procedure for each map is the position which assures optimal coverage of the area

Fig. 11 3D Path followed by a robot team in a coverage scenario in Birmensdorf area

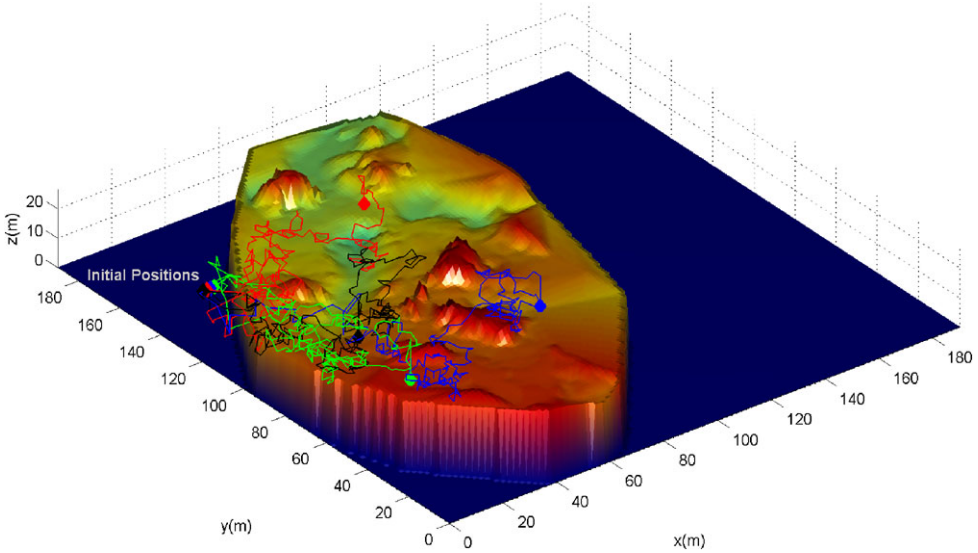
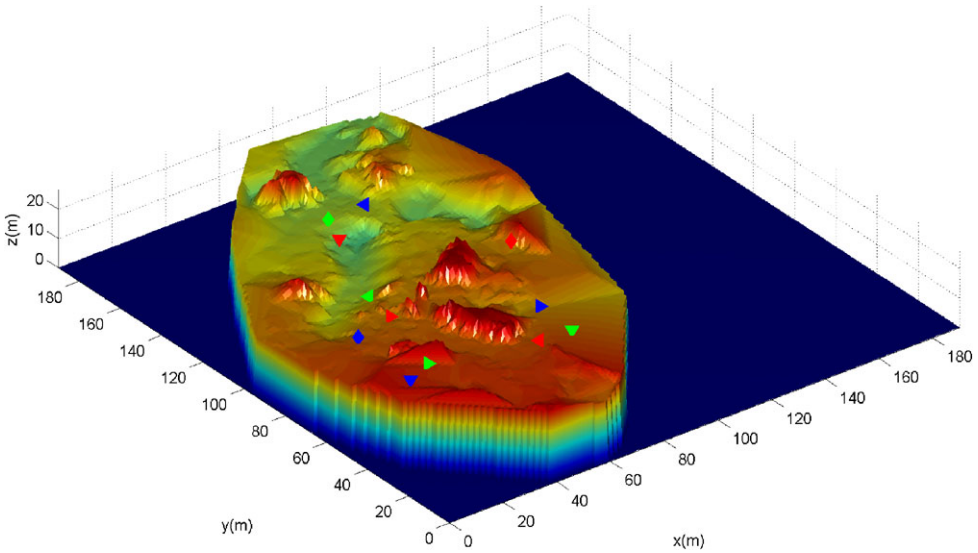


Fig. 12 Final configurations of three robot teams starting from different initial positions for the Birmensdorf area



with the given team. This optimal positions are used as an input to the new map which is produced by the aerial robot which performs the mapping procedure. An aerial robot has flew over the Birmensdorf area and based on this flight 8 successive maps of different sizes were produced and used as an input to the CAO algorithm. In Table 2 we present the performance of a team of four robots for the 8 successive maps, in term of coverage percentage. In all cases the proposed framework provided satisfactory results in terms of coverage percentage.

Table 1 Coverage percentage for different initial configurations in the Birmensdorf area

(% of coverage)			
Test case	1	2	3
Initial configuration	44.49	40.49	56.81
Final configuration	98.55	99.52	99.56

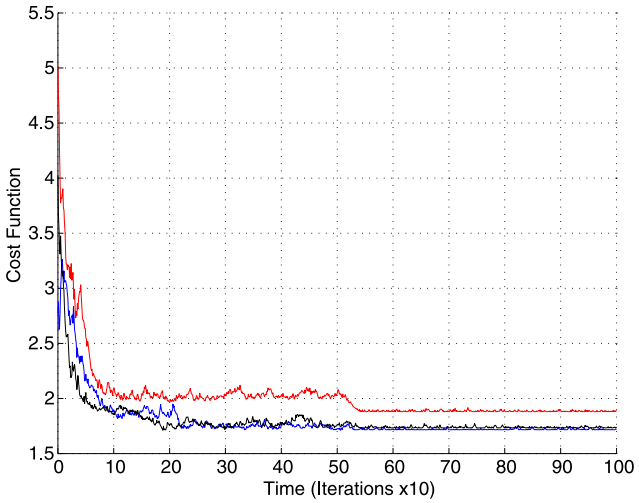


Fig. 13 Comparative cost functions for different initial robot team configurations in ETHZ's hospital area

7 Discussion and conclusions

A two-step procedure to align a swarm of flying vehicles to perform surveillance coverage has been presented and formally analyzed. Initially a state-of-the-art visual-SLAM algorithm tracks the pose of the camera while, simultaneously, building an incremental map of the surrounding environment, autonomously, given that sufficient (arbitrary) visual features are available. In unprepared outdoor environments, the requirement of having sufficient features is generally fulfilled. The reconstructed mesh map of the environment is used as the input to the second part of the procedure where a cognitive based methodology is used to maximize the area monitored by a team of aerial robots. The proposed approach has the following key advantages with respect to previous works:

– it does not require any a priori knowledge on the environment;
– it works in any given environment, without the necessity to make any kind of assumption about its topology;
– it can incorporate any kind of constraints;
– its complexity is low allowing real time implementations;
– it requires low weight and cost sensors, which makes it ideal for aerial robot applications;

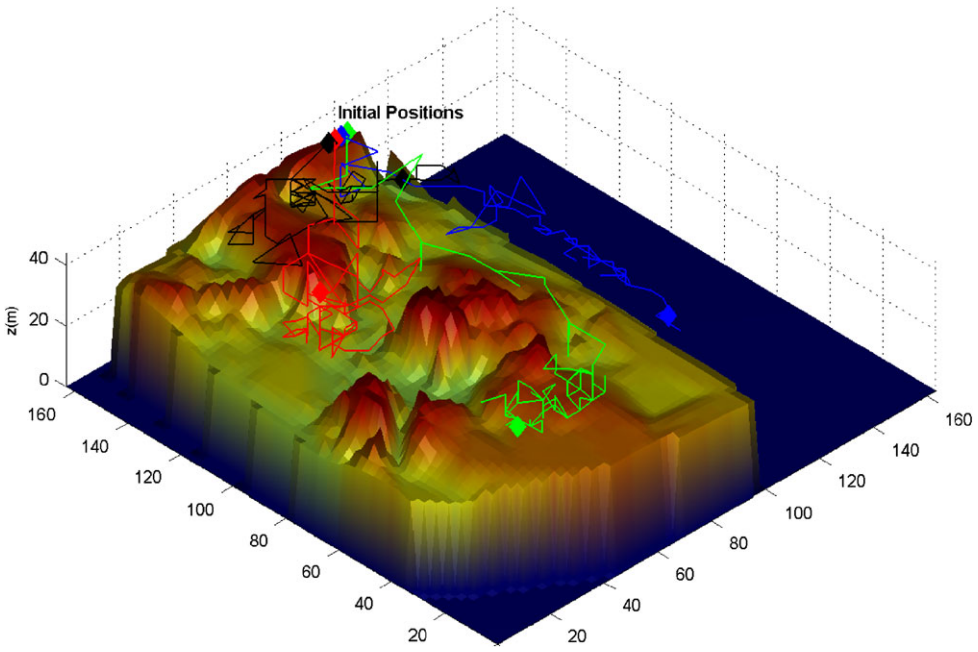


Fig. 14 3D Path followed by a robot team in a coverage scenario in the ETHZ's hospital area

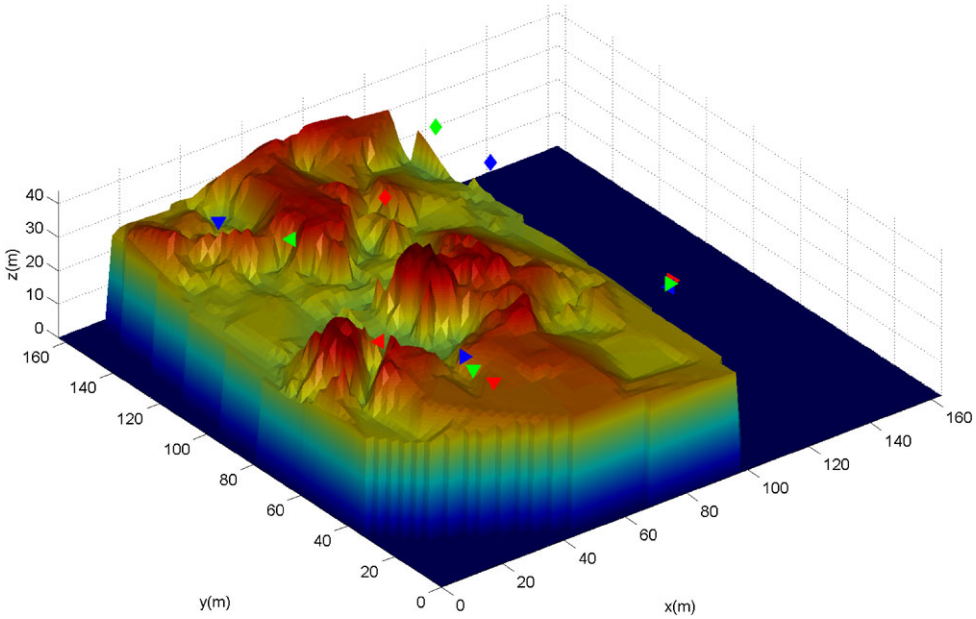


Fig. 15 Final configurations of three robot teams starting from different initial positions for the ETHZ's hospital area

Table 2 Incremental scenario in the Birmensdorf area

Test case	1	2	3	4	5	6	7	8
Initial % of coverage	69.83	85.37	63.82	65.57	49.94	75.32	74.2	81.21
Final % of coverage	94.5	98.01	95.44	95.32	72.56	79.56	76.72	90.5
% of the final map	5.46	6.55	9.63	16.86	59.98	70.23	81.8	100

– it builds itself the metric map required for the optimization procedure.

The advantages of the proposed methodology make it suitable for real implementations and the results obtained through experimentation give us the motivation to adopt the CAO also in other frameworks. We are interested into expanding the proposed approach by using the distributed version of the CAO algorithm introduced in Renzaglia et al. (2011) for 2D environments. In the aforementioned work, the aim of each member of a robotic team, was to minimize the overlapping of its field of view with the obstacles of the environment and with the fields of view of the other robots. In other words, for each robot the problem is like maximizing the surface monitored while it is moving in an environment with both static obstacles and dynamic obstacles, which are the fields of view of the other robot. This approach is closer to real world applications since it does not depended into a centralized scheme with all the known disadvantages. This approach will allow us to include communications constraints. We are also interested in incorporating more realistic constraints including sensor limitations. To properly adapt this approach significant implementation challenges exist in the case of real aerial vehicles, related mainly with computational power limitations. The same approach appropriately modified is currently under investigation for coordinated exploration and target tracking, where the CAO algorithm is combined with an extended kalman filter estimator. We expect that many important tasks in mobile robotics can be approached by CAO-based algorithms, due to the fact that the CAO approach does not require an a priori knowledge of the environment and it has low complexity. Both these issues are fundamental in mobile robotics.

Acknowledgements The research leading to these results has received funding from the European Communities Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 231855 (sFly).

References

Ascending Technologies GmbH, website, http://www.asctec.de.

Bloesch, M., Weiss, S., Scaramuzza, D., & Siegwart, R. (2010). Vision based MAV navigation in unknown and unstructured environments. In *IEEE international conference on robotics and automation (ICRA)* (pp. 21–28).

Breitenmoser, A., Metzger, J., Siegwart, R., & Rus, D. (2010). Distributed coverage control on surfaces in 3D space. In *IEEE international conference on robotics and intelligent system (IROS)*, Taipei, Taiwan (pp. 5569–5576).

Breitenmoser, A., Schwager, M., Metzger, J., Siegwart, R., & Rus, D. (2010). 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 (pp. 4982–4989).

Chen, H., & Xu, Z. (2006). 3D map building based on stereo vision. In *Proceedings of the IEEE international conference on networking, sensing and control, ICNSC* (pp. 969–973).
Cortés, J., Martínez, S., Karataş, T., & Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, *20*(2), 243–255.
Ganguli, A., Cortés, J., & Bullo, F. (2005). Maximizing visibility in nonconvex polygons: nonsmooth analysis and gradient algorithm design. *American Control Conference*, *2*, 792–797.
Ganguli, A., Cortés, J., & Bullo, F. (2007). Visibility-based multiagent deployment in orthogonal environments. In *American control conference*, New York, USA (pp. 3426–3431).
Gurdan, D., Stumpf, J., Achtelik, M., Doth, K.-M., Hirzinger, G., & Rus, D. (2007). Energy-efficient autonomous four-rotor flying robot controlled at 1 kHz. In *IEEE international conference on robotics and automation (ICRA)* (pp. 361–366).
Howard, A., Matarić, M. J., & Sukhatme, G. S. (2002). Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem. In *Proceedings of the 6th international conference on distributed autonomous robotic system (DARS)* (pp. 299–308).
Howard, A., Matarić, M. J., & Sukhatme, G. S. (2002). An incremental deployment algorithm for mobile robot teams. In *Proceedings of the IEEE international conference on robotics and intelligent system (IROS)*, Lausanne, Switzerland (pp. 2849–2854).
In So, K., & Kanade, T. (1990). High resolution terrain map from multiple sensor data. In *Intelligent robots and systems*.
Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *International symposium on mixed and augmented reality* (pp. 225–234).
Kosmatopoulos, E. B. (2009). An adaptive optimization scheme with satisfactory transient performance. *Automatica*, *45*(3), 716–723.
Kosmatopoulos, E. B., & Kouvelas, A. (2009). Large-scale nonlinear control system fine-tuning through learning. *IEEE Transactions on Neural Networks*, *20*(6), 1009–1023.
Kosmatopoulos, E. B., Papageorgiou, M., Vakouli, A., & Kouvelas, A. (2007). Adaptive fine-tuning of nonlinear control systems with application to the urban traffic control strategy TUC. *IEEE Transactions on Control Systems Technology*, *15*(6), 991–1002.
Lacroix, S., Jung, I., & Mallet, A. (2001). Digital elevation map building from low altitude stereo imagery. In *Proc. of the 9 th int. symposium on intelligent robotic systems*.
O'Rourke, J. (1987). *Art gallery theorems and algorithms*. New York: Oxford University Press.
Pimenta, L., Kumar, V., Mesquita, R. C., & Pereira, G. (2008). Sensing and coverage for a network of heterogeneous robots. In *47th IEEE conference on decision and control*, Cancun, Mexico (pp. 3947–3952).
Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). ROS: an open-source robot operating system. In *ICRA workshop on open source software*.
Renzaglia, A., Doitsidis, L., Martinelli, A., & Kosmatopoulos, E. B. (2010). Cognitive-based adaptive control for cooperative multirobot coverage. In *Proceedings of the IEEE international conference on robotics and intelligent system (IROS)*, Taipei, Taiwan (pp. 3314–3320).
Renzaglia, A., Doitsidis, L., Martinelli, A., & Kosmatopoulos, E.B. (2011). Adaptive-based distributed cooperative multi-robot coverage. In *Proceedings of the American control conference (ACC)*, San Francisco, CA, USA.
Renzaglia, A., Doitsidis, L., Martinelli, A., & Kosmatopoulos, E. B. (2012, in press). Multi-robot 3d coverage of unknown areas. *International Journal of Robotics Research*. doi:10.1177/0278364912439332.
Schwager, M., McLurkin, J., & Rus, D. (2006). Distributed coverage control with sensory feedback for networked robots. In *Proceedings of robotics: science and systems*, Philadelphia, USA.
Schwager, M., Julian, B. J., & Rus, D. (2009). Optimal coverage for multiple hovering robots with downward facing camera. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, Kobe, Japan (pp. 3515–3522).
Strasdat, H., Montiel, J. M. M., & Davison, A. J. (2011). Real-time monocular SLAM: why filter. In *IEEE international conference on robotics and automation (ICRA)* (pp. 2657–2664).
Thrun, S., Burgard, W., & Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proc. of the IEEE international conference on robotics and automation (ICRA)* (Vol. 1, pp. 321–328).
Triebel, R., Pfaff, P., & Burgard, W. (2006). Multi-level surface maps for outdoor terrain mapping and loop closing. In *2006 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2276–2282).
Urrutia, J. (2000). Art gallery and illumination problems. In *Handbook of computational geometry* (pp. 973–1027).
Weiss, S., & Siegwart, R. (2011). Real-time metric state estimation for modular vision-inertial systems. In *IEEE international conference on robotics and automation (ICRA)* (pp. 4531–4537).
Weiss, S., Achtelik, M., Kneip, L., Scaramuzza, D., & Siegwart, R. (2011). Intuitive 3D maps for MAV terrain exploration and obstacle avoidance. *Journal of Intelligent & Robotic Systems*, *61*, 473–493.
Weiss, S., Scaramuzza, D., & Siegwart, R. (2011). Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments. *Journal of Field Robotics*, *28*(6), 1–21.

Lefteris Doitsidis received his diploma degree from the Production Engineering and Management Department of the Technical University of Crete, Chania, Greece, in 2000, 2002 and 2008, respectively. From 2002 to 2008, he has been a researcher at the Intelligent Systems and Robotics Laboratory of the same department. From August 2003 to June 2004 he was a visiting scholar at the Department of Computer Science and Engineering, University of South Florida, FL, USA. He was a member of the Center of Robot Assisted Search and Rescue. Since 2004 he is with the Department of Electronics, Technological Educational Institute of Crete where he is currently an Assistant Professor. He is also an adjunct senior researcher at the Informatics & Telematics Institute of Greece, CERTH, since 2010. He is the author of 30 publications, in international journals, conference proceedings and book chapters. His research interests lie in the areas of multirobot teams, design of novel control systems for robotic applications, autonomous operation and navigation of unmanned vehicles, cooperative control and optimization. He is also active in the areas of fuzzy logic and evolutionary computation. Currently he is involved in the sFLY EU/FP7 research project and he is the technical project manager of the NOPTILUS EU/FP7 research project.

Stephan Weiss received his Master degree in Electrical Engineering and Information Technology from the Federal Institute of Technology (ETH) in Zurich, Switzerland in 2008. Since then, he is working as a PhD candidate at the Autonomous Systems Lab of ETH under Roland Siegwart. His research interests lie in the area of autonomous navigation for micro aerial vehicles. The focus lies on multi-sensor fusion and self-calibration for real-world power-on-and-go systems with special interest in vision based approaches.

Alessandro Renzaglia received his B.Sc. and M.Sc. degrees in physics from the University of Rome "La Sapienza", Italy in 2005 and 2007, respectively. He is currently working toward the Ph.D. degree in applied mathematics and computer science at INRIA Rhône-Alpes, Grenoble, France. His research interests lie in the area of multirobot systems, cooperative control and optimization.

Markus W. Achtelik received his Diploma degree in Electrical Engineering and Information Technology in 2009 from the Technische Universitat Munchen. Since then, he is working as a Ph.D. candidate at the Autonomous Systems Lab of ETH Zurich under Roland Siegwart. His main research interest lies in autonomous operation and state estimation of multiple micro aerial vehicles fusing information from IMU and vision based approaches.

Elias Kosmatopoulos received the Diploma, M.Sc. and Ph.D. degrees from the Technical University of Crete, Greece, in 1990, 1992, and 1995, respectively. He is currently an Associate Professor with the Department of Electrical & Computer Engineering, Democritus University of Thrace, Greece. He was an Assistant Professor with the Department of Production Engineering and Management, Technical University of Crete (TUC), Greece and a Research Assistant Professor and Research Associate Professor with the Department of Electrical Engineering-Systems, University of Southern California (USC) and a Postdoctoral Fellow with the Department of Electrical & Computer Engineering, University of Victoria, B.C., Canada. Dr. Kosmatopoulos' research interests are in the areas of neural networks and intelligent control. He is the author of over 40 journal papers and book chapters and over 80 conference publications, on intelligent and learning control and optimization techniques and their application to various areas such as traffic and transportation systems, process systems, energy efficient buildings, structural systems, and intelligent vehicles and robotics

Roland Siegwart is a full professor for autonomous systems at ETH Zurich since July 2006. He has a Diploma in Mechanical Engineering (1983) and Ph.D. in Mechatronics (1989) from ETH Zurich. Previous positions include professor at EPFL (1996–2006) R&D director at MECOS Traxler AG (1990–1990) and senior researcher in robotics at ETH (1990–1996). He holds visiting positions at Stanford University (1989/90 and 2005) and at NASA Ames (2005). He was co-initiator and founding Chairman of Space Center EPFL, Vice Dean of the School of Engineering EPFL and currently he is director of studies in ME at ETH. Roland Siegwart is member of the Swiss Academy of Engineering Sciences and the Research Council of the Swiss National Science Foundation, and IEEE Fellow. He served as Vice President for Technical Activities (2004/05), Distinguished Lecturer (2006/07) and AdCom Member (2007–2009) of the IEEE Robotics and Automation Society. He is coordinator of various large projects and co-founder of several spin-off companies.

Davide Scaramuzza got his Ph.D. (2008) in computer vision and robotics at ETH Zurich. His Ph.D. thesis won the Robotdalen Scientific Award, sponsored by EU and IEEE. He worked as post-doc at both ETH Zurich Autonomous Systems Lab and at UPenn GRASP Lab, Philadelphia. He is currently professor in human-oriented robotics at the University of Zurich and is the coordinator of the European project "sFly," which focuses on autonomous navigation of teams of micro UAVs in urban environments using vision as sole sensor modality. He is also lecturer of the Masters course "Autonomous Mobile Robots". He is author of the first open-source "Omnidirectional Camera Calibration Toolbox for MATLAB" and co-author of the book "Introduction to Autonomous Mobile Robots," published by MIT Press. Finally, he is reviewer for many international conferences and journals. He is member of the IEEE Robotics and Automation Society and first author of several top-ranked journals in both computer vision and robotics.