
Design and Development of Aerial Robotic Systems for Sampling Operations in Industrial Environment

Rodrigo Munguia, Edmundo Guerra, Sarquis Urzua,
Yolanda Bolea and Antoni Grau

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.70005>

Abstract

This chapter describes the development of an autonomous fluid sampling system for outdoor facilities, and the localization solution to be used. The automated sampling system will be based on collaborative robotics, with a team of a UAV and a UGV platform travelling through a plant to collect water samples. The architecture of the system is described, as well as the hardware present in the UAV and the different software frameworks used. A visual simultaneous localization and mapping (SLAM) technique is proposed to deal with the localization problem, based on authors' previous works, including several innovations: a new method to initialize the scale using unreliable global positioning system (GPS) measurements, integration of attitude and heading reference system (AHRS) measurements into the recursive state estimation, and a new technique to track features during the delayed feature initialization process. These procedures greatly enhance the robustness and usability of the SLAM technique as they remove the requirement of assisted scale initialization, and they reduce the computational effort to initialize features. To conclude, results from experiments performed with simulated data and real data captured with a prototype UAV are presented and discussed.

Keywords: UAV, collaborative robotics, inertio-visual odometry, visual SLAM, GPS

1. Introduction

The development of aerial robots has become one of the most active fields of research in the last decade. Innovations in multiple fields, like the lithium polymer batteries, microelectromechanical sensors, more powerful propellers, and the availability of new materials and prototyping technologies, have opened the field to researchers and institutions, who used to be denied

access given the costs, both economic and in specialized personnel. The new-found popularity of this research field has led to the proliferation of advancements in several areas [1] which used to ignore the possibilities of aerial robots given the limited capacities they presented.

One of the environments where aerial robots are putting a foothold for the first time is industry. Given the level of accountability, certification, and responsibility required in industry, the field had been always reluctant to the introduction of experimental state of the art technologies. But thanks to the wider experimentation in aerial robots, resilience and performance robustness levels have been improving, making them an option when solving several industry problems. Currently, these problems are focused in logistical aspects and related operations, like distribution of goods and placement of them in otherwise hard to access points. An example of this kind of applications would be the surveying and monitoring of fluids in installations with multiple basins and tanks, for example, a wastewater processing plant.

Basin sampling operations generally require multiple samples of fluids in several points with a given periodicity. This makes the task cumbersome, repetitive, and depending on the features of the environment and other factors, potentially dangerous. As such, automatization of the task would provide great benefits, reducing the efforts and risks taken by human personnel, and opening options in terms of surveying scheduling.

One of the challenges that any autonomous Unmanned Aerial Vehicle (UAV) has to face is that of estimating its pose with respect to the relevant navigational frames with accuracy. The estimation methodology here discussed was formulated for estimating the state of the aerial vehicle. In this case, the state is composed of the variables defining the location and attitude as well as their first derivatives. The visual features seen by the camera are also included into the system state. On the other hand, the orientation estimation can be estimated in a robust manner by most flight management units (FMUs), with the output of the attitude and heading reference system (AHRS) frequently used as a feedback to the control system for stabilization.

In order to account for the uncertainties associated with the estimation provided by the attitude and heading reference systems (AHRSs), the orientation is included into the state vector and is explicitly fused into the system. Regarding the problem of position estimation, it cannot be solved for applications that require performing precise maneuvers, even with global positioning system (GPS) signal available. Therefore, some additional sensory information is integrated into the system in order to improve its accuracy, namely monocular vision.

The use of a monocular camera as the unique sensory input of a simultaneous localization and mapping (SLAM) system comes with a difficulty: the estimation of the robot trajectory, as well as the features map, can be carried out without metric information. This problem was pointed out since early approaches like [2]. If the metric scale wants to be recovered, it is necessary to incorporate some source of metric information into the system. In this case, the GPS and the monocular vision can operate in a complementary manner. In the proposed method, the noisy GPS data are used to incorporate metric information into the system, in periods where it is available. On the other side, the monocular vision is used for refining the estimations when the GPS is available or for performing purely visual-based navigation in periods where the GPS is unavailable.

In this work, we present an automated system designed with the goal of automate sampling tasks in an open air plant and propose a solution for the localization problem in industrial environments when GPS data are unreliable. The final system should use two autonomous vehicles: a robotic ground platform and a UAV, which would collaborate to collect batches of samples in several tanks. The specifications and designs of the system are described, focusing on the architecture and the UAV. The next section describes the vision-based solution proposed to deal with the localization for navigation problem, commenting several contributions done with respect to a classical visual SLAM approach. Results discussing the performance and accuracy of said localization techniques are presented, both based on simulations and real data captured with the UAV described in Section 2. Finally, the conclusions discuss the next step in the testing and development of the system and the refining of the localization technique.

2. System architecture

The architecture proposed to deal with the fluid sampling task aims to maximize the capability to reach with accuracy the desired points of operation and measurement; and minimize the risks associated with the process. The risks for human operators are removed or minimized, as they can perform their tasks without exposing themselves to the outdoor industrial environment. To achieve this, the system will present two different robotic platforms: a quadcopter UAV acts as sample collector, picking fluid samples from the tanks; and a Unmanned Ground Vehicle (UGV) platform acts as a collector carrier, transporting both the collector and the samples. As the risks associated with the operation of the sample collector UAV are mainly related to the flight operations, the UAV will travel generally safely landed on the collector carrier, where it could be automatically serviced with replacement sample containers or battery charges.

2.1. Sampling system architecture and communications

The designed architecture of the system and its expected operation process can be observed in **Figure 1**. The architecture has been divided into several blocks, so it can fit into a classical deployment scheme in an industrial production environment. The analytics technicians at the laboratory can use the scheduling and control module interface to order the collection of a batch of samples. This is called collection order, detailing a number of samples, which basin must they come from, and if there are any required sampling patterns or preferences. The process can be set to start at a scheduled time, and stops may be enforced, i.e., samples on a specific tank cannot be taken before a set hour.

The collection order is formed by a list of GPS coordinates (with optional parameters, like times to perform operations), which is processed in the central server of the system. This produces a mission path, which includes a route for the sample carrier, with one or more stops. The mission path also includes the sampling flight that the sample collector UAV has to perform with the sample carrier stop. The data of this flight, called collection mission, will be transmitted by the sample carrier to the sample collector and will contain a simple trajectory that the sample collector has to approximately follow, with height indications to avoid obstacles, until the sampling point is reached.

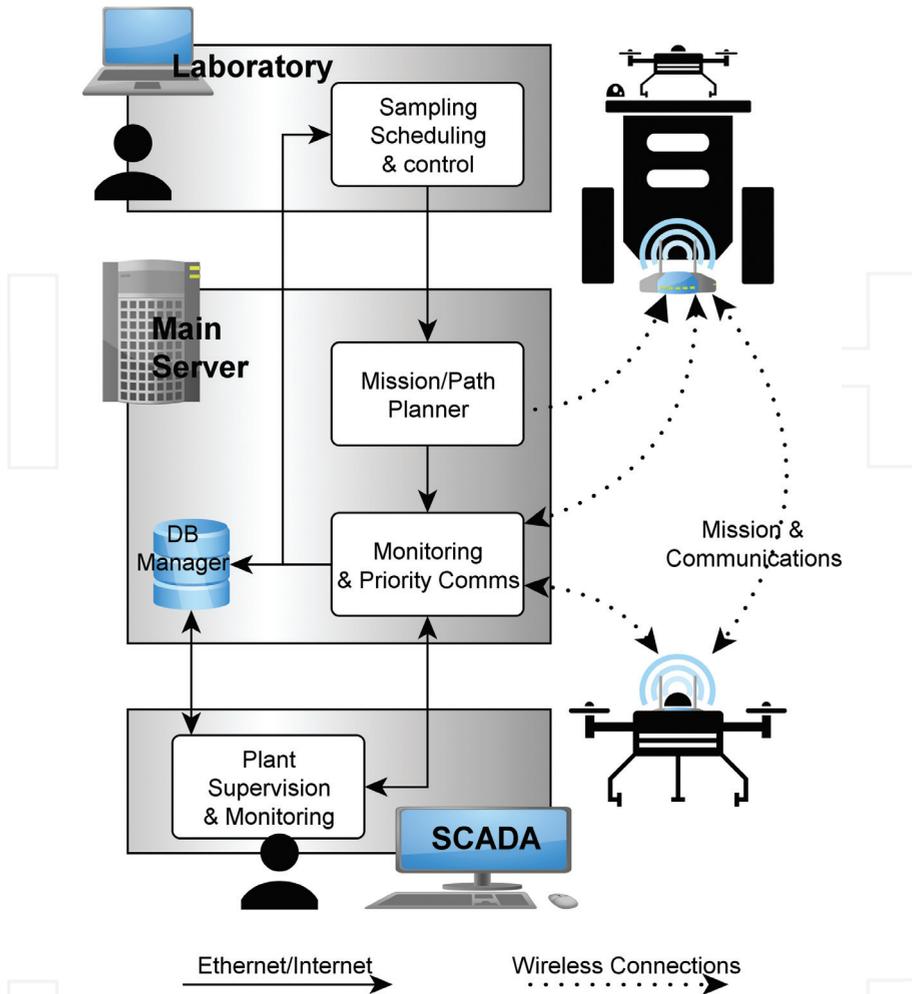


Figure 1. Proposed system distribution and communications diagram.

The different paths and trajectories are generated by the path planner module, which works on a two-dimensional (2D) grid map model of the outdoors. When data are available, the grid is to be textured with the map obtained from the visual SLAM technique proposed. The utilization of the grid model allows introducing additional data, like occupancy, possible obstacles, or even scheduling the accessibility of an area, e.g., we can set a path normally used by workers during certain hours to be avoided at those times. An energy minimization planning technique, essentially a simplified approach to Ref. [3], is used to obtain the trajectories, considering a set of criteria.

1. All samples in the same basin/tank should be capture consecutively;
2. Minimizing the expected flight effort for the UAV (heuristically assuming that the trajectory is a polyline of vertical and horizontal movements);

3. Minimize penalties for restricted access areas;
4. Minimize the trajectory of the UGV.

The communications with the autonomous platforms of the system will be performed through 4G in order to allow video streaming during the prototyping and testing phases. The data produced by the different elements of the system will be stored and logged to allow posterior analysis. The rest of the communications can be performed through any usual channel in industrial environments, be it a local network, Internet, VPN..., as the segmented architecture allows discrete deployment. The communications between the sample carrier and the sample collector will be performed through ZigBee, although they present Wi-Fi modules to ease development and maintenance tasks.

A subset of the communication protocols is considered priority communications. This includes the supervision and surveying messages in normal planned operation, and those signals and procedures that can affect or override normal operation, e.g., emergency recall or landing. For the UAV, the recall protocol uses a prioritized list of possible fallback points so that the UAV tries to reach them. If the sample carrier is present, the UAV will try to land on it, searching for its landing area through fiduciary markers. If it is not possible, the UAV will try another fallback point in the list, until it lands or the battery is below a certain threshold: in that case, it will just land in the first clear patch of ground available, though this will generally require dropping the sampling device.

2.2. Sample collector aerial robot architecture

The UAV built as a sample collector, UAV is a 0.96 m diameter quadcopter deploying four 16" propellers with T-Motor MN4014 actuators. The custom-built frame supports the propeller blocks at 5° angle and is made of aluminum and carbon fiber. A PIXHAWK kit is used as a flight management unit (FMU), with custom electronics to support 2 6 s 8000 mAh batteries. An Odroid U4 single-board-computer is used to perform the high level task and deal with all noncritical processes. Beyond the sensors present in the FMU (which includes AHRS and GPS), the UAV presents a front facing camera USB (640×480@30fps) for monitoring purposes, an optical flow with ultrasound sensor facing downward, and a set of four ultrasound sensors deployed in a planar configuration to detect obstacles collision. The FX4 stack [4] is used to manage flying and navigation, while a Robot Operating System (ROS) [5] distribution for ARM architectures is run at the Odroid Single Board Computer (SBC), supporting MavLink for communications. The communication modules and sensors are described at hardware level in **Figure 2**.

The hardware used weights approximately 4250 kg, while the propellers provide a theoretical maximum lift of 13,900 g. The sample collection device, including the container, is being designed with a weight below of 300 g, meaning that UAV, with the sample capture system and up to 1000 ml of water-like fluid, would keep the weight/lift ratio around 0.4.

A simplified diagram of the operation process is shown in **Figure 3**. The communications are divided into two blocks, those connecting the UAV with the main network for supervision and emergency control, including video streaming with MavLink [6] over 4G, and those that will connect it to the sample carrier during routine operation. This routine operation includes receiving

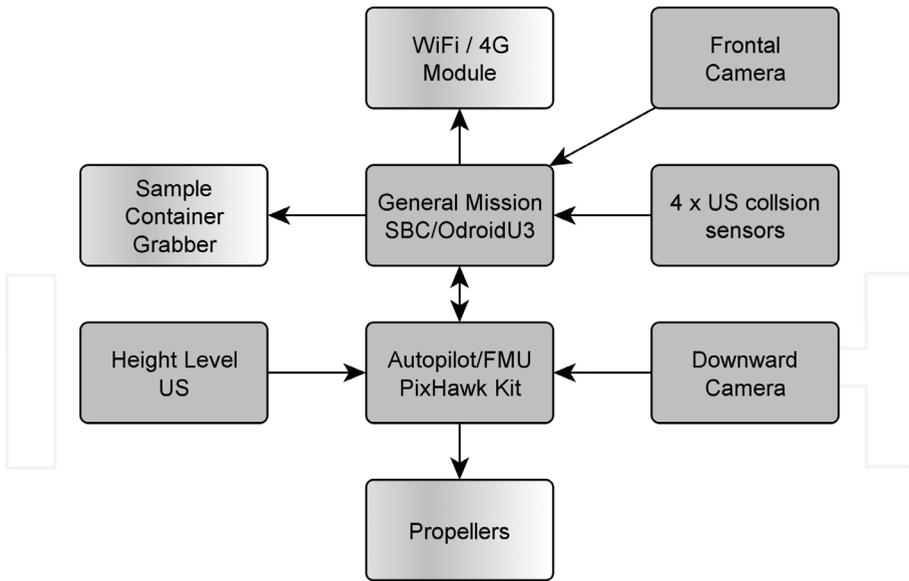


Figure 2. Sample collector aerial robot hardware and communications diagram.

the collection mission detailing the trajectory to the sampling point and the signal to start the process. The action planner module in the UAV supervises the navigation tasks, making sure that the trajectory waypoints are reached and the managing the water sample collector control module.

The localization and positioning of the UAV are solved through a combination of GPS and visual odometry estimation (see Section 4). To approach the water surface, the downward looking ultrasonic height sensor will be used, as vision-based approaches are unreliable on reflective surfaces. To ease the landing problem, the sample carrier will present fiduciary markers to estimate the pose of the drone. This allows landing operations while inserting the sample container into a socket, similarly to an assisted peg-in-the-hole operation. After the UAV is landed, the sample collector control module releases the sample container, and the sample carrier will replace the container with a clean one. After the container is replaced, the carrier will send a signal to the UAV so that the new container is properly locked, while the filled one will be stored.

As the autonomous navigation depends mainly on the localization and positioning, it is largely based in GPS and visual odometry. This combination allows, combined with the ultrasound to find height and avoid possible obstacle, to navigate the environment event if the GPS signal is unreliable. Though there are more accurate alternatives to solve the SLAM problem in terms of sensors, namely using RGB-D cameras or Lighting Detection and Ranging (LiDAR)-based approaches, they present several limitations that make them unsuitable for outdoor industrial environments. These limitations are in addition to the penalty imposed by their economical cost, especially for models industry levels of performance and reliability.

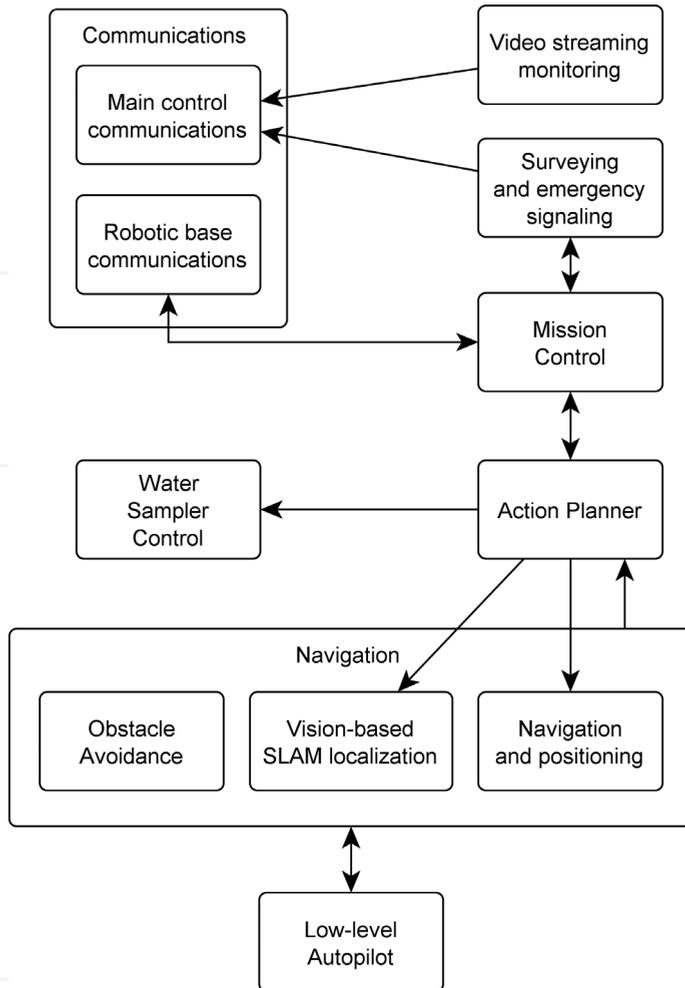


Figure 3. Sample collector UAV robot block diagram.

In the case of RGB-D, most of the sensors found in the market are unreliable in outdoors as they use IR or similar lightning frequencies. The subset of time-of-flight RGB-D sensors presents the same limitation as the LiDAR sensors: they are prone to spurious measurements in environments where the air is not clear (presence of dusty, pollen and particles, vapors from tanks), they present large latencies, making them unfit for real-time operation of UAVs, and they are generally considered not robust enough for industrial operation. In the case of LiDAR, and RGB-D if the SLAM/localization technique used focuses on depth measurements, there is an additional issue present: these approaches normally rely in computationally demanding optimization techniques to achieve accurate results, and the computer power available in an UAV is limited.

3. UAV localization and visual odometry estimation

The drone platform is considered to freely move in any direction in $R^3 \times SO(3)R^3 \times SO(3)$, as shown in **Figure 4**.

As the proposed system is mainly intended for local autonomous vehicle navigation, i.e., localize the sample collector during the different collection missions, the local tangent frame is used as the navigation reference frame. The initial position of the sample collector landed over the sample carrier is used to define the origin of the navigation coordinates frame, and the axes are oriented following the navigation convention NED (North, East, Down). The magnitudes expressed in the navigation, sample collector drone (robot), and camera frame are denoted, respectively, by the superscripts NN, RR, and CC. All the coordinate systems are right-handed defined. The proposed method will be taken mainly into account the AHRS and the downward monocular camera, though it also uses data from the GPS during an initialization step.

The monocular camera is assumed to follow the central-projection camera model, with the image plane in front of the origin, thus forming a noninverted image. The camera frame C is considered right handed with the z-axis pointing toward the field of view. It is considered that

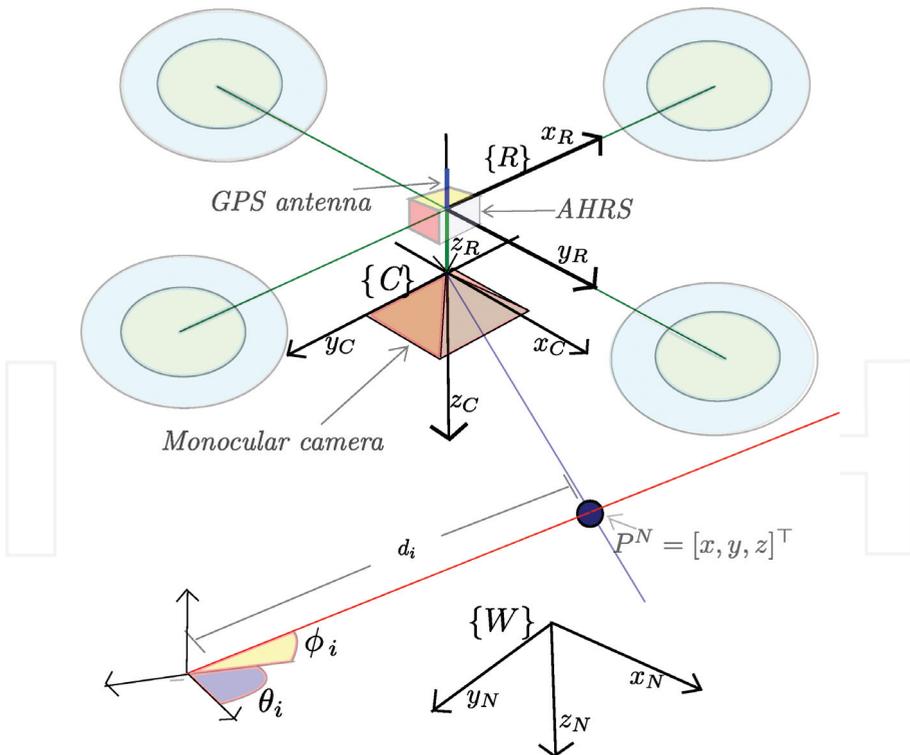


Figure 4. Coordinate systems: the local tangent frame is used as the navigation reference frame NN. AHRS: attitude and heading reference system.

the pixel coordinates are denoted with the $[u, v]$ pair convention and follow the classical direct and inverse observation models [7].

The attitude and heading reference system (AHRS) is a device used for estimating the vehicle orientation, while it is maneuvering. The most common sensors integrated with AHRS devices are gyroscopes, accelerometers, and magnetometers. The advances in micro-electro-mechanical systems (MEMS) and microcontrollers have contributed to the development of inexpensive and robust AHRS devices (e.g., [8–10]).

In the case of the deployed FMU, the accuracy and reliability provided by its AHRS are enough to directly fuse its data into the estimation system. Thus, AHRS measurements are assumed to be available at high rates (50–200 Hz) and modeled according to

$$y_a^N = a^N + v_a \tag{1}$$

where $a^N = [\varphi_v, \theta_v, \psi_v]^T$, being φ_v , θ_v , and ψ_v Euler angles denoting, respectively, the roll, pitch, and yaw of the vehicle; with v_a being Gaussian white noise.

The global positioning system (GPS) is a satellite-based navigation system that provides 3D position information for objects on or near the Earth’s surface, studied in several works [11, 12]. The user-equivalent range error (UERE) is a measurement of the cumulative error in GPS position measurements caused by multiple sources of error. These error sources can be modeled as a combination of random noise and slowly varying biases [11]. According to a study [13], the UERE is around 4.0 m (σ); in this case, 0.4 m (σ) corresponds to random noise.

In this work, it is assumed that position measurements y_r can be obtained from the GPS unit, at least at the beginning of the trajectory, and they are modeled by

$$y_r = r^N + v_r \tag{2}$$

where v_r is Gaussian white noise and r^N is the position of the vehicle. As GPS measurements are usually in geodetic coordinates, Eq. (2) assumes that they have been converted to the corresponding local tangent frame for navigation, accounting for the transformation between the robot collector frame and the antenna.

3.1. Problem formulation

The objective of the estimation method is to compute the system state x

$$x = [x_v, y_1^N, y_2^N, \dots, y_n^N]^T \tag{3}$$

where the system state x can be divided into two parts: one defined by x_v and corresponding to the sample collector UAV state, and the other one corresponding the map features. In this case, y_i^N defines the position of the i th feature map. The UAV state x_v is composed, at the same time, of

$$x_v = [q^{NR}, \omega^R, r^N, v^N]^T \tag{4}$$

where q^{NR} represents the orientation of the vehicle with respect to the local world (navigation) frame by a unit quaternion, and $\omega^R = [\omega_x, \omega_y, \omega_z]$ represents the angular velocity of the UAV

expressed in the same frame of reference. $r^N = [p_x, p_y, p_z]$ represents the position of the UAV expressed in the same navigation frame, with $v^N = [v_x, v_y, v_z]$ denoting the linear velocities. The location of a feature y_i^N (noted as y_i for simplicity) is parameterized in its Euclidean form:

$$y_i^N = [p_{x_i}, p_{y_i}, p_{z_i}]^T \tag{5}$$

The proposed system follows the classical loop of prediction-update steps in the extended Kalman filter (EKF) in its direct configuration; working at the frequency of the AHRS. Thus, both the vehicle state and the feature estimations are propagated by the filter, see **Figure 5**.

At the start of a prediction-update loop, the collector UAV state estimation x_v takes a step forward through the following unconstrained constant-acceleration (discrete) model:

$$\begin{aligned} q_{k+1}^{NC} &= \left(\cos \|W\| I_{4 \times 4} + \frac{\sin \|W\|}{\|W\|} W \right) q_k^{NC} \\ \omega_{k+1}^C &= \omega_k^C + \Omega^C \\ r_{k+1}^N &= r_k^N + v_k^N \Delta t \\ v_{k+1}^N &= v_k^N + V^N \end{aligned} \tag{6}$$

In the model represented by Eq. (6), a closed form solution of $\dot{q} = 1/2(W)q$ is used to integrate the current velocity rotation ω^C over the quaternion q^{NC} .

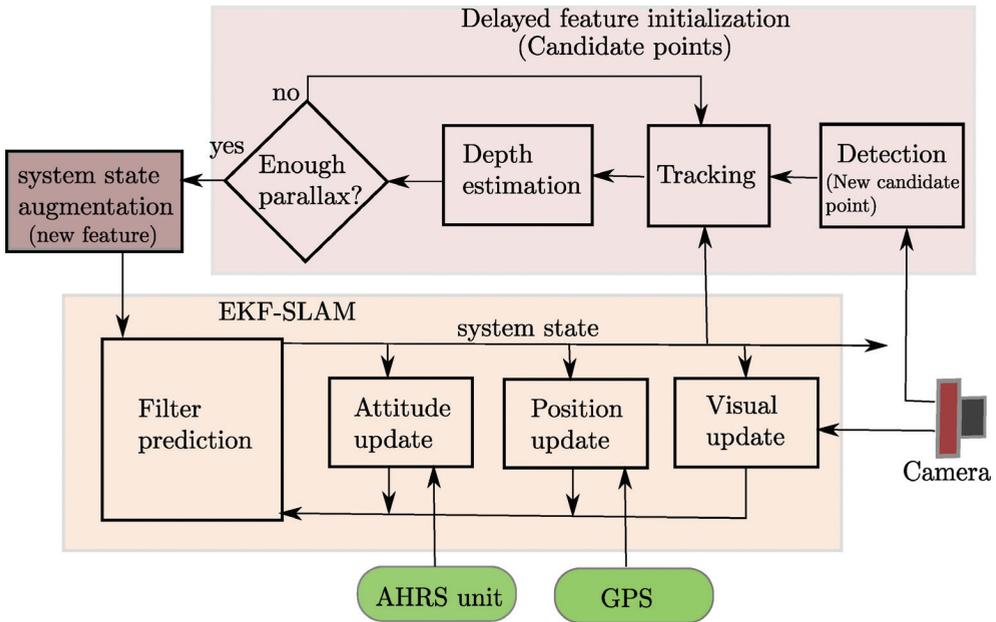


Figure 5. Block diagram showing the architecture of the system. EKF-SLAM: extended Kalman filter simultaneous localization and mapping.

At every step, it is assumed that there is an unknown linear and angular velocity with acceleration zero-mean and known-covariance Gaussian processes σ_v and σ_{ω} , producing an impulse of linear and angular velocity: $V^N = \sigma_v^2 \Delta t$ and $\Omega^C = \sigma_{\omega}^2 \Delta t$. It is assumed that the map features y_i remain static (rigid scene assumption). Then, the state covariance matrix P takes a step forward by

$$P_{k+1} = \nabla F_x P_k \nabla F_x^T + \nabla F_u Q \nabla F_u^T \tag{7}$$

where Q is the noise covariance matrix of the process (a diagonal matrix with the position and orientation uncertainties); ∇F_x is the Jacobian of the nonlinear prediction model (Eq. (7)); and ∇F_u is the Jacobian of the concerning the derivatives of the input process represented by the unknown linear and angular velocity impulses assumed.

3.2. Visual features: detection, tracking, and initialization

In order to retrieve the depth of a visual feature, the monocular camera, equipped in the UAV, must observe it at least two times while moves along the flight trajectory. The parallax angle is defined by the two projections of those measurements. The convergence of the depth of the features depends on the evolution of the parallax angle. In this work, a method that is based on a stochastic technique of triangulation is used for computing an initial hypothesis of depth for each new feature prior to its initialization into the map. The initialization method is based on previous author’s work [7].

3.2.1. Detection stage

The visual-based navigation method requires a minimum number of visual features y_i observed at each frame. If this number of visual features is lower than a threshold, then new visual features are initialized into the stochastic map. The Shi-Tomasi corner detector [14] is used for detecting new visual points in the image that will be taken as candidates to be initialized into the map as new features.

When a feature is detected for the first time at k frame, a candidate feature c_i is stored:

$$c_1 = \left[\begin{matrix} (t_{c_0}^N)^T \\ \theta_0, \phi_0, P_{y_{c_i}}, z_{uv} \end{matrix} \right] \tag{8}$$

where $z_{uv} = [u, v]$ is the location of the visual feature in pixel coordinates in image frame k ; $y_{c_i} = [t_{c_0}^N, \theta_0, \phi_0]^T$, complying the inverse observation model $y_{c_i} = h(\hat{x}, z_{uv})$. Thus, y_{c_i} models a 3D ray originating in the optical center of the camera when the feature is first observed, and pointing to the infinite, with azimuth and elevation θ_0 and ϕ_0 , respectively, according to previous work [7]. At the same time, $P_{y_{c_i}}$ stores, as a 5×5 covariance matrix, the uncertainties of y_{c_i} . $P_{y_{c_i}}$ are computed as $P_{y_{c_i}} = \nabla Y_{c_i} P \nabla Y_{c_i}^T$, where P is the system covariance and is the Jacobian of the observation model $h(\hat{x}, z_{uv})$ with respect to the state and coordinates u, v . A square patch around $[u, v]$, generally with 11 pixels sides, is also stored, to keep the appearance of the landmark.

3.2.2. Tracking of candidate features

The active search visual technique [15] can be used for tracking visual features that form part of the system state. In this case, the information included in the system state and the system

covariance matrix is used for defining images regions where the visual feature is searched. On the other hand, in the case of the candidate features, there is not enough information for applying the active search visual technique. A possibility for tracking the candidate features is to use of general-purpose visual tracking approach [14]. This kind of methods uses only the visual input and do not need information about the system dynamics; however, their computational cost is commonly high. In order to improve the computational performance, a different technique is proposed. The idea is to define very thin elliptical regions of search in the image that are computed based on the epipolar constraints.

After the k frame when the candidate feature was first detected, this one is tracked at subsequent frames $k + 1, k + 2 \dots k + n$. In this case, the candidate feature is predicted to lie inside the elliptical region S_c (see Figure 6). The elliptical region S_c is centered in the point $[u, v]$, and it is aligned (the major axis) with the epipolar line defined by image points e_1 and e_2 .

The epipole is computed by projecting the line t_{c0}^N stored in c_l to the current image plane using the camera projective model. As there is not any available depth information, the semiline stored in c_l is considered to have origin t_{c0}^N , and a length $d = 1$, producing point e_1 and e_2 .

The orientation of the ellipse S_c is determined by $\alpha_c = \text{atan2}(e_y, e_x)$, where e_y and e_x represent the y and x coordinates, respectively, of the epipole e , and $e = e_2 - e_1$. The size of the ellipse S_c is determined by its major and minor axis, respectively, a and b . This ellipse S_c represents a probability region where the candidate point must lie in the current frame. The proposed tracking method is intended to be used during an initial short period of time, applying cross-correlation operators. During this period, some information will be gathered in order to compute a depth hypothesis for each candidate point, prior to its initialization as a new map feature.

On the other hand, during this stage, the available information of depth about the candidate features is not well-conditioned. For this reason, it is not easy to determine an adaptive and optimal size of the search image region. Therefore, according to the kind of application, the

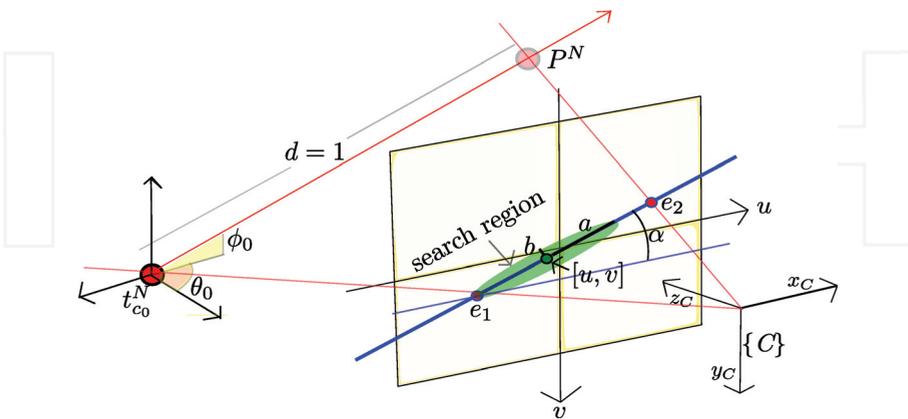


Figure 6. The established search region to match candidate points is constrained to ellipses aligned with the epipolar line.

parameter a is chosen empirically. In the experiments presented in this work, good results were obtained choosing $a = 20$ pixels.

3.2.3. Depth estimation

Each time a new image location $z_{uv} = [u, v]$ is obtained for a given candidate c_l , a depth hypothesis is computed, through stochastic triangulation, as described in previous work [16]. In previous authors' work [17], it was found that the estimates of the feature depth can be improved by passing the hypotheses of depth by a low-pass filter. Thus, only when parallax α_i is greater than a specific threshold ($\alpha_i > \alpha_{min}$), a new feature $y_{new} = [p_{xir} \ p_{yir} \ p_{zri}]^T = h(c_l, d)$ is added to the system state vector x , as Eq. (3).

In order to correctly operate the proposed method, a minimum number of map features must be maintained inside the camera field of view. For example in Ref. [21], it is stated that a minimum number of three features are required for the operation of monocular SLAM. In practice, of course, it is better to operate with a higher minimum number of features. The above requires to initialize continuously new features into the map.

The initialization time period that takes a candidate point to become a map feature depends directly on the evolution of the parallax angle. At the same time, the evolution of the parallax depends on the depth of the feature and the movement of the camera. For example, the parallax for a near point can increase very quickly due to a small movement of the camera. In practice, it is assumed that the dynamics of the aerial vehicle allows tracking a sufficient number of visual features for initialization and measurement purposes. Experimentally, at least for nonaggressive typical flight maneuvers, we have not found yet major problems in order to accomplish the above requirement.

3.3. Prediction-update loop and map management

Once initialized, the visual features y_i are tracked by means of an active search technique, using zero-mean normalized cross-correlation to match a given feature, and the patch that notes its appearance to a given pixel to be found in a search area. This search area is defined using the innovation covariance. The general methodology for the visual update step can be found in previous works [7, 16, 18], where the details in terms of mathematical representation and implementation are discussed. In this work, the loop closing problem and the application of SLAM to large-scale environments are not addressed. Although it is important to note that SLAM methods that perform well locally can be extended to large-scale problems using different approaches, such as global mapping [19] or global optimization [20].

On the other side, when an attitude measurement y_a^N is available, the system state is updated. Since most low-cost AHRS devices provide their output in Euler angles format, the following measurement prediction model $h_a = h(\hat{x}_v)$ is used:

$$\begin{bmatrix} \theta_v \\ \phi_v \\ \psi_v \end{bmatrix} = \begin{bmatrix} a \tan 2(q_3q_4 - q_1q_2), 1 - 2(q_2^2 - q_3^2) \\ a \sin (-2(q_1q_3 - q_2q_4)) \\ a \tan 2(q_3q_4 - q_1q_2), 1 - 2(q_3^2 - q_4^2) \end{bmatrix} \quad (9)$$

During the initialization period, position measurements y_r are incorporated into the system using the simple measurement model $h_r = h(\hat{x}_v)$:

$$h_r = [p_x, p_y, p_z]^T \quad (10)$$

The regular Kalman update equations [7, 16, 18] are used to update attitude and position whenever is required, but using the corresponding Jacobian ∇H and measurement noise covariance matrix R related to the AHRS observation model.

In this work, the GPS signal is used for incorporating metric information into the system in order to retrieve the metric scale of the estimates. For this reason, it is assumed that the GPS signal is available at least during some period at the beginning of the operation of the system. In order to obtain a proper operation of the system, this period of GPS availability should be enough for allowing the convergence of the depth for some initial features. After this initialization period, the system is capable of operating using only visual information.

In Ref. [21], it is shown that only three landmarks are required for setting the metric scale in estimates. Nevertheless, in practice, there is often a chance that a visual feature is lost during the tracking process. In this case, it is convenient to choose a threshold $n \geq 3$ of features that present convergence in order to end the initialization process. In this work, good experimental results were obtained with $n = 5$.

An approach for testing features convergence is the Kullback distance [22]. Nevertheless, the computational cost of this test is quite high. For this reason, the following criterion is proposed for this purpose:

$$\max(\text{eig}(P_{y_i})) < \frac{\|y_i - r^N\|}{100} \quad (11)$$

where P_{y_i} is the covariance matrix of the y_i feature, and it is extracted from the system covariance matrix P . If the larger eigenvalue of P_{y_i} is smaller than a centime of the distance between the feature and the camera, then it is assumed that the uncertainty of the visual features is small enough to consider it as an initial landmark.

4. Results

In this section, the results related to the proposed visual odometry approach for UAV localization in industrial environments are discussed. The results obtained using synthetic data from simulations as well as that obtained from experiments with real data are presented. The experiments were performed in order to validate the performance, accuracy, and viability of the proposed localization method, considering a real outdoor scenario. Previous works by the authors [7, 18] and other researchers [2] have proven that similar solutions can reach real-time performance, being of more interest development of robust solutions that can be optimized further down the line. The proposed method was implemented in MATLAB©.

4.1. Experiments with simulations

The values of the parameters used to simulating the proposed method were taken from the following sources: (i) the parameters of the AHRS were taken from Ref. [8], (ii) the model used for emulating the GPS error behavior was taken from Ref. [23], and (iii) the parameters of the monocular camera are the same for real camera used in real-data experimentation.

In order to validate the performance of the proposal, two different scenarios were simulated (see **Figure 7**) as follows:

1. The environment of the aerial robot is composed of landmarks uniformly distributed over the ground. The quadrotor performs a circular flight trajectory, at a constant altitude, after the taking off.
2. The environment of the aerial robot is composed of landmarks randomly distributed over the ground. The quadrotor performs and eight-like flight trajectory, at a constant altitude, after the taking off.

In simulations, the data association problem is not considered, that is, it is assumed that visual features can be detected and tracked without errors. Also, it is assumed that the aerial robot can be controlled perfectly.

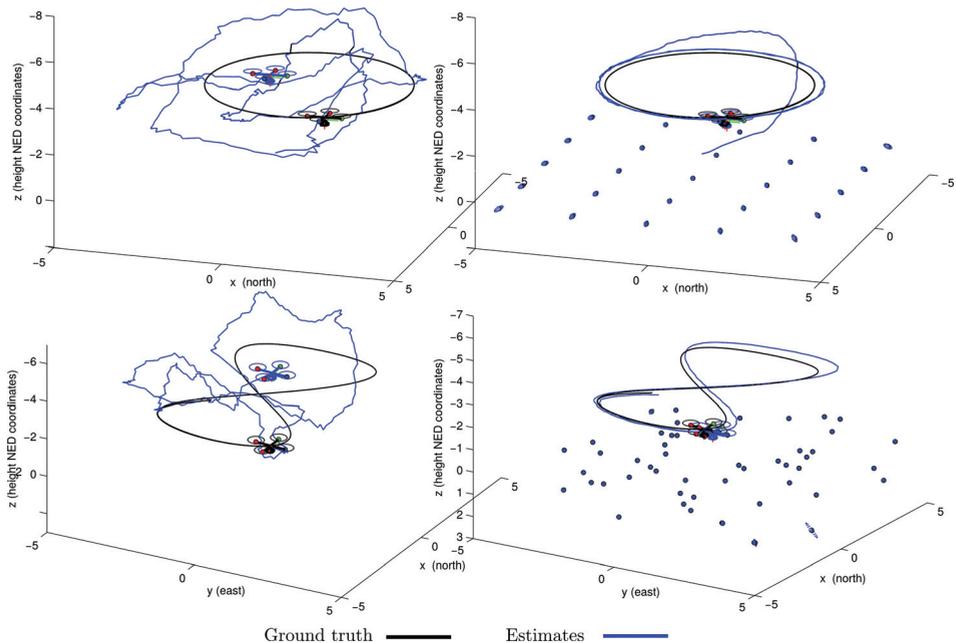


Figure 7. Estimation of two flight trajectories obtained with the filtered GPS data (left plots), and with the scheme proposed of visual-based navigation (right plots).

In order to obtain a statistical interpretation of the simulations results, the MAE (mean absolute error) was computed for 20 Monte Carlo runs. The MAE was calculated for the following cases:

1. Trajectory estimated using only filtered data from the GPS.
2. Trajectory estimated using visual information in combination with GPS data, during all the simulation.
3. Trajectory estimated using visual information and GPS data, but only during the initialization period.

In the results presented in **Figure 8**, it can be appreciated the typical SLAM behavior, every time when the aerial robot flight near to its initial position, when the visual-based scheme is used. In this case, note that when the initial visual features are recognized again, the error is minimized. On the other hand, in the case where GPS data is fused into the system during the whole trajectory, it can be appreciated an influence of the GPS error when the aerial robot flight near to its initial position. In this latter case, the error is less minimized. The above results can suggest that for some scenarios, it is better to navigate relying only on visual information.

Also, it is important to note that for trajectories where the aerial robot moves far away from the initial position, the use of GPS data can be very useful because an upper bound is imposed to the error drift which is inherent to the navigation scheme based only on vision.

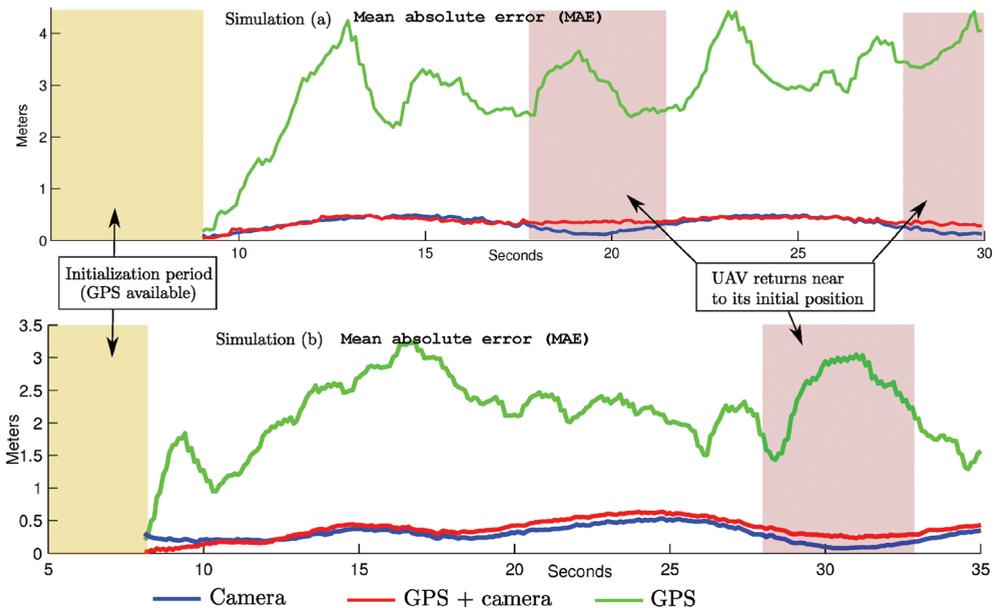


Figure 8. Mean absolute error (MAE) in position computed from two simulations (a and b) out of 20 Monte Carlo runs: (upper plot) simulation (a) results; (lower plot) simulation (b) results.

It is important to note that errors related to the slow-time varying bias of the GPS can be modeled in Eq. (2) by considering a bigger measurement noise covariance matrix. However, in experiments, it was found that if this matrix is increased too much then the convergence of initial visual features can be affected. Future work could include an adaptive approach for fusing GPS data, or for instance, to include the bias of the GPS into the system state in order to be estimated.

4.2. Experiments with real data

The custom build sample collector UAV was used to perform experiments with real data. In experiments, the quadrotor has been manually radio-controlled to capture data. The data captured from the GPS, AHRS, and frames from the camera were synchronized and stored in an ROS data set. The frames with a resolution of 320×240 pixels, in gray scale, were captured at 26 fps. The flights of the quadrotor were conducted in industry-like facilities.

The surface of the field is mainly flat and composed of asphalt, grass, and dirt, but the experimental environment also included some small structures and other manmade elements. In average eight to nine GPS, satellites were visible at the same time.

In order to evaluate the estimates obtained with the proposed method, the flight trajectory of the quadrotor was determined using an independent approach. In this case, the position of the camera was computed, at each frame, with respect to a known reference composed of four marks placed on the floor forming a square of known dimensions. The perspective on four-point (P4P) technique described in [24] was used for this purpose.

As it was explained earlier, an initialization period was used for incorporating GPS readings in order to set the metric scale of the estimates. After the initialization period, the estimation of the trajectory of the aerial robot was carried out using only visual information.

The same methodology used with simulated data was employed with real data. Therefore, the same experimental variants were tested: (i) GPS, (ii) GPS + camera, (iii) camera (GPS only during the initialization period). All the results were obtained averaging 10 experimental outcomes. **Figure 9** shows the evolution of the estimated flight trajectory along the time, which was obtained with each experimental variant. **Table 1** shows a summary of the above experimental results. In this case, in order to compute the error in position, the trajectory computed with the P4P technique was used as ground truth.

It worthwhile to note that analyzing the above results, it can be verified similar conclusions that they were obtained with simulations: The use of GPS exclusively can be unreliable for determining the vehicle position in the case of fine maneuvers. In flight trajectories near to the initial reference, the error can be slightly low relying only on visual information.

Regarding the application of the proposed method in a real-time context, the number of features that are maintained in the system state is considerably below an upper bound that should allow a real-time performance, for instance by implementing the algorithm in C or C++ languages. In particular, since early works in monocular SLAM as Davison [25], it was

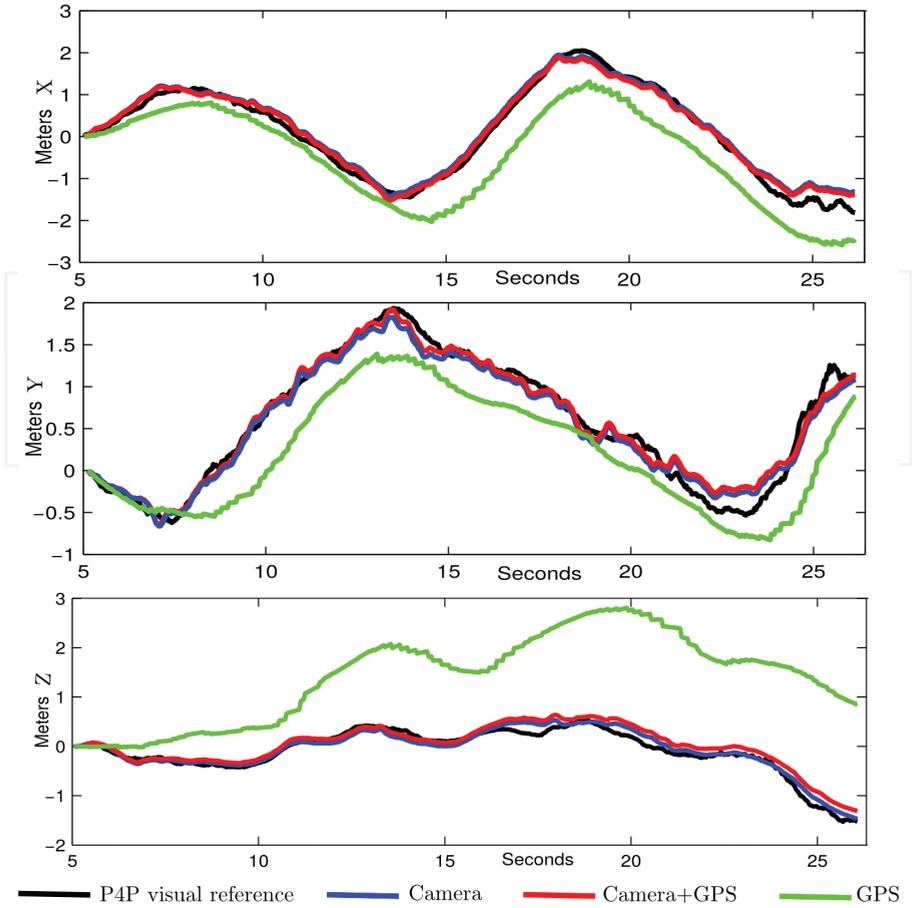


Figure 9. Flight trajectory estimated with: (i) P4P visual reference, (ii) camera, (iii) camera + GPS, and (iv) GPS. The position is expressed in north coordinates (upper plot), east coordinates (middle plot), and altitude coordinates (lower plot). In every case, an initialization period of 5 s was considered with GPS availability.

Method	NOF	aMAE
GPS	–	$1.70 \pm 0.77\sigma$
Camera + GPS	$56.4 \pm 10.2\sigma$	$0.21 \pm 0.11\sigma$
Camera	$57.9 \pm 9.3\sigma$	$0.20 \pm 0.09\sigma$

NOF stands for the average number of visual features contained within the stochastic map. The aMAE is the average mean absolute error in position (in meters).

Table 1. Summary of results with real data.

shown the feasibility of real-time operation for EKF-based methods for maps composed of up to 100 features.

5. Conclusions

An industrial system to automatize the water sampling processes on outdoor basins in a wastewater treatment plant has been proposed, with novel research to solve the localization for navigation problem under unreliable GPS. The architecture of the whole system has been described, while specifications at the hardware level have been presented for those elements designed completely, including the sample collector UAV, and the proposed localization technique has been described and validated with experiments performed over simulated and real data. The localization technique presented can be described as a vision-based navigation and mapping system applied to UAV.

The proposed estimation scheme is similar to a pure monocular SLAM system, where a single camera is used for estimating concurrently both the position of the camera and a map of visual features. In this case, a monocular camera equipped in the aerial robot, which is pointing to the ground, is used as the main sensory source. On the other hand, in the proposed scheme additional sensors, which are commonly available in this kind of robotic platforms, are used for solving the typical technical difficulties which are related to purely monocular systems.

One of the most important challenges, regarding the use of monocular vision in SLAM, is the difficulty in estimating the depth information of visual features. In this case, a method based on a stochastic technique of triangulation is proposed for this purpose.

Perhaps the other most relevant challenge, regarding the use of monocular vision, has to do with the difficulty of retrieving the metric scale of estimates. In this work, it is assumed that the GPS signal is available during an initial period of time which is used for setting the metric scale of the estimates. After the initialization period, the proposed system is able to estimate the position of the flying vehicle using only visual information.

For some scenarios, it was seen that the exclusive use of filtered GPS data can be unreliable for performing fine maneuvers. This is due to the noise implicit in GPS measurements. In this context, the following conclusions were found based on the experiments with real data as well as with simulations:

1. Even the use of very noisy GPS data, during an initial short period of time, can be enough for setting the metric scale of the estimates obtained by a monocular SLAM system.
2. The integration of GPS measurements can be avoided for flight trajectories near to the origin of the navigation frame.

Acknowledgements

This research has been funded by EU Project AEROARMS project reference H2020-ICT-2014-1-644271, <http://www.aeroarms-project.eu/>.

Author details

Rodrigo Munguía¹, Edmundo Guerra², Sarquis Urzua¹, Yolanda Bolea² and Antoni Grau^{2*}

*Address all correspondence to: antoni.grau@upc.edu

1 Department of Computer Science, CUCEL, University of Guadalajara, Guadalajara, Mexico

2 Automatic Control Department, Technical University of Catalonia, Barcelona, Spain

References

- [1] Ping JTK, Ling AE, Quan TJ, Dat CY. Generic unmanned aerial vehicle (UAV) for civilian application—A feasibility assessment and market survey on civilian application for aerial imaging. In: 2012 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (STUDENT). 2012. pp. 289-294
- [2] Davison AJ, Reid ID, Molton ND, Stasse O. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2007 Jun;29(6):1052-1067
- [3] Ganganath N, Cheng CT, Tse CK. A constraint-aware heuristic path planner for finding energy-efficient paths on uneven terrains. *IEEE Transactions on Industrial Informatics*. 2015 Jun;11(3):601-611
- [4] Meier L, Honegger D, Pollefeys M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). 2015. pp. 6235-6240
- [5] Quigley M, Conley K, Gerkey B, Faust J, Foote T, Leibs J, et al. ROS: An open-source robot operating system. In: ICRA Workshop on Open Source Software. 2009. p. 5
- [6] MAVLink Micro Air Vehicle Communication Protocol—QGroundControl GCS [Internet]. Available from: <http://qgroundcontrol.org/mavlink/start> [Accessed: January 26, 2017]
- [7] Munguía R, Grau A. Monocular SLAM for visual odometry: A full approach to the delayed inverse-depth feature initialization method. *Mathematical Problems in Engineering*. pp. 1-26; 2012
- [8] Munguía R, Grau A. A practical method for implementing an attitude and heading reference system. *International Journal of Advanced Robotics Systems*. 2014 Apr 29;11(4):62
- [9] Wang M, Yang Y, Hatch RR, Zhang Y. Adaptive filter for a miniature MEMS based attitude and heading reference system. In: PLANS 2004 Position Location and Navigation Symposium (IEEE Cat No04CH37556). 2004. pp. 193-200
- [10] Jurman D, Jankovec M, Kamnik R, Topič M. Calibration and data fusion solution for the miniature attitude and heading reference system. *Sensors and Actuators A: Physical*. 2007 Aug 26;138(2):411-420

- [11] Spilker Jr. JJ, Axelrad P, Parkinson BW, Enge P. Global Positioning System: Theory and Applications, Volume I [Internet]. American Institute of Aeronautics and Astronautics; 1996. 1. (Progress in Astronautics and Aeronautics). Available from: <http://arc-test.aiaa.org/doi/book/10.2514/4.866388> [Accessed: January 27, 2017]
- [12] Grewal MS, Weill LR, Andrews AP. Global positioning systems, inertial navigation, and integration. New York: John Wiley; 2001. p. 392
- [13] GPS: Essentials of Satellite Navigation: Compendium : Theories and Principles of Satellite Navigation, Overview of GPS/GNSS Systems and Applications. U-Blox; 2009. p. 176
- [14] Shi J, Tomasi C. Good Features to Track. 1994. pp. 593-600
- [15] Davison AJ, Murray DW. Mobile robot localisation using active vision [Internet]. Springer; 1998. Available from: <http://link.springer.com/chapter/10.1007/BFb0054781> [Accessed: September 30, 2015]
- [16] Munguia R, Grau A. Monocular SLAM for visual odometry. In: IEEE International Symposium on Intelligent Signal Processing, 2007 WISP 2007. 2007. pp. 1-6
- [17] Munguía R, Grau A. Concurrent Initialization for bearing-only SLAM. *Sensors*. 2010 Mar 1;10(3):1511-1534
- [18] Guerra E, Munguia R, Bolea Y, Grau A. New validation algorithm for data association in SLAM. *ISA Transactions*. 2013 May 20;52:662-671
- [19] Munguia R, Grau A. Closing loops with a virtual sensor based on monocular SLAM. *IEEE Transactions on Instrumentation and Measurement*. 2009;58(8):2377-2384
- [20] Strasdat H, Montiel JMM, Davison AJ. Real-time monocular SLAM: Why filter? In: 2010 IEEE International Conference on Robotics and Automation (ICRA). 2010. pp. 2657-2664
- [21] Belo FAW, Salaris P, Fontanelli D, Bicchi A. A complete observability analysis of the planar bearing localization and mapping for visual servoing with known camera velocities. *International Journal of Advanced Robotic Systems*. 2013 Apr 17;10(4):197
- [22] Bailey T. Constrained initialisation for bearing-only SLAM. In: 2003 Proceedings ICRA'03 IEEE International Conference on Robotics and Automation [Internet]. 2003. pp. 1966-1971. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1241882 [Accessed: June 4, 2013]
- [23] Rankin J. An error model for sensor simulation GPS and differential GPS. In: IEEE Position Location and Navigation Symposium, 1994. 1994. pp. 260-266
- [24] Chatterjee C, Roychowdhury VP. Algorithms for coplanar camera calibration. *Machine Vision and Applications*. 2000;12(2):84-97
- [25] Davison AJ. Real-time simultaneous localisation and mapping with a single camera. In: IEEE International Conference on Computer Vision. 2003. pp. 1403-1410