RESEARCH ARTICLE

Delayed Monocular SLAM Approach Applied to Unmanned Aerial Vehicles

Rodrigo Munguia[1☯]*, **Sarquis Urzua**[1☯], **Antoni Grau**[2☯]*

1 Department of Computer Science, CUCEI, University of Guadalajara, Guadalajara, México, **2** Automatic Control Dept, Technical University of Catalonia, 08034 Barcelona, Spain

☯ These authors contributed equally to this work.
* rodrigo.munguia@upc.edu (RM); antoni.grau@upc.edu (AG)

Citation: Munguia R, Urzua S, Grau A (2016) Delayed Monocular SLAM Approach Applied to Unmanned Aerial Vehicles. PLoS ONE 11(12): e0167197. doi:10.1371/journal.pone.0167197

Editor: Jun Xu, Beihang University, CHINA

Received: March 13, 2016

Accepted: November 10, 2016

Published: December 29, 2016

Data Availability Statement: All experiment data can be found in the following URL: https://figshare.com/articles/Experiments/4029111.

Funding: This research has been funded with European Union AEROARMS Project with reference H2020-ICT-2014-1-644271, http://www.aeroarms-project.eu/. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

In recent years, many researchers have addressed the issue of making Unmanned Aerial Vehicles (UAVs) more and more autonomous. In this context, the state estimation of the vehicle position is a fundamental necessity for any application involving autonomy. However, the problem of position estimation could not be solved in some scenarios, even when a GPS signal is available, for instance, an application requiring performing precision manoeuvres in a complex environment. Therefore, some additional sensory information should be integrated into the system in order to improve accuracy and robustness. In this work, a novel vision-based simultaneous localization and mapping (SLAM) method with application to unmanned aerial vehicles is proposed. One of the contributions of this work is to design and develop a novel technique for estimating features depth which is based on a stochastic technique of triangulation. In the proposed method the camera is mounted over a servo-controlled gimbal that counteracts the changes in attitude of the quadcopter. Due to the above assumption, the overall problem is simplified and it is focused on the position estimation of the aerial vehicle. Also, the tracking process of visual features is made easier due to the stabilized video. Another contribution of this work is to demonstrate that the integration of very noisy GPS measurements into the system for an initial short period of time is enough to initialize the metric scale. The performance of this proposed method is validated by means of experiments with real data carried out in unstructured outdoor environments. A comparative study shows that, when compared with related methods, the proposed approach performs better in terms of accuracy and computational time.

1 Introduction

There are still important problems to be solved in autonomous robotics, and simultaneous localization and mapping (SLAM) is one of them. This paper tries to tackle this problem and contributes to give even more autonomy to mobile robots. Regarding the term SLAM, it is used to refer to a map building process in an unknown space and the use of this map to navigate through such an space tracking the position in a simultaneous process. Usually this map is built using the sensors that the device (an aerial vehicle in this case) have on board, (see [1, 2] for a complete survey).

Many different kinds of sensors can be used for implementing SLAM systems, for instance, laser ([3–5]), sonar ([6–8]), sound sensors ([9, 10]), RFID ([11, 12]) or computer vision ([13–15]). The selection of such a sensor technology has a great impact on the algorithm used in SLAM and, depending on the application and other factors, each technology has some strong and weak points.

This work proposes a novel vision-based SLAM method to be applied to a quadcopter. In the case of small unmanned aerial vehicles (UAVs), there exist several limitations regarding to the design of the platform, mobility and payload capacity that impose considerable restrictions on the available computational and sensing resources. Recently, the availability of lighter laser range finders has allowed the use of this kind of sensors in small UAVs. Some examples of SLAM systems with application to UAVs that make use of laser range finders are: [16, 17] and [18]. While a good performance can be obtained with laser range finders, video cameras still represent an excellent choice for its use in small UAVs. Those devices provide many data and can be hardware-embedded in aerial vehicles for their low weight and consumption at an affordable cost.

Specifically, monocular vision presents significant advantages respect other camera configurations (mainly stereo-vision). A single camera does not present the problem of a stereo rig with a fixed baseline between cameras limitating the operational range. But as a drawback the use of a single camera means to face some technical challenges: depth information has to be retrieved with many frames and, therefore, robust techniques for recovering the feature depth are needed. Some examples of recent works about general monocular SLAM systems that have shown great results are: [19–21].

Related work: There are different approaches for implementing monocular SLAM systems applied to aerial vehicles which some of them are variations of more general methods. In [22] SURF visual features are used within an EKF-based (Extended Kalman Filter) SLAM scheme. In this case, features are initialized into the state by using the undelayed inverse depth (UID) method, proposed in [23]. In [24] an homography-based SLAM approach is proposed. In this case homography-based techniques are used to compute the UAV relative translation and rotation by means of the images. The visual odometer is then integrated into the SLAM scheme via an EKF. The work in [25] also uses an homography-based method for estimating the motion of the vehicle. The computed motion is used as input of an EKF-SLAM that fuses inertial measurements. Initialization of features is done by the UID method. In [26], an EKF-based approach is proposed where feature depth is computed by triangulation between visual correspondences using SIFT descriptors. In [27] a method that estimates depth and vehicle states, by exploiting the orthogonality of indoor environments, is proposed. The SLAM formulation used in that work is the FastSLAM algorithm proposed in [28]. In [29] a fully navigation scheme (control and estimation) is proposed. In this case the Parallel Tracking and Mapping (PTAM) algorithm, described in [30], is used for implementing the SLAM system. In [31] an EKF scheme is embedded into the PTAM algorithm for fusing IMU (inertial measurement unit) data, in order to recover the absolute scale of estimations. In [32] a variation of the PTAM algorithm is proposed to be applied in environments with very few visual features. In [33] another variation of the PTAM algorithm is proposed. A Bayesian filter that explicitly models outlier measurements is used to estimate the depth of feature locations: a 3D point is only inserted in the map when the corresponding depth-filter has converged.

As it can appreciated from the above approaches in literature, most of them are filter-based methods, Keyframe methods (PTAM), or a mixture of them. While Keyframe methods are shown to give accurate results when the availability of computational power is enough, filtering-based SLAM methods might be beneficial if limited processing power is available [15].

Objectives and contributions: In this work authors propose a new filter-based monocular SLAM scheme. The method presented in this research has been designed for taking advantage of hardware resources commonly available in this kind of platforms. The performance of the method is validated by means of experiments with real data carried out in unstructured outdoor environments. An extensive comparative study is presented for contrasting the operative and effectiveness of this proposal respect to other relevant methods. One of the contributions of this work is to present a novel technique for estimating the features depth. The proposed approach is based on a stochastic technique of triangulation. While this technique is inspired in a previous authors' work [34], crucial and contributive modifications have been introduced in order to accommodate it to the particularities of the current application:

- In this work, the camera is mounted over a servo-controlled gimbal that counteracts the changes in attitude of the quadcopter. Due to the above assumption, the overall problem is simplified and it is focused on the position estimation of the MAV. Also, the tracking process of visual features is made easier due to the stabilized video.
- Instead of using an external pattern of known dimensions, in this work the GPS signal is used during a short initial period of time for recovering the metric scale of the estimates.
- Features are directly parametrized in their euclidean form, instead of the inverse depth parametrization. The consequence is a reduction of the computational cost of the filter due to the reduction of the dimension of the system state.
- A novel technique for the tracking process of candidate points is proposed. In this case the search of visual features is limited to regions of the image circumscribed by ellipses derived from epipolar constraints. The consequence is an improvement in the execution time.

Compared with other methods presented in literature, one of the contributions of this work is to demonstrate that the integration of very noisy GPS measurements into the system for an initial short period is enough to initialize the metric scale. For example in [35] the monocular scale factor is retrieved from a feature pattern with known dimensions. In [29] and [36], the map is initially set by hand, by aligning the first estimates with the ground-truth in order to get the scale of the environment. Additionally, the proposed approach is simpler when compared with similar approaches, because the estimation of the camera orientation is avoided by using the servo-controlled gimbal. In [26] feature depth is computed by direct triangulation between visual correspondences using SIFT descriptors. In this work, a novel technique, which is based on patch-correlation, is used for the tracking process of candidate points. It is well known that local descriptors like SIFT or SURF are more robust that the use of patch-correlation techniques for matching visual features. Nevertheless, the stabilized video and the stochastic nature of the whole initialization method makes reliable the technique proposed in this work for tracking candidate points, with the implicit gain in terms of computational cost.

Perhaps, the most extended technique that is used for initializing map features in EKF-SLAM is the UID based methods (e.g. [22, 25]). Nevertheless, the comparison study presented in this work shows that the proposed method can surpass the UID method in terms of accuracy and computational time, at least for the kind of application studied.

Paper outline: Section 2 states the problem description and assumptions. Section 3 describes the proposed method in a detailed manner. In Section 4 experimental results are shown together with a comparative study and the discussion about those results and, finally, Section 5 presents the conclusions of this work.

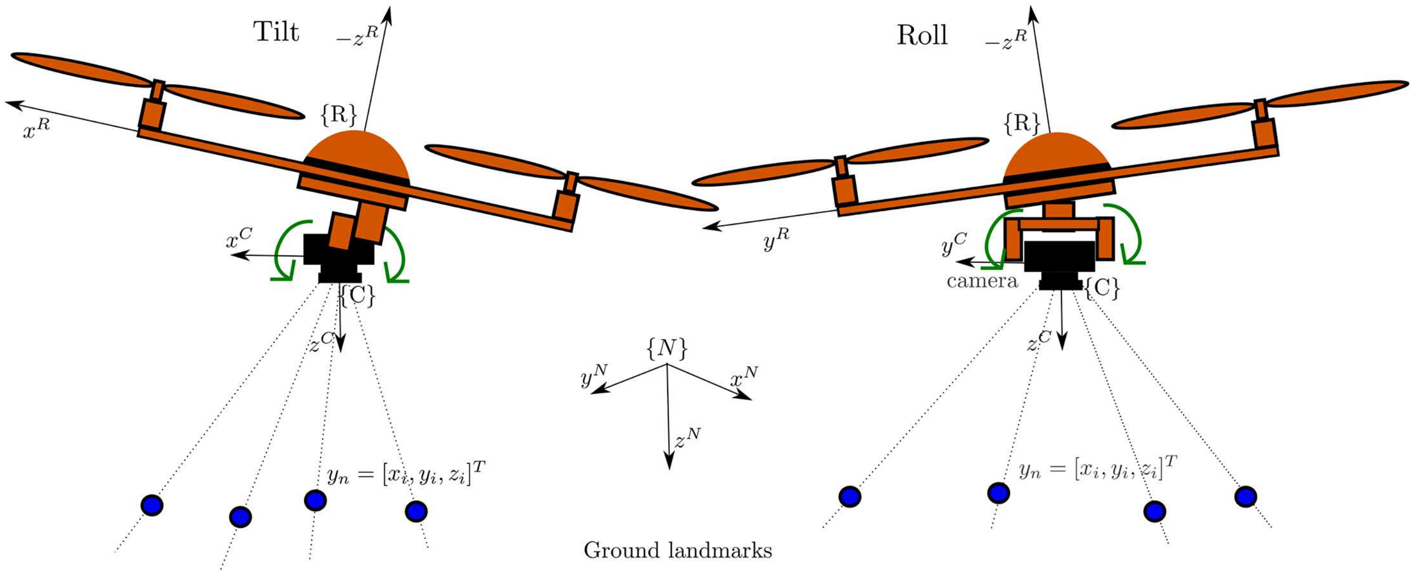

Fig 1. Coordinate systems: the local tangent frame is used as the navigation reference frame ^{N}. Monocular camera is mounted over a servo-controlled gimbal that counteracts the changes in attitude of the quadcopter.

doi:10.1371/journal.pone.0167197.g001

2 Assumptions

The platform considered in this work is a quadrotor with free movements in any direction in $\mathbb{R}^3 \times SO(3)$, shown in Fig 1. However, it is important to highlight that the proposed monocular SLAM method could be applied to other kind of platforms. The proposed method is mainly intended for local autonomous vehicle navigation. In this case, the local tangent frame is used as the navigation reference frame. Thus, the initial position of the vehicle defines the origin of the navigation coordinates frame. The navigation system follows the NED (North, East, Down) convention. The magnitudes expressed in the navigation and in the camera frame are denoted respectively by the superscripts ^{N} and ^{C}. All the coordinate systems are right-handed defined. It is also assumed that the location of the origin of camera frame respect to other elements of the quadcopter (e.g. GPS antenna) is known and fixed. In this case, the position of the origin of the vehicle can be computed from the estimated location of the camera.

In aerial vehicles applications, the attitude estimation is well handled by available systems (e.g. [37] and [38]), therefore, this work will focus in position estimation. Also, it is assumed that the monocular camera is mounted over a servo-controlled gimbal (see Fig 1). This kind of accessory, used mainly for stabilizing video capture, has become very common in aerial applications. In our case, the gimbal is configured in order to counteract the changes in attitude of the quadcopter, and therefore stabilizing the orientation of the camera towards the ground (down axis in NED coordinates). The above consideration has two important consequences: i) the tracking process of visual features is made easier due to the stabilized video, ii) the orthogonal matrix R^{CN}, defining the rotation of the camera frame to the navigation frame, is assumed to be known.

An standard monocular camera is considered. In this case, a central-projection camera model is assumed. The image plane is located in front of the camera's origin where a non-inverted image is formed. The camera frame ^{C} is right-handed with the z-axis pointing to the field of view.

The $\mathbb{R}^3 \Rightarrow \mathbb{R}^2$ projection of a 3D point located at $p^N = (x, y, z)^T$ to the image plane $p = (u, v)$ is defined by:

$$u = \frac{x'}{z'} \quad v = \frac{y'}{z'} \tag{1}$$

Let u and v be the coordinates of the image point p expressed in pixel units, and:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} p^C \tag{2}$$

Let p^C be the same 3D point p^N, but expressed in the camera frame ^{C} by $p^C = R^{NC} p^N$. Let R^{NC} be the rotation matrix that allows to transform from the navigation frame ^{N} to the camera frame ^{C}. Also, it is fulfilled that $R^{NC} = (R^{CN})^T$, and R^{CN} is known by the use of the gimbal.

Inversely, a directional vector $h^C = [h^C_x, h^C_y, h^C_z]^T$ can be computed from the image point coordinates u and v as

$$h^C(u, v) = \left[\frac{u_0 - u}{f}, \frac{v_0 - v}{f}, 1\right]^T \tag{3}$$

Vector h^C points from the camera optical center position to the 3D point location and it can be expressed in the navigation frame by $h^N = R^{CN} h^C$. Note that for the $\mathbb{R}^2 \Rightarrow \mathbb{R}^3$ mapping case, defined in Eq 3, depth information is lost.

The distortion caused by the camera lens is considered through the model described in [39]. Using the former model (and its inverse form), undistorted pixel coordinates (u, v) can be obtained from (u_d, v_d) and conversely. In this case, it is assumed that the intrinsic parameters of the camera are already known: focal length f, principal point (u_0, v_0), and radial lens distortion $k_1, \ldots, k_n$.

3 Method description

3.1 Problem description

The main goal of the proposed method is to estimate the following system state x:

$$x = [x_v, y_1, y_2, \ldots, y_n]^T \tag{4}$$

where x_v represents the state of the camera-quadcopter, and y_i represents the location of the i-th feature point in the environment. At the same time, x_v is composed of:

$$x_v = [r^N, v^N]^T \tag{5}$$

Let $r^N = [p_x, p_y, p_z]$ represent the position of the vehicle (camera) expressed in the navigation frame. Let $v^N = [v_x, v_y, v_z]$ denote the linear velocity of the robot expressed in the navigation frame. The location of a feature y_i is parametrized in its euclidean form:

$$y_i = [p_{x_i} p_{y_i}, p_{z_i}]^T \tag{6}$$

3.2 Prediction

The work presented in this paper is motivated by the application of monocular SLAM to small aerial vehicles. In this case, and due to limited resources commonly available in this kind of applications, the filtering-based SLAM methods seem to be still more appropriate than Keyframe methods. Moreover, filtering-based methods are better suited for incorporating, in a simple manner, additional sensors to the system. In this sense, most robotic applications make use of multiple sensor inputs.

The architecture of the system is defined by the typical loop of prediction-updates steps in the EKF in direct configuration, where the EKF propagates the vehicle state as well as the feature estimates. In this case, the camera-vehicle system state x_v takes a step forward by the following simple model:

$$\begin{cases} r^N_{k+1} = r^N_k + v^N_k \Delta t \\ v^N_{k+1} = v^N_k + V^N \end{cases} \tag{7}$$

At every step, it is assumed that there is an unknown linear velocity with acceleration zero-mean and known-covariance Gaussian processes σ_a, producing an impulse of linear velocity: $V^N = \sigma^2_a \Delta t$.

It is assumed that the map features y_i remain static (rigid scene assumption) so $x_{k+1} = [x_{v(k+1)}, y_{1(k)}, y_{2(k)}, \ldots, y_{n(k)}]^T$.

The state covariance matrix $\boldsymbol{P}$ takes a step forward by:

$$P_{k+1} = \nabla F_x P_k \nabla F^T_x + \nabla F_u Q \nabla F^T_u \tag{8}$$

where Q and the Jacobians ∇F_x, ∇F_u are defined as:

$$\nabla F_x = \begin{bmatrix} \frac{\partial f_v}{\partial x_v} & 0_{6\times n} \\ 0_{n\times 6} & I_{n\times n} \end{bmatrix}, \nabla F_u = \begin{bmatrix} \frac{\partial f_v}{\partial u} & 0_{6\times n} \\ 0_{n\times 3} & 0_{n\times n} \end{bmatrix}, Q = \begin{bmatrix} U & 0_{3\times n} \\ 0_{n\times 3} & 0_{n\times n} \end{bmatrix}, \tag{9}$$

Let $\frac{\partial f_v}{\partial x_v}$ be the derivatives of the equations of the nonlinear prediction model (Eq 7) with respect to the robot state x_v. Let $\frac{\partial f_v}{\partial u}$ be the derivatives of the nonlinear prediction model with respect to the system input u. Uncertainties are incorporated into the system by means of the process noise covariance matrix $U = \sigma^2_a I_{3\times 3}$, through parameter σ^2_a.

3.3 Detection of candidate points

The proposed method states that a minimum number of features y_i is considered to be predicted appearing in the image, otherwise new features should be added to the map. In the latter case, new points are detected in the image through a random search. For this purpose, Shi-Tomasi corner detector [40] is applied, but other detectors could also be used. These points in the image, which are not added yet to the map, are called candidate points (see Fig 2). Only image areas free of both, candidate points and mapped features, are considered for detecting new points with the saliency operator.

At the k-th frame, when a visual feature is detected for the first time, the following entry c_l is stored in a table:

$$\mathrm{c}_l = \left[(t^N_{c_0})^T, \theta_0, \phi_0, P_{y_i}, u, v \right] \tag{10}$$

where $y_{c_i} = [(t^N_{c_0})^T, \theta_0, \phi_0]$ models a 3D semi-line, defined on one side by the vertex $(t^N_{c_0})^T$, corresponding to the current optical center coordinates of the camera expressed in the navigation frame, and pointing to infinite on the other side with azimuth and elevation θ_0 and ϕ_0,

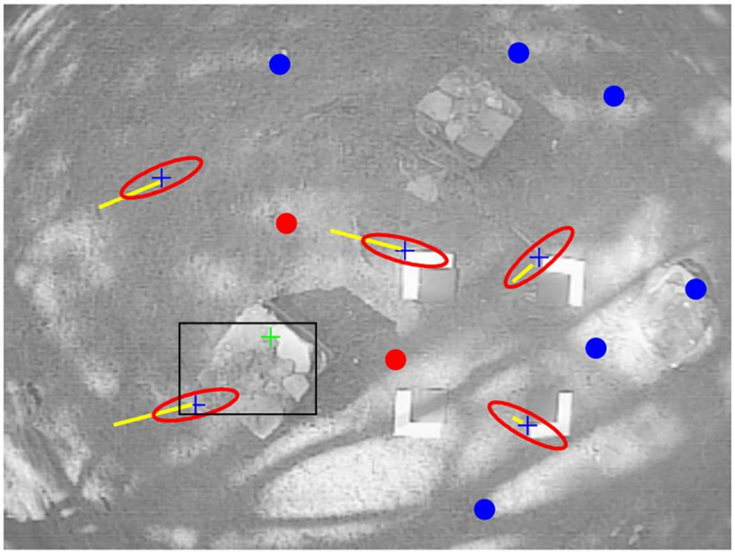

Fig 2. New candidate points are randomly detected in image regions that are empty of map features or candidate points being tracked. In this frame, the black rectangle indicates the current search region where new candidate points have been detected (green cross mark). In order to speed up the tracking process of candidate points, a search region is established constrained by ellipses (in red) aligned with the epipolar lines (in yellow). Candidate points being tracked are indicated by blue cross marks. Visual features already mapped are indicated by dots. Red dots indicate unsuccessfully matches.

doi:10.1371/journal.pone.0167197.g002

respectively, being:

$$\begin{aligned} \theta_0 &= \mathrm{atan2}(h_y^N, h_x^N) \\ \phi_0 &= \mathrm{acos}\left(\frac{h_z^N}{\sqrt{(h_x^N)^2 + (h_y^N)^2 + (h_z^N)^2}}\right) \end{aligned} \tag{11}$$

where $h^N = [h_x^N, h_y^N, h_z^N]^T$ is computed as indicated in Section 2. Let P_{y_i} be a 5 × 5 covariance matrix which models the uncertainty of y_i. Therefore $P_{y_i} = JPJ^T$, where P is the system covariance matrix and J is the Jacobian matrix formed by the partial derivatives of the function $y_{c_i} = h(x, z_{uv})$ with respect to $[x, z_{uv}]^T$. Let $[u, v]$ be the location in the image of the candidate point.

Also, a $p \times p$-pixel window, centered in $[u, v]$ is extracted and linked to the corresponding candidate point.

3.4 Tracking of candidate points

At every subsequent frame $k + 1, k + 2...k + n$, the location of candidate points is tracked. In this case, a candidate point is predicted to appear inside an elliptical region S centered in the point $[u, v]$, taken from c_l, see Fig 3.

In order to optimize the speed of the search, the major axis of the ellipse is aligned with the epipolar line defined by image points e_1 and e_2.

The epipole e_1 is computed by projecting $t_{c_0}^N$, which is stored in c_l, to the current image plane by Eqs 1 and 2. The point e_2 is computed by projecting the 3D point p^N defined by the

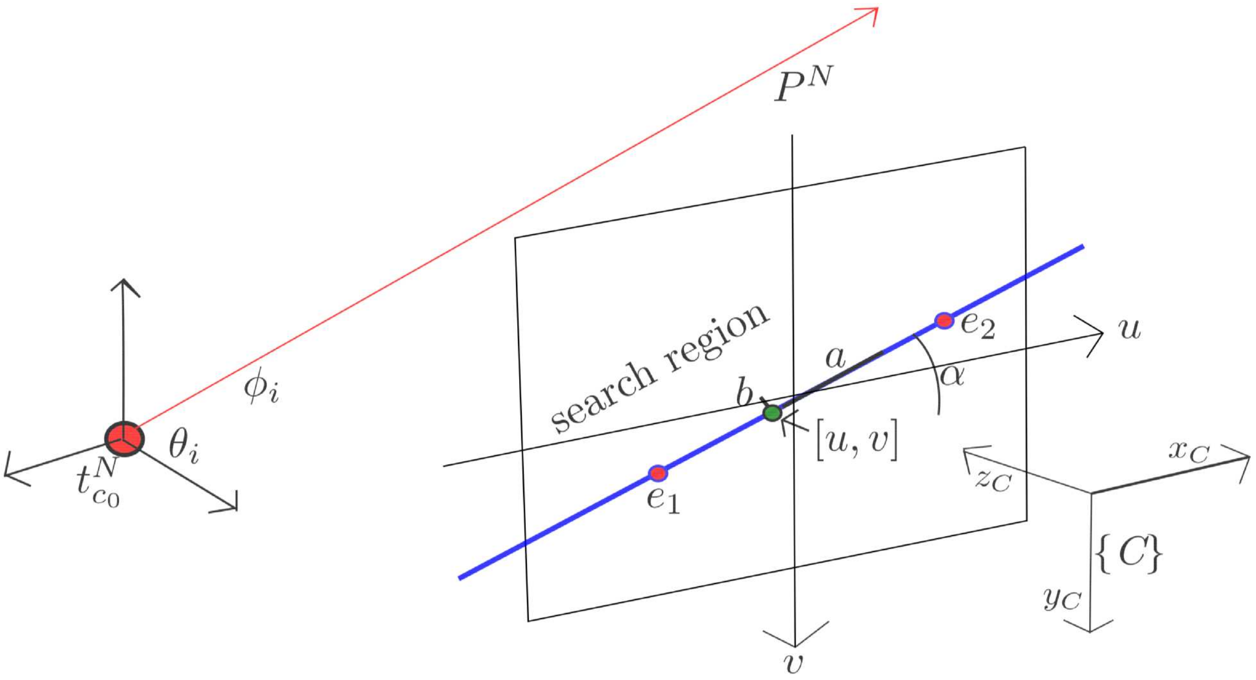

Fig 3. The established search region for matching candidate points is constrained by ellipses aligned with the epipolar line.

doi:10.1371/journal.pone.0167197.g003

data stored in c_l, through Eqs 1 and 2 also, but assuming a depth equal to one ($d = 1$). In this case, p^N models a 3D point located at:

$$p^N = t^N_c + d(m(\theta_i, \phi_i)) \tag{12}$$

where $m(\theta_i, \phi_i)$ is a directional unitary vector defined by:

$$m(\theta_i, \phi_i) = (\cos\theta_i \sin\phi_i,\ \sin\theta_i \sin\phi_i,\ \cos\phi_i)^T \tag{13}$$

The orientation of the ellipse S_c is determined by $\alpha_c = \text{atan2}(e_y, e_x)$ where $e = e_2 - e_1$ and e_y, e_x represent the y and x coordinates respectively of e. The size of the ellipse S_c is determined by its major and minor axis, respectively a and b. In this case a and b are free parameters constrained to $b \ll a$.

The ellipse S_c is represented in its matrix form by:

$$\begin{aligned} S_c &= R_c \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} R_c^T \\ R_c &= \begin{bmatrix} \cos\alpha_c & -\sin\alpha_c \\ \sin\alpha_c & \cos\alpha_c \end{bmatrix} \end{aligned} \tag{14}$$

The ellipse S_c represents a probability region where the candidate point must lie in the current frame. In this case, patch cross-correlation is applied over all the image locations $[u_S, v_S]$ within the search region. If the score of a location $[u_S, v_S]$, determined by the best cross-correlation between the candidate patch and the n patches defined by the region of search, is higher than a threshold, then this pixel location $[u_S, v_S]$ is considered as the current candidate point location. Thus, c_l is updated with $[u, v] = [u_S, v_S]$.

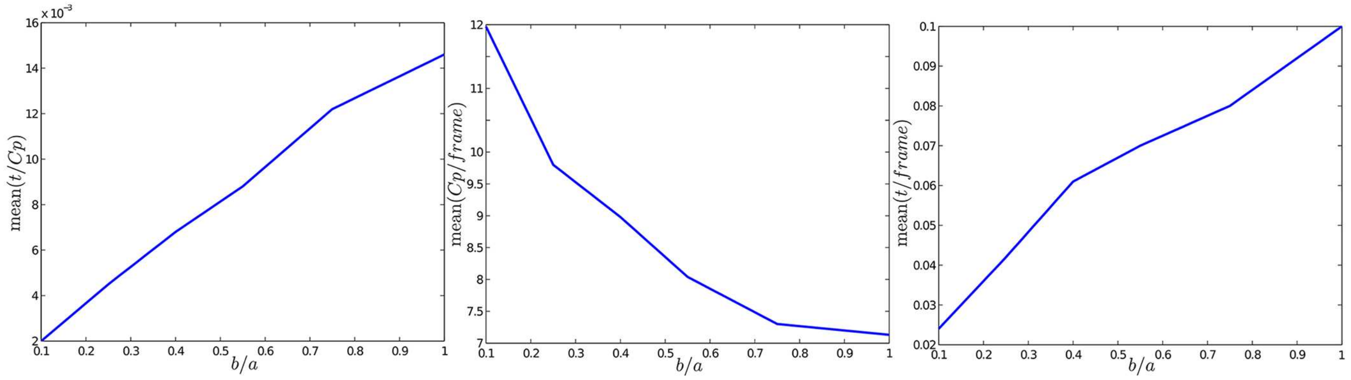

Fig 4. Variation of the relation between ellipse S_c axes (*b*/*a*). Left plot: average tracking time for a candidate point. Middle plot: average number of candidate points being tracked at each frame. Right plot: average total time per frame.

doi:10.1371/journal.pone.0167197.g004

At this stage, there is not yet reliable information about the depth of candidate points. For this reason, it is difficult to determine an optimal size of the search ellipse. In this case, the parameter a is chosen empirically in function of the particularities of the application as the velocity of the vehicle and the frame rate of the video. In this work, good results were found with a value of $a = 20$ pixels.

Nevertheless, the effects obtained by the variation of the relation of (b/a), which determines the proportion of the ellipse, can be investigated. In this case, some experimental results were obtained using the same methodology described in Section 4. The results can be summarized as follows (see Fig 4):

- As the ellipse tends to be a circle, the time required to track a candidate point increases considerably (left plot).
- When the ellipse tends to be a circle the number of candidate points being tracked is lower (middle plot). This is because when the ellipse is too thin, some candidate points are lost and new ones must be detected.
- When the parameter b is chosen in order to define a very thin ellipse, the total time required for the whole tracking process of candidate points is much lower (right plot).

Based on the above results, the value of parameter b is recommended to be ten times lower than a.

It is important to note that no extra effort is put in order to obtain a more robust descriptor. There are two main reasons for supporting this approach: i) The method proposed for tracking the candidate points is applied only during an initial short period when a new visual feature is detected. During this initial period, prior to the initialization of the candidate point as a new map feature, some information about the feature depth is gathered. ii) Different from the general problem of the monocular SLAM, the stabilized video also makes easier the tracking process of candidate points.

3.5 Feature initialization

Depth information cannot be obtained in a single measurement when bearing sensors (e.g. a projective camera) are used. To infer the depth of a feature, the sensor must observe this feature repeatedly as this sensor moves freely through its environment, estimating the angle from the feature to the sensor center. The difference between those angle measurements is the

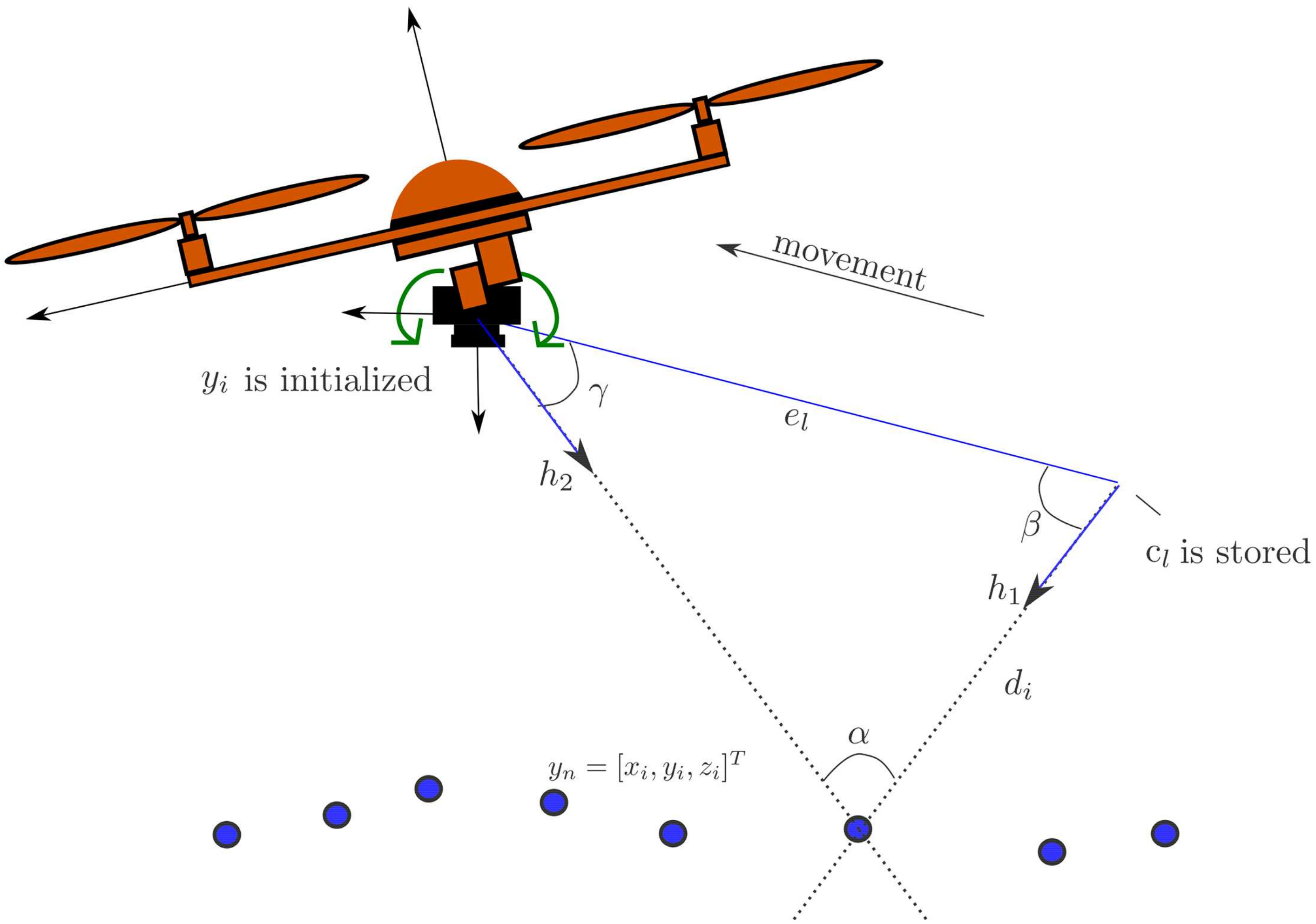

Fig 5. An hypothesis of depth d_i of a candidate point is computed by triangulation between the first location when the point was detected and the current location of the aerial vehicle.

doi:10.1371/journal.pone.0167197.g005

parallax angle. Actually, parallax is the key that allows estimating features depth. In case of indoor sequences, a displacement of centimeters could be enough to produce parallax; on the other hand, the more distant the feature, the more the sensor has to travel to produce parallax (see Fig 5).

Every time that a new image location $z_{uv} = [u, v]$ is obtained for a candidate point c_l, an hypothesis of depth d_i is computed by:

$$d_i = \frac{\|e_l\| \sin\gamma}{\sin\alpha} \tag{15}$$

Let $\alpha_i = \pi - (\beta + \gamma)$ be the parallax. Let $e_l = t^N_{c_0} - t^N_c$ indicate the displacement of the camera from its first observation to its current position with:

$$\beta = \cos^{-1}\left(\frac{h_1 \cdot e_l}{\|h_1\|\|e_l\|}\right) \quad \gamma = \cos^{-1}\left(\frac{h_2 \cdot -e_l}{\|h_2\|\|e_l\|}\right) \tag{16}$$

Let β be the angle defined by h_1 and e_l. Let h_1 be the normalized directional vector $m(\theta, \phi) = (\cos\theta \sin\phi, \sin\theta \sin\phi, \cos\phi)^T$ computed taking θ_0, ϕ_0 from c_l, and where γ is the angle defined by h_2 and $-e_l$. Let $h_2 = h^N$ be the directional vector pointing from the current camera optical center to the feature location computed as indicated in Section 2 from the current measurement z_{uv}.

At each step, the hypothesis of depth d_i is low-pass filtered because the depth estimated by triangulation varies considerably, specially for low parallax. In previous authors' work [34] is demonstrated that only a few degrees of parallax is enough to reduce the uncertainty in depth estimations.

When parallax α_i is greater than a specific threshold ($\alpha_i > \alpha_{min}$) a new feature $y_{new} = [p_{x_i}, p_{y_i}, p_{z_i}]^T$ is added to the system state vector x:

$$x_{new} = [x_{old}; y_{new}]^T \tag{17}$$

where

$$y_{new} = t^N_{c_0} + m(\theta_0, \phi_0) d_i \tag{18}$$

The system state covariance matrix P is updated by:

$$P_{new} = \begin{bmatrix} P_{old} & 0 \\ 0 & P_{y_{new}} \end{bmatrix} \tag{19}$$

Let $P_{y_{new}}$ be the 3×3 covariance matrix which models the uncertainty of the new feature y_{new}, and:

$$P_{y_{new}} = J \begin{bmatrix} P_{y_i} & 0 \\ 0 & \sigma_d^2 \end{bmatrix} J^T \tag{20}$$

In Eq 20, P_{y_i} is taken from c_l. Let σ_d^2 be a parameter modelling the uncertainty of process of depth estimation. Let J be the Jacobian matrix formed by the partial derivatives of the function $y_{new} = h(c_l, d_i)$ with respect to $[(t^N_{c_0})^T, \theta_0, \phi_0, d_i]^T$.

3.6 Visual updates and map management

The process of tracking visual features y_i is conducted by means of an active search technique [41]. In this case, and in different way from the tracking method described in subsection 3.4, the search region is defined by the innovation covariance matrix S_i, where $S_i = \nabla H_i P_{k+1} \nabla H_i^T + R_i$.

Assuming that for the current frame, n visual measurements are available for features $y_1, y_2, \ldots, y_n$, then the filter is updated with the Kalman update equations as:

$$\begin{cases} x_k = x_{k+1} + K(z - h) \\ P_k = P_{k+1} - KSK^T \\ K = P_{k+1} \nabla H^T S^{-1} \\ S = \nabla H P_{k+1} \nabla H^T + R \end{cases} \tag{21}$$

where $z = [z_{uv_1}, z_{uv_2}, \ldots, z_{uv_n}]^T$ is the current measurement vector. Let $h = [h_1, h_2, \ldots, h_n]^T$ be the current prediction measurement vector. The measurement prediction model $h_i = (u, v) = h(x_v, y_i)$ has been defined in Section 2. Let K be the Kalman gain. Let S be the innovation covariance matrix. Let $\nabla H = [\nabla H_1, \nabla H_2, \ldots, \nabla H_n]^T$ be the Jacobian formed by the partial derivatives of the measurement prediction model $h(x)$ with respect to the state x, as:

$$\nabla H_i = \left[\frac{\partial h_i}{\partial x_v}, \ldots 0_{2\times 3} \ldots, \frac{\partial h_i}{\partial y_i}, \ldots 0_{2\times 3} \ldots\right] \qquad (22)$$

Let $\frac{\partial h_i}{\partial x_v}$ be the partial derivatives of the equations of the measurement prediction model h_i with respect to the robot state x_v. Let $\frac{\partial h_i}{\partial y_i}$ be the partial derivatives of h_i with respect to feature y_i. Note that $\frac{\partial h_i}{\partial y_i}$ has only a nonzero value at the location (indexes) of the observed feature y_i. Let $R = (I_{2n\times 2n})\sigma^2_{uv}$ be the measurement noise covariance matrix.

A SLAM framework that works reliably in a local way can easily be applied to large-scale problems using different methods, such as sub-mapping, graph-based global optimization [15] or global mapping [42]. Therefore, in this work, large-scale SLAM and loop-closing are not considered. Nevertheless these problems have been intensively studied in the past. Candidate points whose tracking process fails are pruned from the system. In a similar way, visual features with high percentage of mismatching are removed from the system state and covariance matrix.

3.7 Metric scale and System initialization

Even when GPS signal is available, the problem of position estimation could not be solved for some specific scenarios, for instance in an application requiring performing precision manoeuvres in a complex environment. In this case, and due to several sources of error, the position obtained with a GPS can vary even for meters in a matter of seconds for a static location [43]. In such a scenario, the use of GPS readings, smoothed or not, as feedback signal of the control system can be unreliable because the control is not able to discriminate between sensor noise or actual small movements of the vehicle (see Fig 6).

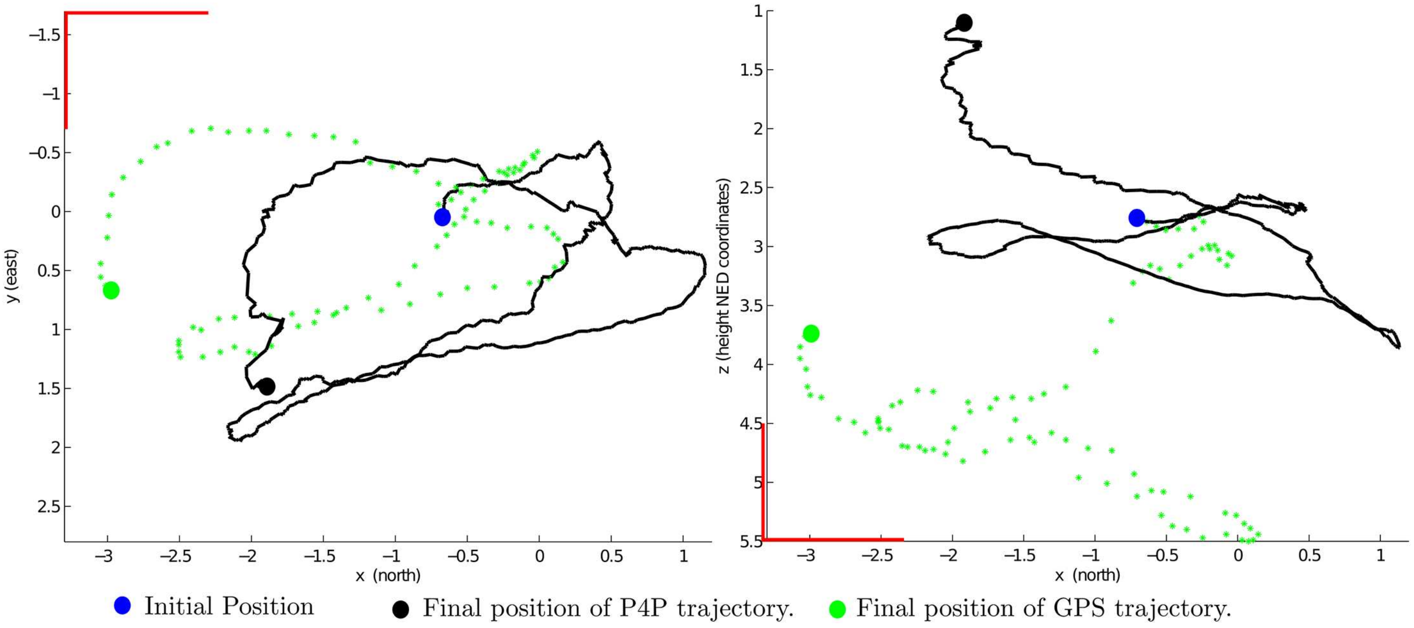

Fig 6. Example of the GPS position measurements obtained for a flight trajectory. Top view (left plot) and lateral view (right plot). In this case, the flight trajectory has been computed using the P4P method described in the Appendix. Observe the error drift in GPS readings.

doi:10.1371/journal.pone.0167197.g006

On the other hand, in a robotics context, obtaining the metric scale of the world can be a tough requirement. However, one of the most challenging aspects of working with monocular sensors has to do with the impossibility of directly recovering the metric scale of the world. If no additional information is used, and a single camera is used as the solely source of data to the system, the map and trajectory can only be recovered without metric information [14]. In this case, neither monocular vision nor GPS are suitable to be used separately for navigation purposes.

In this work, noisy data obtained from the GPS is incorporated into the system at the beginning in order to incorporate the metric information of the environment. After some initial period of convergence, where the system is considered to be in a initialization mode, the system can operate relying only on visual information.

Position measurements obtained from the GPS are modelled by:

$$y_r = r^N + v_r \tag{23}$$

where v_r is a Gaussian white noise with PSD σ_r^2; and r^N has been already defined in Eq 7.

Commonly, position measurements are obtained from GPS devices in geodetic coordinates (*latitude*, *longitude* and *height*). Therefore, in Eq 23 it is assumed that GPS position measurements have been previously transformed to their corresponding local tangent frame coordinates. It is also assumed that the offset between the GPS antenna and the vehicle frame has been taken into account in the previous transformation.

For system updates, the simple measurement model $h_r = h(x_v)$ is used:

$$h_r = [p_x, p_y, p_z]^T \tag{24}$$

In the next Section, the demonstration that the proposed method is robust enough to be initialized with noisy GPS measurements will be shown.

4 Experimental Results

4.1 Experimental setup

In Fig 7 is shown the vehicle that authors used to obtain real data for experiments, the platform is a customized quadrotor. Such a platform uses an Ardupilot unit, [44], as flight controller. As main sensors, the platform is equipped with a radio telemetry unit (3DR at 915MHz), GPS unit (NEO-M8N), camera (DX201 DPS) with wide angle lens and a video transmitter (at 5–8 GHz). The camera is mounted over a very low-cost gimbal which is servo-controlled by standard servos. During the experiments, the quadrotor has been controlled by radio in a manual way.

For capturing sensor data and digitalized video from the vehicle a software application has been built by authors in C++ language. The protocol used for reception/transmission is MAVLINK protocol [45]. GPS and AHRS (Attitude and Heading Reference System) data are synchronized between them and recorded in a database for their study. Video frames have been acquired at a resolution of 320x240 gray scale pixels at 25 *fps*. All the experiments have been performed in an outdoor park with trees, which surface is almost flat with grass and some dirt areas. Flight observations include some plants and small structured parts. In average 9–10 GPS satellites are visible at the same time. Finally, a MATLAB implementation of the proposed method was executed offline over the dataset in order to estimate the flight trajectory and the map of the environment. In experiments, for evaluating the performance of the proposed method, the technique P4P described in the Appendix was used in order to have an external reference of the flight trajectory. In the following website reader can download the different

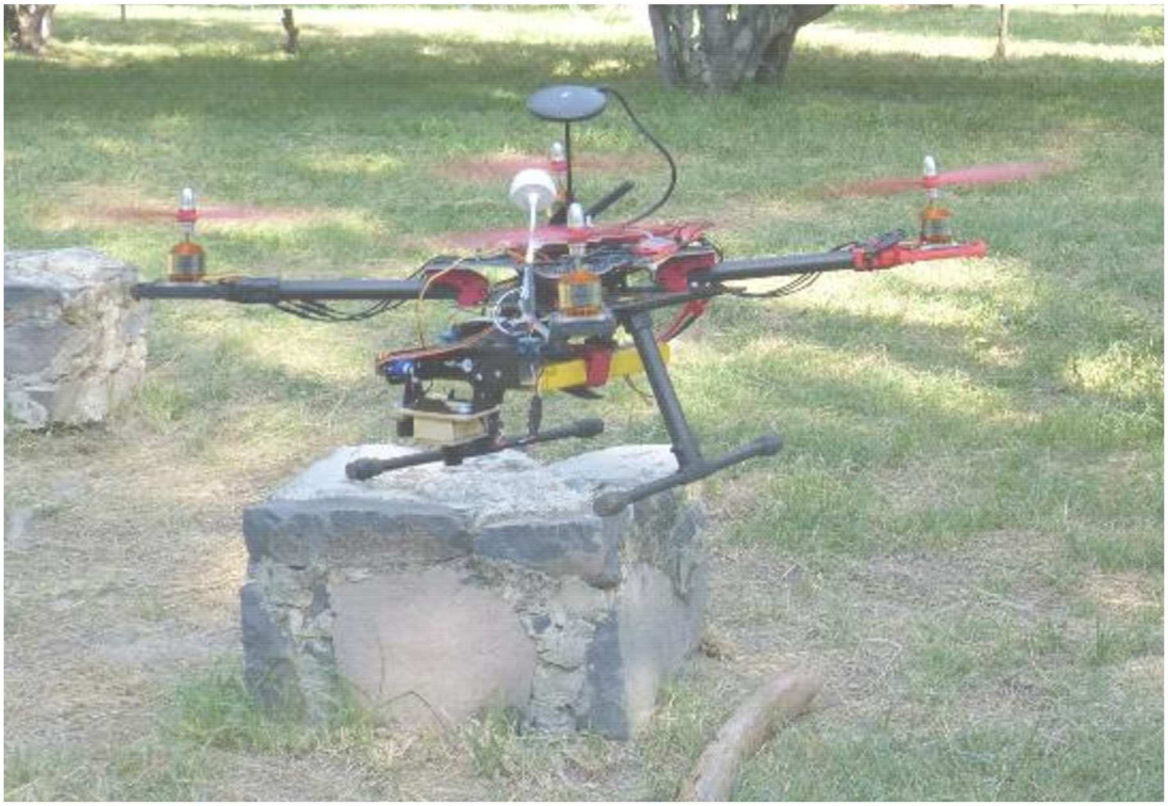

Fig 7. Data obtained from the sensors of a radio-controlled quadrotor has been used for testing the proposed method. A urban park was used as flight field.

doi:10.1371/journal.pone.0167197.g007

files containing all the data collected by robot sensors. This data has been used by authors to perform the experiments contained in this research paper (https://figshare.com/articles/Experiments/4029111).

4.2 Flight trajectories

Two different flight trajectories (F_a and F_b) were performed over the test field. In both cases, an initial period of 5 seconds (from $t = 0s$ to $t = 5s$) of flight was considered for initialization purposes as it was explained in section 3.7. Fig 8 shows some frames of the video recorded in flight F_a. At the beginning of the trajectory (left plot), at instant $t = 2.84s$, the first features are added to the system state. Note that at this moment, most of the tracked points are considered as candidate points. At instant $t = 10.23s$ (middle plot), the system is operating relying only on visual information for estimating the position of the quadcopter and the map of the environment. The right plot shows a frame at instant $t = 30.11s$. Fig 9 shows a 3D perspective of the estimated map and trajectory for both flight trajectories F_a and F_b. In the next subsection, a more detailed analysis of the experimental results is presented.

4.3 Comparative study

A comparative study has been performed in order to gain more insight about the performance of the proposed delayed monoSLAM (DE) method. For this purpose, the DE method has been tested against the popular undelayed inverse depth method (UID), and its variant, the undelayed inverse depth to euclidean method (UID2E). The implementation of the UID and

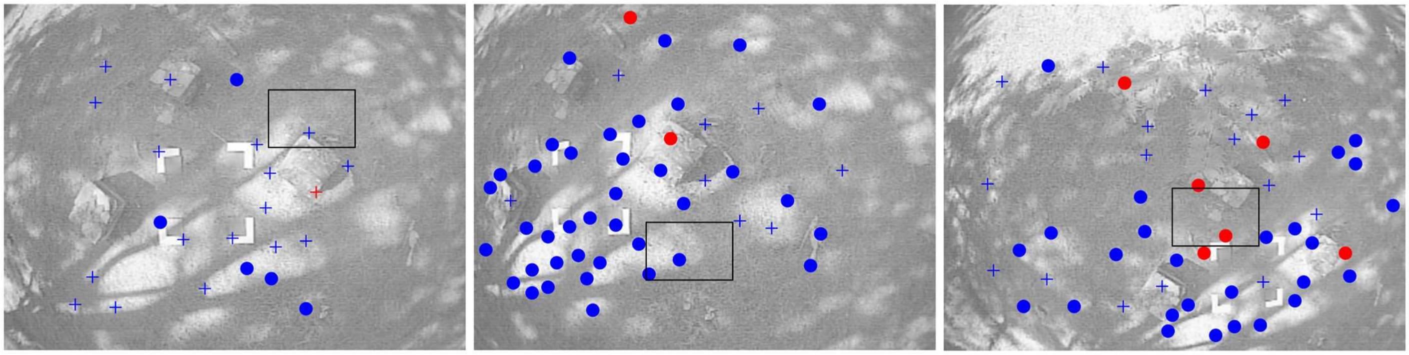

Fig 8. Frames taken from flight F_a: instant t = 2.84s (left plot), instant t = 10.23s (middle plot) and instant t = 30.11s (right plot). Candidate points being tracked are indicated by blue-cross marks. Visual features already mapped are indicated by dots. Red dots indicate unsuccessfully matches. Also note the four marks used for computing the external P4P flight trajectory.

doi:10.1371/journal.pone.0167197.g008

UID2E methods are based respectively on [23] and [46]. The UID and UID2E methods have been chosen because the undelayed inverse depth method has become almost a standard for implementing filter-based monocular-SLAM systems. In experiments, the 1-point RANSAC method [47] has been used for validating the visual matches of map features. In the particular case of the DE method, no extra validation technique was used for the matching process of candidate points. For the DE method, a value of $\alpha_{min} = 5^\circ$ has been used. For the UID and UID2E methods, values of $\rho_{ini} = 1$ and $\sigma_{\rho_{ini}} = 1$ have been used. In general all the methods are tested under the same conditions. Only the parameter σ_r^2, used for modelling the uncertainty in GPS readings during the initialization period has slightly been tuned for each method in order to produce a good initial metric convergence.

The search of new candidate points in each frame is conducted in a random manner for the DE method as well as the search of new features in UID and UID2E methods. For this reason,

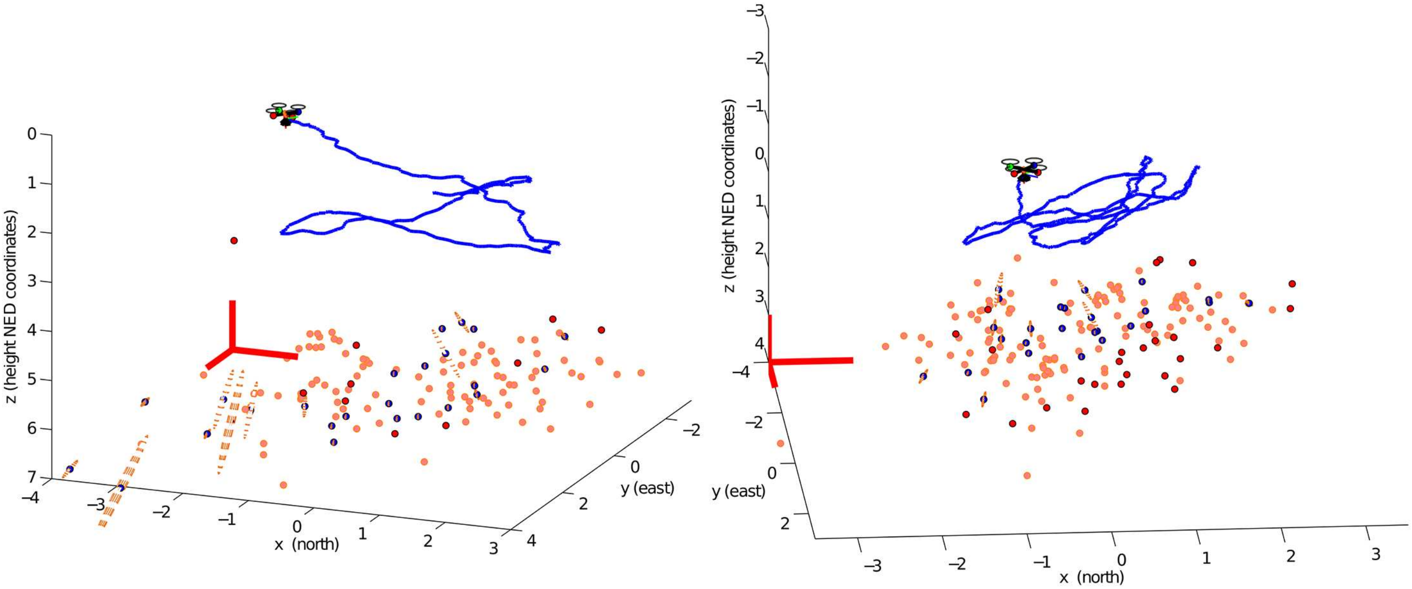

Fig 9. Estimated map and trajectory 3D plots obtained with the proposed delayed monoSLAM method: flight F_a (left plot) and flight F_b (right plot). Uncertainty in features position is indicated by 3D ellipses. Physical structure of the environment is partially recovered observing visual features.

doi:10.1371/journal.pone.0167197.g009

Table 1. Results for flight trajectory F_a.

Method	MD(p)	NIF	NDF	ETF (s)	TTE (s)	aMAE (m)
DE	15	127 ±7σ	76 ±4σ	.49 ±.11σ	268 ±16σ	.19 ±.08σ
UID	15	302 ±20σ	233 ±32σ	.62 ±.14σ	336 ±19σ	.29 ±.18σ
UID2E	15	305 ±18σ	246 ±23σ	.59 ±.11σ	323 ±9σ	.50 ±.36σ
DE	20	90 ±8σ	56 ±11σ	.34 ±.07σ	187 ±5σ	.20 ±.11σ
UID	20	218 ±11σ	175 ±12σ	.43 ±.09σ	234 ±17σ	.31 ±.23σ
UID2E	20	216 ±6σ	171 ±5σ	.40 ±.07σ	217 ±6σ	.42 ±.30σ
DE	25	64 ±2σ	39 ±6σ	.26 ±.06σ	141 ±7σ	.23 ±.13σ
UID	25	159 ±9σ	124 ±5σ	.29 ±.06σ	162 ±9σ	.32 ±.19σ
UID2E	25	162 ±9σ	133 ±8σ	.30 ±.05σ	164 ±12σ	.53 ±.36σ

doi:10.1371/journal.pone.0167197.t001

the results of the methods can vary at each run. In this case, in order to have a better statistical appreciation of the performance of each method, 10 Monte Carlo runs have been executed for computing each result.

Tables 1 and 2 show the results obtained respectively for the flight trajectory F_a and F_b. The number of visual features being tracked at each frame can affect the performance of monocular SLAM methods. For this reason, the methods have been tested by setting three different values of minimum distance (MD) between the visual features being tracked. In this case, the bigger the value, the lesser the number of visual features that can be tracked. Also, in experiments, features are removed from the system state if they are predicted to appear in the image but are not tracked in 25 periods.

Under the above conditions, Tables show the results obtained after applying the three different methods at the end of their trajectories. Some features have been computed for each method (DE, UID and UID2E) such as: i) number of the features initialized into the system state (NIF), ii) number of features deleted from the system state (NDF), iii) execution time per frame (ETF), iv) total time of execution (TTE) and v) average mean absolute error (aMAE) of the vehicle position. For computing the aMAE, the P4P trajectory has been used as an independent reference of the vehicle position (see the Appendix). However, it is important to note that the trajectory obtained by the P4P technique should not be considered as a perfect reference of groundtruth. Despite this consideration, the results obtained still reflect in a very good fashion the performance of every method.

Fig 10 shows the estimated position obtained with each method for the flight trajectories F_a and F_b. A plot for each NED coordinate (north, east and down) is given. Only the results obtained with a minimum distance between features higher than 20 pixels (MD = 20) are

Table 2. Results for flight trajectory F_b.

Method	MD(p)	NIF	NDF	ETF (s)	TTE (s)	aMAE (m)
DE	15	278 ±13σ	210 ±12σ	.58 ±.10σ	364 ±20σ	.29 ±.17σ
UID	15	319 ±11σ	244 ±13σ	.69 ±.16σ	428 ±22σ	.36 ±.16σ
UID2E	15	328 ±8σ	245 ±7σ	.65 ±.13σ	405 ±15σ	.52 ±.32σ
DE	20	185 ±11σ	140 ±8σ	.39 ±.06σ	242 ±10σ	.32 ±.20σ
UID	20	217 ±6σ	164 ±3σ	.45 ±.10σ	281 ±16σ	.34 ±.15σ
UID2E	20	220 ±4σ	167 ±2σ	.42 ±.08σ	260 ±9σ	.54 ±.36σ
DE	25	143 ±11σ	107 ±10σ	.29 ±.05σ	180 ±11σ	.31 ±.19σ
UID	25	162 ±4σ	121 ±5σ	.34 ±.07σ	213 ±9σ	.35 ±.17σ
UID2E	25	170 ±6σ	129 ±9σ	.32 ±.08σ	201 ±15σ	.52 ±.32σ

doi:10.1371/journal.pone.0167197.t002

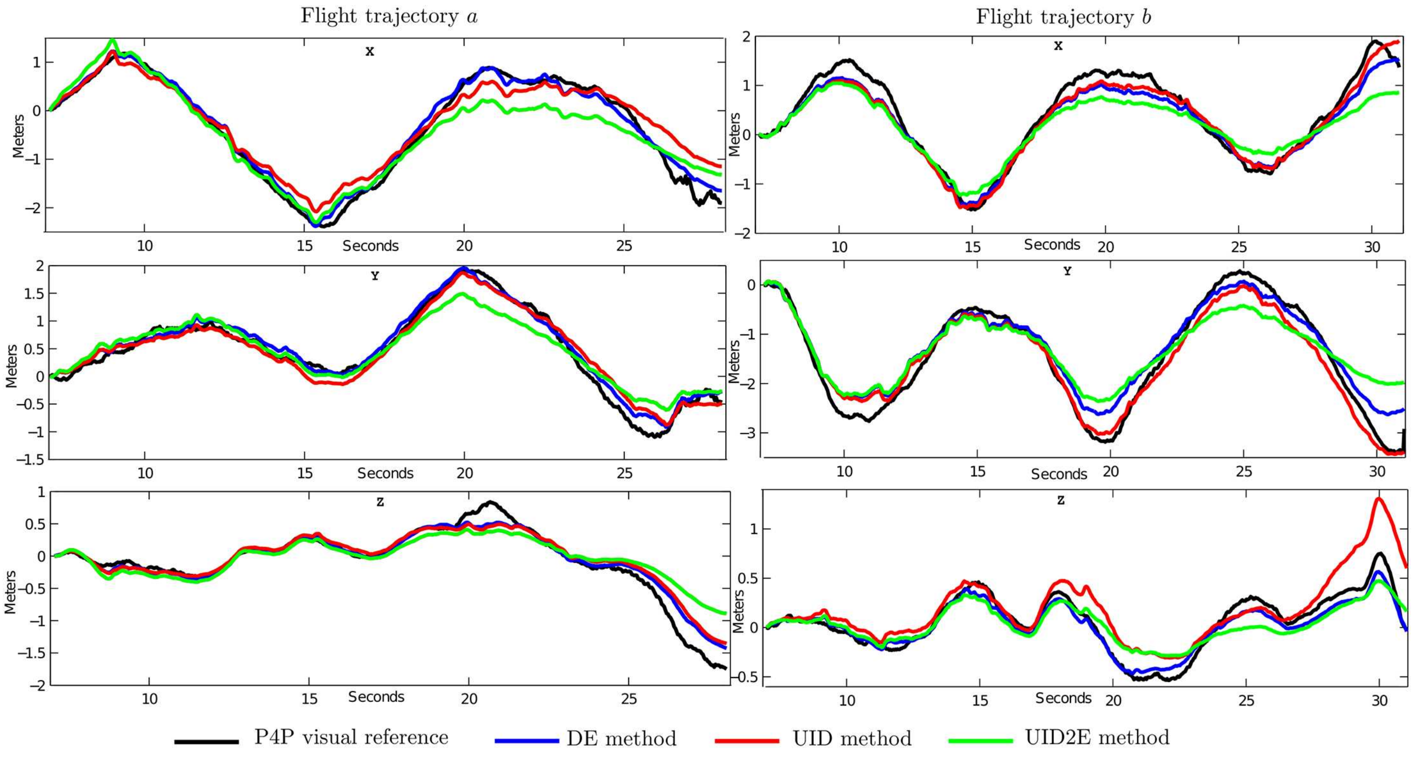

Fig 10. Comparative study of the estimated trajectory of the quadrotor obtained with: i) P4P visual reference (black); ii) DE method (blue); iii) UID method (red); and iv) UID2E method (green). Results are presented in NED coordinates: north (upper plots), east (middle plots) and down (lower plots).

doi:10.1371/journal.pone.0167197.g010

presented. Fig 11 illustrates an example of the estimated map and trajectory that have been obtained with every method. For this figure, top and lateral views are presented.

4.4 Discussion

According to the results of the comparative study some implications can be inferred. A slightly variation in the number of features, that are allowed to be tracked at each frame, can significantly affect the number of features that are initialized into the system state. In this case, a reduction of 10 pixels in the MD produces about twice of features initialized. Indeed, an increment of the features initialized into the system state implies an increment of the computational time. On the other hand, theoretically and due to the increment of information available, the increase of tracked features should improve the estimated trajectory. However, results do not show a considerable improvement in this sense. In this case, only with the trajectory F_a, a consistent but minor improvement was obtained with the increase of features, but with an increment of about twice the computational time.

Regarding to the average mean absolute error (aMAE) computed for the estimated trajectory of the quadrotor, the DE method has shown consistently slightly better results with respect to the UID method. However, it is important to note that the difference could be within the margin of error of the methodology followed for computing de aMAE. Unfortunately, statistics about this margin of error are not available. For this reason, according to the results DE method can offer at least a similar performance in accuracy with respect to the UID method. On the other hand, the UID2E method shows, in every case, the worst behaviour of all the methods. It is worth noting that, for this application, the UID2E method has exhibited a

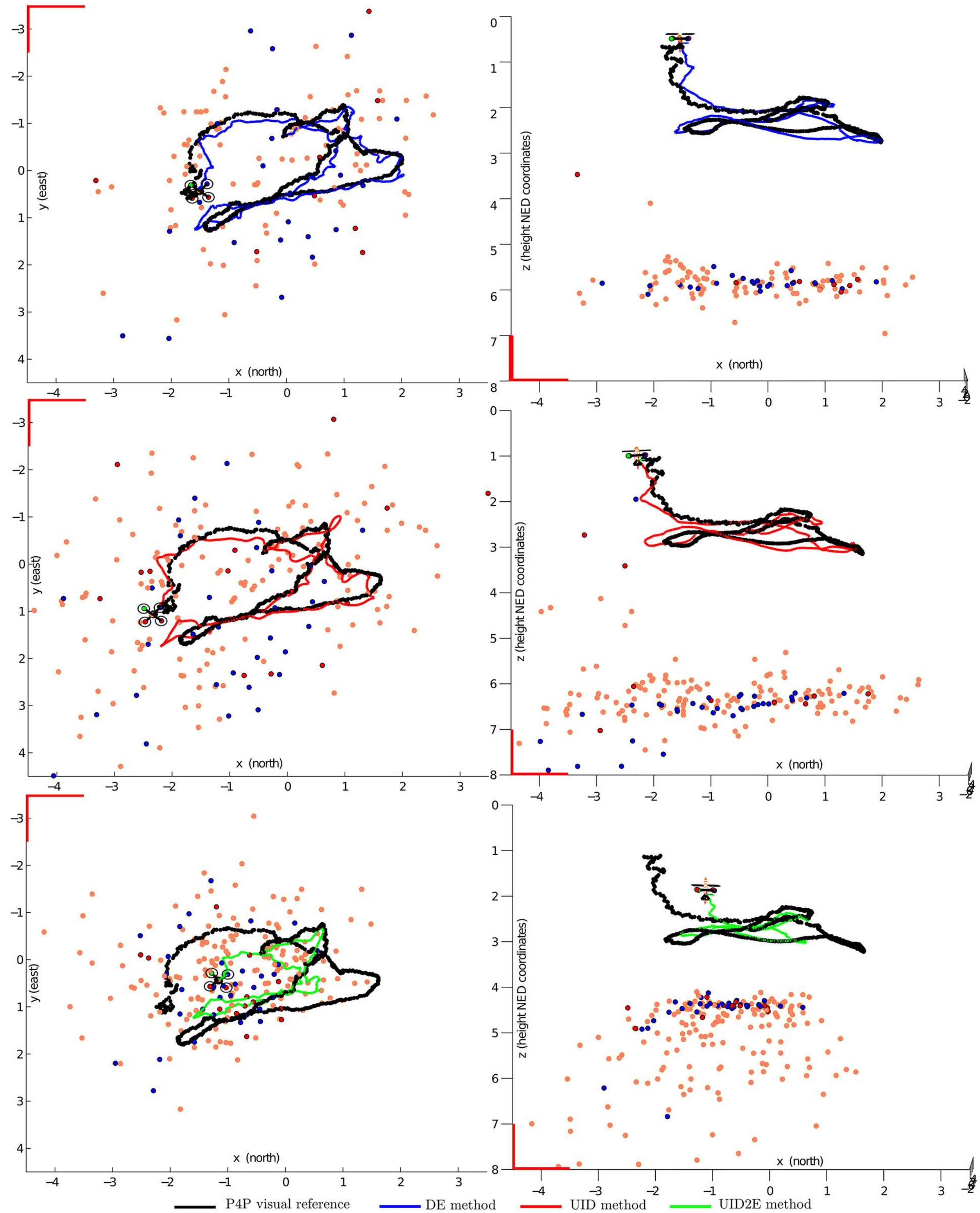

Fig 11. Comparative study of the map and trajectory estimated for the flight F_a, with: i) DE method (upper plots); ii) UID method (middle plots); and iii) UID2E method (lower plots). Only top and lateral views are shown (left and right plots respectively). In each case the P4P visual reference is indicated in black. Features deleted from the system state are indicated by small orange spheres. Features contained into the system state, at the end of the trajectory, are indicated by small blue and red spheres. Blue and red spheres mean respectively successful and unsuccessful matches.

doi:10.1371/journal.pone.0167197.g011

considerable tendency to drift in the metric scale of the estimations. In Fig 11 (lower plots), it can be clearly appreciated this phenomenon where is specially notorious the degradation in scale of the estimated map.

Regarding to the computational efficiency of the methods, it is clear that the proposed DE method presents the best results. This result can be explained for two reasons: the use of the euclidean parametrization and the use of less but stronger visual features.

In the case of the undelayed methods, the use of the inverse depth (ID) parametrization becomes mandatory due to the nonlinear nature of the measurement equation when features are initialized right after they are detected. On the other hand, ID parametrization requires six parameters instead of three euclidean ones. Therefore, as the number of features increases, with the ID parametrization the length of the state tends to have twice the length that it has with the euclidean parametrization. For the EKF-based approaches, the above ID parametrization has as consequence a well known increment in the computational cost. In this sense, the UID2E method was designed for improving the computational efficiency of the UID method. Features whose depth converge are converted from the ID to euclidean parametrization. Results validate this claim, however, for the application presented in this work, the benefit in computational efficiency is minimal compared with the increase of error drift obtained with the UID2E.

For DE method the period used for candidate points tracking is mainly intended for obtaining information about the features depth, prior to its inclusion into the system state. This fact has also the collateral benefit of pruning weak visual features that fail to be tracked in this period. In contrast to the undelayed methods (UID and UID2E), where all the detected visual features are initialized into the system, delayed methods (DE) initialize less but stronger visual features. This is more evident if the number of initialized features is considered (see Tables 1 and 2), as well as the percentage of deleted features with respect to the number of initialized features: DE = 68%, UID = 77% and UID2E = 78%. These figures mean not only that the undelayed methods initialize a lot of useless visual features, but they also mean that the features initialized with the delayed method are better retained into the system.

5 Conclusion

In this work a novel monocular SLAM method with application to a quadcopter has been presented. In this case, a monocular camera is integrated into an UAV in order to provide visual information of the ground. Due to attitude estimation is well handled by available systems for this kind of applications, this research is focused only in position estimation. In order to avoid the need of estimating the camera orientation, a servo-controlled gimbal is used for stabilizing the orientation of the camera towards the ground.

Traditionally, the position estimation of UAVs has been addressed by the use of GPS. However, the GPS is not a fully reliable service as its availability can be limited in urban canyons and is unavailable in indoor environments. Moreover, even when GPS signal is available, the problem of position estimation could not be solved for some specific scenarios, for instance in an application requiring performing precise manoeuvres in a complex environment. Therefore, some additional sensory information should be integrated into the system in order to improve accuracy and robustness. In this context, the use of monocular vision has some advantages in terms of weight, space, energy consumption, or scalability.

On the other hand, two challenging aspects related with monocular sensors have to do with the impossibility of directly recovering the depth of visual features, and the metric scale of the world as well. To address the first aspect, a novel technique for estimating the features depth based in an stochastic technique of triangulation has been presented. Regarding the second aspect, it is assumed that GPS readings are available for some short period at the beginning of the system operation. After this initial period used for incorporating information about the metric scale of the world, the system can operate relying only on visual information for estimating the position of the vehicle.

The performance of the proposed method has been validated by means of experiments with real data carried out in unstructured outdoor environments. To check the contribution of this research, an extensive comparative study is presented for validating the performance of the proposed approach respect similar methodologies. For this kind of aerial application presented in this paper, and according to the experimental results, the proposed method has performed better, in terms of accuracy and execution time, than the UID and UID2E methods.

6 Appendix

6.1 P4P reference trajectory

Experimental setups in natural outdoor environments can be a challenge for small aerial vehicles. Some difficulty arises with the absence of resources available in laboratories (e.g. Vicon system). In this particular case, for fine flight manoeuvres, the trajectory provided by the GPS is useless to be used as a reference of the actual flight trajectory. In this work, in order to have an independent reference for evaluating the performance of the proposal, the following methodology is proposed.

Four marks are placed in the ground, forming a square of known dimensions, see Fig 8. Each corner is a coplanar point with spatial coordinates $[x_i, y_i, 0]$ with $i \in 1, \ldots 4$, and their corresponding four undistorted image coordinates $[u_i, v_i]$ with $i \in 1 \ldots 4$. Then, for each frame a perspective on 4-point (P4P) technique [48], is applied iteratively in order to compute the relative position of the camera with respect to the known metric reference. At each frame, the image location of the four corners is provided by a simple tracking algorithm designed for this purpose.

The P4P technique used for estimating the camera position, defined by R^{CN} and r^N, is based on [49]. The following linear system is formed with the vector b as unknown parameter:

$$\begin{bmatrix} x_1 f & y_1 f & 0 & 0 & -u_1 x_1 & -u_1 y_1 & f & 0 \\ 0 & 0 & x_1 f & y_1 f & -v_1 x_1 & -v_1 y_1 & 0 & f \\ : & : & : & : & : & : & : & : \\ x_4 f & y_4 f & 0 & 0 & -u_4 x_4 & -u_4 y_4 & f & 0 \\ 0 & 0 & x_4 f & y_4 f & -v_4 x_4 & -v_4 y_4 & 0 & f \end{bmatrix} b = \begin{bmatrix} u_1 \\ v_1 \\ : \\ u_4 \\ v_4 \end{bmatrix} \tag{25}$$

where

$$b = \begin{bmatrix} \frac{r_{11}}{r_3} & \frac{r_{12}}{r_3} & \frac{r_{21}}{r_3} & \frac{r_{22}}{r_3} & \frac{r_{31}}{r_3} & \frac{r_{32}}{r_3} & \frac{r_1}{r_3} & \frac{r_2}{r_3} \end{bmatrix}^T \tag{26}$$

The linear system represented in Eq 26 is solved for $b = [b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5 \quad b_6 \quad b_7 \quad b_8]^T$. The camera position is computed from:

$$R^{CN} = \begin{bmatrix} r_3 b_1 & r_3 b_2 & (R_{21}R_{32} - R_{31}R_{22}) \\ r_3 b_3 & r_3 b_4 & (R_{31}R_{12} - R_{11}R_{32}) \\ r_3 b_5 & r_3 b_6 & (R_{11}R_{22} - R_{21}R_{12}) \end{bmatrix} \qquad r^N = [\, r_3 b_7 \quad r_3 b_8 \quad r_3 \,]^T \tag{27}$$

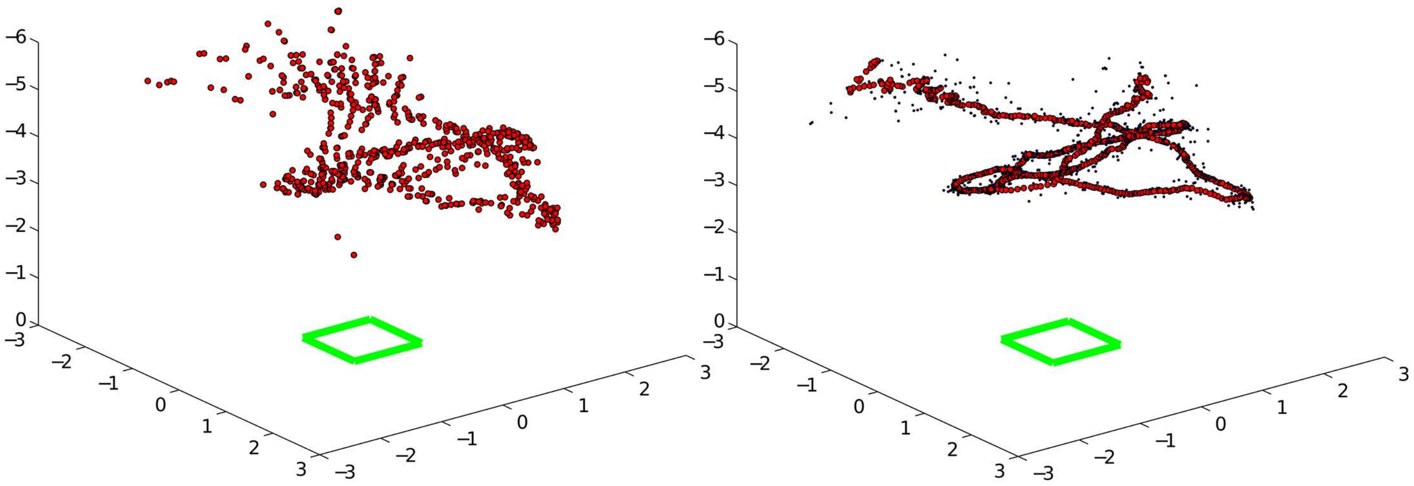

Fig 12. Trajectory obtained by the P4P technique without any filtering (left) and low-pass filtered (right).

doi:10.1371/journal.pone.0167197.g012

where

$$r_3 = \sqrt{\frac{f^2}{b_1^2 + b_3^2 + f^2 b_5^2}} \tag{28}$$

In Eq 27 the third column of matrix R^{CN} is formed by the combinations of the values of first and second column of the same matrix. The results obtained with the above procedure can be very noisy, (see left plot of Fig 12). For this reason, a simple lowpass filter is applied in order to obtain the flight trajectory (right plot, Fig 12).

The P4P trajectory is computed with respect to the metric reference. Trajectories obtained through visual SLAM have their own reference frame. In experiments, both reference frames are aligned in order to make the trajectories coincident at the beginning. In other words, it is assumed that the initial position of the quadcopter is known.

Acknowledgments

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

Author Contributions

Conceptualization: RM.

Data curation: RM SU.

Formal analysis: RM SU AG.

Funding acquisition: AG.

Investigation: RM SU AG.

Methodology: RM SU.

Project administration: AG.

Resources: RM AG.

Software: RM SU.

Supervision: RM AG.

Validation: RM SU.

Visualization: RM AG.

Writing – original draft: RM AG.

Writing – review & editing: RM SU AG.

References

1. Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part I. Robotics Automation Magazine, IEEE. 2006 june; 13(2):99–110. doi: 10.1109/MRA.2006.1638022
2. Bailey T, Durrant-Whyte H. Simultaneous localization and mapping (SLAM): part II. Robotics Automation Magazine, IEEE. 2006 sept; 13(3):108 –117. doi: 10.1109/MRA.2006.1678144
3. Cole DM, Newman PM. Using laser range data for 3D SLAM in outdoor environments. In: Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on; 2006. p. 1556–1563.
4. Zhao H, Chiba M, Shibasaki R, Shao X, Cui J, Zha H. SLAM in a dynamic large outdoor environment using a laser scanner. In: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on; 2008. p. 1455–1462.
5. Bosse M, Roberts J. Histogram Matching and Global Initialization for Laser-only SLAM in Large Unstructured Environments. In: Robotics and Automation, 2007 IEEE International Conference on; 2007. p. 4820–4826.
6. Fallon MF, Folkesson J, McClelland H, Leonard JJ. Relocating Underwater Features Autonomously Using Sonar-Based SLAM. Oceanic Engineering, IEEE Journal of. 2013 July; 38(3):500–513. doi: 10.1109/JOE.2012.2235664
7. Yap TN, Shelton CR. SLAM in large indoor environments with low-cost, noisy, and sparse sonars. In: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on; 2009. p. 1395–1401.
8. Newman P, Leonard J. Pure range-only sub-sea SLAM. In: Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on. vol. 2; 2003. p. 1921–1926 vol.2.
9. Luo RC, Huang CH, Huang CY. Search and track power charge docking station based on sound source for autonomous mobile robot applications. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on; 2010. p. 1347–1352.
10. Munguía R, Grau A. Single Sound Source SLAM. In: Ruiz-Shulcloper J, Kropatsch W, editors. Progress in Pattern Recognition, Image Analysis and Applications. vol. 5197 of Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2008. p. 70–77. Available from: http://dx.doi.org/10.1007/978-3-540-85920-8_9.
11. Kleiner A, Dornhege C, Dali S. Mapping disaster areas jointly: RFID-Coordinated SLAM by Hurnans and Robots. In: Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on; 2007. p. 1–6.
12. Hahnel D, Burgard W, Fox D, Fishkin K, Philipose M. Mapping and localization with RFID technology. In: Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on. vol. 1; 2004. p. 1015–1020 Vol.1.
13. Lemaire T, Berger C, Jung IK, Lacroix S. Vision-Based SLAM: Stereo and Monocular Approaches. International Journal of Computer Vision. 2007; 74(3):343–364. Available from: http://dx.doi.org/10.1007/s11263-007-0042-3.
14. Davison AJ, Reid ID, Molton ND, Stasse O. MonoSLAM: Real-Time Single Camera SLAM. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 2007 june; 29(6):1052 –1067. doi: 10.1109/TPAMI.2007.1049
15. Strasdat H, Montiel JMM, Davison AJ. Real-time monocular SLAM: Why filter? In: Robotics and Automation (ICRA), 2010 IEEE International Conference on; 2010. p. 2657 –2664.
16. Huh S, Shim DH, Kim J. Integrated navigation system using camera and gimbaled laser scanner for indoor and outdoor autonomous flight of UAVs. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2013. p. 3158–3163.

17. Friedman C, Chopra I, Potyagaylo S, Rand O, Kanza Y. Towards Model-Free SLAM Using a Single Laser Range Scanner for Helicopter MAV. In: AIAA Guidance, Navigation, and Control Conference; 2011.

18. Magree D, Johnson EN. Combined laser and vision-aided inertial navigation for an indoor unmanned aerial vehicle. In: 2014 American Control Conference; 2014. p. 1900–1905.

19. Tan W, Liu H, Dong Z, Zhang G, Bao H. Robust monocular SLAM in dynamic environments. In: Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on; 2013. p. 209–218.

20. Engel J, Schöps T, Cremers D. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. LSD-SLAM: Large-Scale Direct Monocular SLAM. Cham: Springer International Publishing; 2014. p. 834–849. Available from: http://dx.doi.org/10.1007/978-3-319-10605-2_54.

21. Mur-Artal R, Montiel JMM, Tardós JD. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics. 2015 Oct; 31(5):1147–1163. doi: 10.1109/TRO.2015.2463671

22. Artieda J, Sebastian J, Campoy P, Correa J, Mondragón I, Martínez C, et al. Visual 3-D SLAM from UAVs. Journal of Intelligent and Robotic Systems. 2009; 55:299–321. Available from: http://dx.doi.org/10.1007/s10846-008-9304-8.

23. Montiel JMM, Civera J, Davison A. Unified inverse depth parametrization for monocular SLAM. In: Proceedings of the Robotics: Science and Systems Conference; 2006.

24. Caballero F, Merino L, Ferruz J, Ollero A. Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs. Journal of Intelligent and Robotic Systems. 2009; 54(1–3):137–161. Available from: http://dx.doi.org/10.1007/s10846-008-9257-y.

25. Wang C, Wang T, Liang J, Chen Y, Zhang Y, Wang C. Monocular visual SLAM for small UAVs in GPS-denied environments. In: Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on; 2012. p. 896–901.

26. Suzuki T, Amano Y, Hashizume T. Development of a SIFT based monocular EKF-SLAM algorithm for a small unmanned aerial vehicle. In: SICE Annual Conference (SICE), 2011 Proceedings of; 2011. p.1656–1659.

27. Celik K, Somani AK. Monocular Vision SLAM for Indoor Aerial Vehicles. Journal of Electrical and Computer Engineering. 2013.

28. Montemerlo M, Thrun S, Koller D, Wegbreit B. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In: In Proceedings of the AAAI National Conference on Artificial Intelligence. AAAI; 2002. p. 593–598.

29. Weiss S, Scaramuzza D, Siegwart R. Monocular SLAM based navigation for autonomous micro helicopters in GPS-denied environments. Journal of Field Robotics. 2011; 28(6):854–874. Available from: http://dx.doi.org/10.1002/rob.20412.

30. Klein G, Murray D. Parallel Tracking and Mapping for Small AR Workspaces. In: Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on; 2007. p. 225 –234.

31. Nutzi G, Weiss S, Scaramuzza D, Siegwart R. Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM. Journal of Intelligent & Robotic Systems. 2011; 61:287–299. Available from: http://dx.doi.org/10.1007/s10846-010-9490-z.

32. Ta DN, Ok K, Dellaert F. Monocular Parallel Tracking and Mapping with Odometry Fusion for MAV Navigation in Feature-lacking Environments. In: IEEE/RSJ IROS'13 International Workshop on Vision-based Closed-Loop Control and Navigation of Micro Helicopters in GPS-denied Environments. Tokyo, Japan; 2013. Available from: http://rpg.ifi.uzh.ch/docs/IROS13workshop/Nguyen.pdf.

33. Forster C, Pizzoli M, Scaramuzza D. SVO: Fast semi-direct monocular visual odometry. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on; 2014. p. 15–22.

34. Munguía R, Grau A. Monocular SLAM for Visual Odometry: A Full Approach to the Delayed Inverse-Depth Feature Initialization Method. Mathematical Problems in Engineering. 2012; 2012.

35. Mirzaei FM, Roumeliotis SI. A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation. Robotics, IEEE Transactions on. 2008 Oct; 24(5):1143–1156. doi: 10.1109/TRO.2008.2004486

36. Forster C, Lynen S, Kneip L, Scaramuzza D. Collaborative monocular SLAM with multiple Micro Aerial Vehicles. In: IROS. IEEE; 2013. p. 3962–3970. http://dblp.uni-trier.de/db/conf/iros/iros2013.html#ForsterLKS13.

37. Munguía R, Grau A. A Practical Method for Implementing an Attitude and Heading Reference System. International Journal of Advanced Robotic Systems. 2014; Vol.11.

38. Euston M, Coote P, Mahony R, Kim J, Hamel T. A complementary filter for attitude estimation of a fixed-wing UAV. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on; 2008. p. 340–345.

39. Bouguet JY. Camera calibration toolbox for Matlab. In: online; 2008. Available from: http://www.vision.caltech.edu/bouguetj/calib_doc.

40. Shi J, Tomasi C. Good features to track. In: Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on; 1994.

41. Davison AJ, Murray DW. Mobile robot localisation using active vision. In: Proceedings of the 5th European Conference on Computer Vision (ECCV 98); 1998.

42. Munguía R, Grau A. Closing Loops With a Virtual Sensor Based on Monocular SLAM. Instrumentation and Measurement, IEEE Transactions on. 2009 Aug; 58(8):2377 –2384. doi: 10.1109/TIM.2009.2016377

43. Parkinson BW. Global Positionig System: Theory and Applications. American Institute of Aeronautics and Astronautics; 1996.

44. Community OS. Ardupilot; 2015. Available from: www.ardupilot.com.

45. Community OS. Ardupilot; 2015. Available from: http://qgroundcontrol.org/mavlink.

46. Civera J, Davison AJ, Montiel JMM. Inverse Depth to Depth Conversion for Monocular SLAM. In: Robotics and Automation, 2007 IEEE International Conference on; 2007. p. 2778–2783.

47. Civera J, Grasa OG, Davison AJ, Montiel JMM. 1-point RANSAC for EKF-based Structure from Motion. In: Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on; 2009. p. 3498–3504.

48. Chatterjee C, Roychowdhury VP. Algorithms for coplanar camera calibration. Machine Vision and Applications. 2000; 12:84–97. doi: 10.1007/s001380050127

49. Ganapathy S. Decomposition of transformation matrices for robot vision. In: Robotics and Automation. Proceedings. 1984 IEEE International Conference on. vol. 1; 1984. p. 130–139.