This is a postprint version of the following published document:

Al-Kaff, Abdulla, Armingol, José María, and de La Escalera, Arturo (2019). A Vision-based Navigation System for Unmanned Aerial Vehicles (UAVs). *Integrated Computer-Aided Engineering*, 26(3), pp. 297 – 310.

DOI: <https://dx.doi.org/10.3233/ICA-190601>

© The authors and IOP Press

A Vision-Based Navigation System for Unmanned Aerial Vehicles (UAVs)

Abdulla Al-Kaff^a, José María Armingol^a and Arturo de La Escalera^{a,*}

^a *Intelligent Systems Lab (LSI) Research Group, Universidad Carlos III de Madrid, Leganés, Madrid, Spain*
E-mails: akaff@ing.uc3m.es, armingol@ing.uc3m.es, escalera@ing.uc3m.es

Abstract. The advances in microelectronics foster the Unmanned Aerial Vehicles (UAVs) to be used in many civil and academic applications that require higher levels of autonomy and flight stabilization. The main objective in this paper is to provide UAVs with a robust navigation system; in order to allow the UAVs to perform complex tasks autonomously and in real-time. The proposed algorithms deal with solving the navigation problem for outdoor and indoor environments, mainly based on visual information that is captured by monocular cameras. This paper covers the topics of Pose Estimation, Navigation Guidance, and Visual Servoing. All the proposed algorithms have been verified with real flights in both indoor and outdoor environments, taking into consideration the visual conditions; such as illumination and textures. The obtained results have been validated against other systems; such as VICON motion capture system, DGPS in the case of pose estimate algorithm. In addition, the proposed algorithms have been compared with several previous works in the state of the art, and the results prove the improvement in the accuracy and the robustness of the proposed algorithms.

Keywords: Navigation Systems, Pose Estimation, Unmanned Aerial Vehicles, Artificial Vision, Visual Servoing

1. Introduction

The field of Unmanned Aerial Vehicles (UAVs) has been mainly limited to the defense and military industries, this is because of the high cost and the complexity of constructing these vehicles. Recently, with the developments in microelectronics, the field of UAVs has encountered a significant focus in several civil and academic applications; such as rescue operations [1], data collection and geophysics exploration [2], inspections [3], and agricultural purposes [4].

Because of the behavior of the missions, and the complexity of the maneuvers required, there is a necessity of increasing its the capabilities of their navigational systems with high levels of robustness and accuracy. The most known navigation systems are based on the Global Positioning System (GPS) [5]. However, the accuracy depends on the number of satellites connected. This number can be insufficient due to buildings, mountains, or nature structure [6]. Furthermore,

in other scenarios (i.e., indoor flying), GPS loses its efficiency because of the absence of information.

Different approaches are used to improve the GPS-based localization. GPS/INS is one of the most popular approaches, at which the data of the Inertial Navigation System (INS) and the GPS are fused to minimize the position error [7]. This approach has two main drawbacks, the dependency on external satellite signals and the lack of precision of the estimated position. A recent survey cover the various navigational systems for UAVs has been presented in [8].

Autonomous navigational systems in unknown and GPS-denied environments are still an open research problem and it is getting more challenging with the Micro Aerial Vehicles (MAVs) due to the limitation in the power and payload constraints. Therefore, using visual sensors such as cameras for the navigation purposes is often preferable.

The problem of understanding the world in its three dimensions using visual sensors is very challenging and critical. Different researchers tend to work with

---

*Corresponding author. E-mail: escalera@ing.uc3m.es.

stereo cameras such as Bumblebee series¹ or ZED² to estimate the 3D information of the surrounding environment. However, using stereo cameras on MAVs constrains the length of the baseline which in turn reduce the accuracy of depth estimation.

Another proposed solution is working with RGB-D sensors; such as Microsoft Kinect V2³ to deal with the problem of the depth estimation [9]. These sensors work well in indoor environments, because these sensors are based on the projection and analyzing a speckle pattern of the IR, which faults and provides unreliable information in outdoors.

Monocular cameras are lightweight sensors, consume low power and are relatively cheap. In addition, they provide rich information of the environment. Therefore, visual sensors are used in infinite number of applications such as motion capture systems [10, 11] or industrial applications [12–16]. From this point of view, and as an extension to prior work presented in [17], this paper presents an architecture to solve the UAV navigation problem in indoors and outdoors; based mainly on monocular vision algorithms, by designing and implementing a robust and efficient positioning system, covering three main topics (Pose Estimation, Navigation Guidance, and Control); to provide a robust navigation system for autonomous UAVs.

The paper is structured as follows: Section 2 presents the state-of-the art of the vision-based navigation systems. In section 3, the general overview of the proposed system is introduced. In section 4, the pose estimation algorithm is presented; focusing on the homography formulations, then in section 5, the waypoint tracking strategy is explained. Section 6 shows the implementation of the Fuzzy Logic Controller (FLC). Section 7 discusses the experimental results. Finally, conclusions are summarized in section 8.

2. Related Work

Pose estimation is the process of estimating the position and orientation of the vehicle based on the data generated by one or more sensors. Visual pose estimation methods in particular, are based on the information provided by the cameras (single camera [18], or multi-cameras [19, 20]).

Blosch *et al.* used a downward looking camera on the Hummingbird quadcopter for a vision based ap-

proach for localization [21]. The pose was estimated using the visual SLAM algorithm, and then a linear quadratic Gaussian control design with loop transfer recovery (LQG/LTR) is applied to stabilize the vehicle at a desired setpoints. Although this algorithm showed good results for hovering mode, the authors did not validate their algorithm in real flights. Furthermore, the use of SLAM algorithm increases of the computational time comparing to visual odometry algorithms.

Another monocular camera-based navigation system for an autonomous quadcopter was presented in [22], to determine only the UAV yaw and vertical speed. One of the limitations of this method is that the UAV can operate only along paths that has been traveled during a human-guided training run. Moreover, these paths can be composed only from straight-line segments with a limited length.

Different approaches based on optical flow have been presented. In [23], Lucas-Kanade method is used on the images captured from single camera in order to estimate the 2D UAV motion. However, this system was verified in a laboratory setup using a fixed rig platform. Whereas in [24], an optical flow-IMU control system has been implemented for hovering and landing maneuvers. However, the main limitation of optical flow algorithms is the sensibility against long-distances motions, which makes it suitable for small motions or hovering maneuvers.

Autonomous UAVs are developed by means of advanced control algorithms in order to perform complex missions. The autopilots are improving the on-board navigations systems to reach autonomous UAVs, where sophisticated set of fuzzy control rules deals with uncertainty in internal state and external environment [25]. Thus, a UAV controller is able to take knowledge to effectively manage flight autonomously.

Intelligent systems are focusing on new techniques; such as neural fuzzy controllers [26] from initial works using Takagi-Sugeno (TS) fuzzy rules [27]. However, recent autonomous UAVs are based on flight control by adjusting the roll, the elevator and throttle [28]. Moreover, fuzzy control systems allow full performance of complex systems, such as object transportation [29], or quadrotor manipulation system [30]. However, these controllers have drawbacks like the large amount of parameters to be tuned, or the necessity of retuning parameters. These disadvantages appear mainly due to plant parameter variations or environmental disturbances, where these problems are overcome by learning control algorithms.

¹www.ptgrey.com

²www.stereolabs.com/

³www.xbox.com/en-US/xbox-one

Fig. 1. Overall UAV Architecture [31]

3. System Overview

As shown in Figure 1, the architecture of the vision-based navigation system consists of three topics:

- **Monocular Vision-based Pose Estimation:** to localize the UAV in 3D by solving the homographies *world-to-frame* and *frame-to-frame*, using monocular images.
- **Navigation Guidance** presents the strategy of following and swapping the waypoints efficiently.
- **Visual Servoing** Implements FLC based on vision data combined with data from IMU and GPS.

4. Pose Estimation

UAV pose estimation strategy is performed by capturing new frame each time, and then estimates its pose

by tracking the feature points between two consecutive frames, and calculating the *world-to-frame* and *frame-to-frame* homographies, as shown in Figure 2.

Cameras usually work with the exposure and contrast properties to deal with the light effects. CMOS cameras deal with a high range of contrast in the image; and there is no optimal solution for adapting the image capturing with illumination that requires short or longer exposure. Therefore, as shown in Figure 3, it is important to correct the illumination of the images before performing any processing. In order to achieve this, the first step is to expand the range of the image intensities; applying histogram equalization technique:

$$I_{equihist} = \frac{L(cdf_i - cdf_{min}) - (N \times M)}{N \times M} \quad (4.1)$$

Fig. 2. Homography-based Pose Estimation Algorithm

where $I_{equihist}$ is the equalized image, L is the number of gray levels, cdf_i is the cumulative frequency of the histogram array $hist$, $cdf_i = cdf_{i-1} + hist_i$, cdf_{min} is the minimum non-zero value of the cumulative distribution function, and (N, M) are the image dimensions.

The second step is controlling the overall brightness of $I_{equihist}$; using *Gamma Correction* method:

$$I_{cor} = I_{equihist}^{\frac{1}{\gamma}} \quad (4.2)$$

(a) Original (b) Corrected

Fig. 3. Example of Illumination Corrected Images.

According to Figure 2, in each frame, a number of reliable keypoints are detected and tracked. Based on the results obtained in our previous works [17, 32], SIFT-FREAK combination is used for keypoint detection. This is because this combination provides high accuracy detection, and low computational time; as shown in Table 1. Then, Brute-Force algorithm is applied to match these keypoints as shown in Figure 4.

These keypoints usually contain outliers due to poor images, or changes in illumination. These outliers can cause a significant error in the motion estimation,

Table 1
Accuracy and Average Time (per feature point) of the detection [17]

Algorithm	Accuracy (%)			Time (ms)
	Noise	Rotation	Scale	
SIFT	45.7	64.7	10.9	3.32
SIFT-FREAK	92.8	97.2	30.0	0.32
SURF	12.8	37.8	16.1	0.52
SURF-FREAK	5.7	5.5	5.9	0.23

Fig. 4. Feature Matching: red points in F_i and blue points in F_{i-1}

therefore, a RANSAC [33] is used to remove the effects of these outliers.

4.1. Homographies Extraction

Let a set of points $q_{i-1} \in \mathbb{R}^2$ from the frame F_{i-1} related to the corresponding points $q_i \in \mathbb{R}^2$ from F_i , then the *frame-to-frame* motion in the image plane can be extracted from the homography as defined in (4.3):

$$q_i = {}^i H_{i-1} q_{i-1} \quad (4.3)$$

where ${}^i H_{i-1}$ is the *frame-to-frame* homography matrix. From 4.3, two linear system equations of the elements of ${}^i H_{i-1}$ can be obtained:

The 3D position in the world plane is similarly mapped to an image point q_i by using the projection matrix P^i as follows:

$$\lambda q_i = P^i q_w = C [R^i | t^i] q_w \quad (4.4)$$

where λ is the homogeneous scale factor, C is the camera calibration matrix, R^i and t^i are the rotation matrix and translation vector respectively, and $q_w \in \mathbb{R}^3 \wedge q_w = (x_w, y_w, z_w, 1)^T$. Assuming q_w lies on a plane $\mathcal{P}$ with $z_w = 0$, then (4.4) is rewritten as follows:

$$\lambda q_i = \hat{P}^i q_{\mathcal{P}} = C [R^i | t^i] q_{\mathcal{P}} \quad (4.5)$$

where $\hat{P}^i$ is a projection matrix which maps the points in the $\mathcal{P}$ plane to the image plane and can be defined as a planar homography, where:

$${}^i H_w = sC [r_1^i \ r_2^i \ t^i] = \hat{P}^i \quad (4.6)$$

where, $s = \frac{1}{\lambda}$ is an arbitrary scale factor.

However, ${}^i H_w$ cannot be calculated directly, therefore, it is required to calculate the initial homography ${}^0 H_w$ of the reference frame (*world-to-reference frame*):

$${}^0 H_w = C [{}^0 r_1 \ {}^0 r_2 \ t^0] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \zeta \end{bmatrix} \quad (4.7)$$

where, ζ is the initial altitude obtained by the ultrasonic sensor or it can be estimated from the size of a detected predefined pattern in the start point on the navigation using the pinhole model equation (4.8) and shown in Figure 5.

$$\frac{-y_{img}}{f} = \frac{\zeta}{d_r} \quad (4.8)$$

where y_{img} is the object height in the image plane, f is the focal length of the camera, and d_r is the distance between the camera optical center and the object.

Fig. 5. Example of the altitude estimation by detecting a known-size pattern (Pattern Size: 891×891 px, Estimated Altitude: 2.791 m).

Thereafter, the related i^{th} frame homographies can be estimated as a sequence of *frame-to-frame* homographies multiplication as follows:

$${}^i H_w = {}^i H_{i-1} {}^{i-1} H_{i-2} \dots {}^1 H_0 {}^0 H_w \quad (4.9)$$

4.2. Homography Decomposition

${}^i H_w$ is decomposed to its rotation and translation by defining the scale factor $\lambda = \frac{1}{s}$ as follows:

$$\begin{bmatrix} {}^i r_1 & {}^i r_2 & {}^i r_3 & t^i \end{bmatrix} = \lambda C^{-1} {}^i H_w \quad (4.10)$$

where $\lambda = \frac{1}{\|C^{-1} h_1\|} = \frac{1}{\|C^{-1} h_2\|}$, and

$$\begin{aligned} r_1 &= \lambda C^{-1} h_1, & r_2 &= \lambda C^{-1} h_2 \\ r_3 &= r_1 \times r_2, & t &= \lambda C^{-1} h_3 \end{aligned} \quad (4.11)$$

The final transformation matrix of rotations $R_x(\phi_{roll})$, $R_y(\theta_{pitch})$ and $R_z(\psi_{yaw})$ can be written as:

$$R_{xyz} = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\theta s_\psi c_\psi - c_\phi s_\psi & s_\theta s_\psi s_\psi + c_\phi c_\psi & s_\phi c_\psi \\ c_\phi s_\psi c_\psi + s_\phi s_\psi & c_\phi s_\psi s_\psi - s_\phi c_\psi & c_\phi c_\psi \end{bmatrix} \quad (4.12)$$

where, $c_\theta = \cos(\theta_{pitch})$, $s_\phi = \cos(\phi_{roll})$ and so on. From (4.12), the three angles are extracted as:

$$\begin{aligned} \phi_{roll} &= \tan^{-1} \left(\frac{s_\phi c_\psi}{c_\phi c_\psi} \right) \\ \theta_{pitch} &= \tan^{-1} \left(\frac{-s_\theta}{c_\phi} \right) \\ \psi_{yaw} &= \tan^{-1} \left(\frac{c_\theta s_\psi}{c_\theta c_\psi} \right) \\ c_\phi &= \sqrt{c_\theta^2 c_\psi^2 + c_\theta^2 s_\psi^2} \end{aligned} \quad (4.13)$$

4.3. Correction of Rotation Matrix

A problem appears when both $c_\theta s_\psi$ and $c_\theta c_\psi$ have very small values or equal to zero, which yields $s_\phi c_\theta$

and $c_\phi c_\theta$ to be very small or equal to zero, and $-s_\theta$ will be close to ± 1 . Then, it is difficult to estimate the angle ϕ_{roll} . However, this is solved by comparing the value of c_θ with an empiric threshold value ($\varepsilon = 1.9073e^{-6}$), and if c_θ falls below this threshold, the matrix elements are reduced to following:

$$R_{xyz} = \begin{bmatrix} 0 & 0 & -(\pm 1) \\ \pm s_\phi c_\psi - c_\phi s_\psi & \pm s_\phi s_\psi + c_\phi c_\psi & 0 \\ \pm c_\phi c_\psi + s_\phi s_\psi & \pm c_\phi s_\psi - s_\phi c_\psi & 0 \end{bmatrix} \quad (4.14)$$

This matrix leads to an example of gimbal lock, at which the x and z axes are brought into alignment by the y rotation, and loses one degree of freedom because ϕ_{roll} and ψ_{yaw} angles act as a single parameter.

Handling this case by setting one of ϕ_{roll} or ψ_{yaw} to zero, allows to derive the other angle, and thus reduces the matrix to the following form:

$$R_{xyz} = \begin{bmatrix} 0 & 0 & -(\pm 1) \\ \pm s_\phi & c_\phi & 0 \\ \pm c_\phi & -s_\phi & 0 \end{bmatrix} \quad (4.15)$$

Hence, ϕ_{roll} can be easily extracted as follow:

$$\phi_{roll} = \tan^{-1}\left(\frac{-s_\phi}{c_\phi}\right) \quad (4.16)$$

ϕ_{roll} sometimes can be obtained with some confused set of values, which leads to produce abrupt changes in the angle values. However, this is not a major problem, and that is because of the close relationship between ϕ_{roll} and ψ_{yaw} as mentioned before. Therefore, any abrupt change in the value of ϕ_{roll} is easily corrected by the value of ψ_{yaw} .

On the other hand, if the value of c_θ exceeds the threshold ε , each of the angles ϕ_{roll} and ψ_{yaw} acts independently, and the rounding errors in one angle are very large and cannot be corrected by the other. However, this is solved by computing the rotation generated from the extracted ϕ_{roll} and θ_{pitch} , then working out the rotation required in ψ_{yaw} to reconstruct its matrix. This is derived by multiplying an unknown ψ_{yaw} matrix by the transpose of the product of ϕ_{roll} and θ_{pitch} matrices:

$$\dot{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{bmatrix} \begin{bmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{bmatrix}^T \begin{bmatrix} \dot{r}_{11} & \dot{r}_{12} & \dot{r}_{13} \\ \dot{r}_{21} & \dot{r}_{22} & \dot{r}_{23} \\ \dot{r}_{31} & \dot{r}_{32} & \dot{r}_{33} \end{bmatrix} \quad (4.17)$$

As the product represents a pure rotation about the z -axis, then ψ_{yaw} must be in the following form:

$$\dot{R} = \begin{bmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.18)$$

Accordingly, from (4.17) and (4.18), the ψ_{yaw} can be extracted as follows:

$$\psi_{yaw} = \tan^{-1}\left(\frac{s_\phi \dot{r}_{31} - c_\phi \dot{r}_{21}}{c_\phi \dot{r}_{22} - s_\phi \dot{r}_{32}}\right) \quad (4.19)$$

5. Waypoint Tracking

This section focuses on the UAV navigation guidance, that deals with the problem of following a pre-defined path, by tracking the generated waypoints that build this path.

Figure 6 describes the steps of following and swapping the waypoints in the tracking phase. This algorithm requires global pose vector of all reference waypoints $(x_r, y_r, z_r, \psi_r)_{1:N}$ of the path, and the UAV global current pose (x_d, y_d, z_d, ψ_d) as inputs.

First, the algorithm reads all these global reference waypoints, and the global UAV current pose as well, and determines the initial heading angle ($\psi_{initial}$), which defines the angle of the UAV in the starting point. Thereafter both poses (*reference* and *current*) are converted to the local coordinates system using the following equations:

$$\begin{aligned} x_{lc_t} &= (x_{gl_t} - x_{gl_{t-1}})\cos(\psi) - (y_{gl_t} - y_{gl_{t-1}})\sin(\psi) \\ y_{lc_t} &= (x_{gl_t} - x_{gl_{t-1}})\sin(\psi) + (y_{gl_t} - y_{gl_{t-1}})\cos(\psi) \\ \psi_{lc_t} &= \psi_{gl_t} - \psi_{initial} \end{aligned} \quad (5.1)$$

where, $(x, y)_{lc_t}$ is the current local position of the UAV, $(x, y)_{gl_t}$ and $(x, y)_{gl_{t-1}}$ are the current and previous global UAV positions, respectively, ψ is the heading angle that are defined as:

$$\psi = \psi_{gl_{initial}} - \psi_{lc_{initial}} \quad (5.2)$$

and ψ_{lc_t} is the current local heading angle.

In the next step, the algorithm checks if the waypoint has been reached or not; by comparing the current pose of the UAV, and the waypoint, taking in consideration some tolerance values, which are $\pm 10\%$ of

Fig. 6. Waypoint Tracking Algorithm.

the value of the heading angle and an area with a radius of 25 cm around the waypoint, where:

$$\begin{aligned} x_{uav_t} &= x_{waypoint_t} \pm 25cm \\ y_{uav_t} &= x_{waypoint_t} \pm 25cm \\ \psi_{uav_t} &= \psi_{waypoint_t} \pm 0.1 \times \psi_{waypoint_t} \end{aligned} \quad (5.3)$$

Once the UAV is located within the waypoint tolerance area, the algorithm seeks the next waypoint. In the case if the UAV does not reach the waypoint, the algorithm estimates the status and the direction of the UAV, as it is shown in Table 2. In which, the UAV performs thirteen possible maneuvers with different navigation directions; that are estimated by comparing the position and the heading angle of the waypoint with the UAV pose; for example, if the reference waypoint position in all the axes is greater than UAV position, and the difference in the heading angle between the waypoint and the UAV is negative, thereafter, the UAV performs a Turn-Left 'TL' and Upward 'U' maneuvers coupled with the motion in the Forward-Left direction 'FL', and so on.

Finally, after estimating the navigation direction of the UAV, the fuzzy controller is applied to send the

flight commands. This process keeps tracking the waypoints till the end of the size of the global pose vector of the reference waypoints. After reaching the last point, the UAV performs an autonomous landing.

Table 2
Possible Navigation Directions

Direction	Angle	Enumerate
Forward	pitch -	F
Backward	pitch +	B
Slide-Left	roll -	SL
Slide-Right	roll +	SR
Turn-Left	yaw -	TL
Turn-Right	yaw +	TR
Forward Left	pitch - roll -	FL
Forward Right	pitch - roll +	FR
Backward Left	pitch + roll -	BL
Backward Right	pitch + roll +	BR
Upward	throttle +	U
Downward	throttle -	D
Hover	pitch = roll = yaw = throttle = 0	H

6. Visual Servoing

The third topic deals to close the control loop; based on the information provided by the visual sensors as well as the other on-board sensors; to control the stability and the maneuvers of the UAV; as it is shown in Figure 7. In this section, the implemented control system is FLC which is based on the Mamdani method [34], that is used to control a non-linear variable; for example, the *roll* or *pitch* angle of a UAV.

The numeric values of the output variable are obtained through the global mass center; to find the crisp single output value in the defuzzification step. The weight of the output is estimated as follows:

$$\sum_{i=1}^n z_i \prod \mu(z_i) \quad (6.1)$$

The notations that are used in the rule base to represent the input and output values are defined in Table 3.

Table 3
Fuzzy rule base notation

Notation	Representation
NB	Negative-Big
NM	Negative-Medium
N	Negative
Z	Zero
P	Positive
PM	Positive-Medium
PB	Positive-Big

Fig. 7. UAV Fuzzy Control System

Fig. 8. Membership functions of Altitude and Yaw controllers.

6.2. Stability Fuzzy-Controller

This subsection presents the controller of the stability mode, where, *pitch* and *roll* angles are controlled to maintain the UAV in a hover mode, with zero x - y displacement. Two rule sets have been created (one for each angle). The angles values are automatically adjusted; in order to maintain the position of the UAV with horizontal velocities V_x and V_y equal to zero. The angles and the velocities are estimated from the on-board IMU without considering any positioning information from the vision system. Thus, for each angle output control, two inputs are defined (horizontal velocity and its corresponding angle).

The controller is developed based on five sets for each input, using trapezoidal memberships functions. It has been found that using the trapezoidal functions provides more stability in the motion of controlling the horizontal velocities and (roll, pitch) angles than the Gaussian functions. Figure 9 shows the membership functions of pitch angle.

The controller is achieved by the fuzzy rule-base matrix, which is composed of two decision tables of 25 rules for each angle; which consists of direct rules. For example if the velocity in x -axis is NB and the pitch angle is Z, then, the control command is NB, and so on. The reason of using two decision tables is because that the rules are different for each angle, this is because

Fig. 9. Membership functions of Pitch Control in Stability mode.

6.3. Position Fuzzy-Controller

This controller is designed to maintain the position of the UAV (*Hover mode*) or to track and follow the position of the predefined waypoints.

Similarly to the stability mode, the controller has the angles and horizontal velocities from the IMU sensor as inputs. However, the changes in the distance have been added as a new input to the controller.

The distance has been calculated as the difference between the actual UAV position to the hovering point position, or to the waypoint position. The new x - y position of the UAV is estimated using the vision-based algorithm.

The fuzzy rule-base matrix of the angles consists of 75 rules, based on the three inputs and one output. Fig-

(c) Δx -distance input

Fig. 10. Membership functions of Pitch Control in Position mode.

7. Experimental Results

The proposed algorithms were tested and verified with data gathered from real flights in outdoors (open space $100\text{m} \times 100\text{m}$) and indoors (room of dimensions $10\text{m} \times 8\text{m} \times 5\text{m}$), without obstacles and with different visual conditions; such as ground texture and illumination.

7.1. Platform

The experiments were performed using a quadcopter with *Pixhawk* control system as shown in Figure 11. The quadcopter is equipped with SJ4000 wireless camera which provides a resolution of 640×480 color images, and mounted on 2D gimbal to provide stability of the camera. Furthermore, the processing is performed in Intel i7-3770 at 3.4 GHz CPU, with 6 GB DDR3 RAM embedded system.

Fig. 11. Experimental Platform

7.2. Results

7.2.1. Visual Odometry

Outdoors: the flights are performed following a pre-defined path (ground truth), with altitudes between 1 m to 4 m. The results obtained from the visual system are compared with the data of the DGPS system.

Figure 12 shows an example on one of the outdoor experiments, in which, the UAV performs a random-shape flight with total distance on 132.3 m. In this experiment, it is observed that the DGPS loses its accuracy from the beginning because the *Start* point is close to a building wall. In addition, it provides an imprecise signal data with error of 75 cm in a comparison to the ground truth. In contrast, the vision system was able to follow the path and keep the position as the ground truth. The total error obtained by the vision system is 50 cm while the DGPS error is 110 cm.

From the outdoor experiments, the maximum error generated by the vision system is ± 50 cm in (x,y) -axis and ± 30 cm in z -axis.

Indoors: free flying movements were performed with a maximum altitude of 2.5 m, and the estimated results are compared with the results obtained by 9-cameras VICON motion capture system.

Table 4
Pitch control decision table (*Position mode*)

		Pitch Angle (<i>deg</i>)																								
		NB	N	Z	P	PB						NB	N	Z	P	PB						NB	N	Z	P	PB
X-Velocity (<i>mm/s</i>)	NB	Z	Z	N	N	NB						Z	Z	NB	NB	NB						Z	Z	NB	NB	NB
	N	P	P	Z	N	N						P	Z	N	NB	NB						P	Z	NB	NB	NB
	Z	P	P	P	Z	N						PB	P	Z	N	NB						P	Z	N	N	N
	P	PB	PB	P	Z	N						PB	PB	P	Z	N						P	P	Z	N	N
	PB	PB	PB	PB	Z	Z						PB	PB	PB	Z	Z						PB	P	P	Z	Z
$\Delta X < 0$											$\Delta X = 0$										$\Delta X > 0$					
ΔX (<i>mm</i>)																										

Figure 13a shows the estimated 2D motion by the VICON compared to the vision system. Whilst Figure 13b shows the 3D motion obtained by the two systems. From these experiments, it was found that the maximum error generated by the vision system is ± 35 cm in (x,y) -axis and ± 20 cm in z -axis.

Table 5 shows the comparison between the different flights environment with the total distance, the mean drift in each axis, the error in the final position, the standard deviation and the total accuracy. The error in the final position was calculated by assuming the maximum error, that generates an error (100 %) is located in a radius of 1 meter from the *End* point of the ground truth (Figure 14).

Moreover, Table 6 shows a comparison of the pose estimation accuracy between four approaches from the bibliography and the proposed algorithm. The results show that the best performance of the proposed algo-

Table 5
Accuracy

Environment	Indoor	Outdoor
Number of Flights	30	30
Total Distance (m)	676.5	3742.1
Mean (x,y) Drift (m)	± 0.21	± 0.38
Mean (z) Drift (m)	± 0.20	± 0.30
Mean Position Error (%)	0.68	0.53
Max Position Error (%)	1.19	1.38
Standard Deviation	0.26	0.19
Accuracy (%)	99.32	99.47

rithm reaching a 99 % of accuracy and a standard deviation of 0.23.

Fig. 12. Outdoor flight - Random shape - total distance: 132.3 m (Ground truth, Vision and DGPS).

Table 6
Comparison of Visual Odometry Algorithms

Algorithm	Altitude Range (m)	MD (m) ^a	MPE (%) ^b	SD ^c	Accuracy (%)
Wang [35]	VO	8	14.88	—	8.87
	VO-IMU	8	1.98	—	1.16
Optical Flow	Monocular [23]	1	—	3.0	2.0
	Stereo [36]	3-5	± 0.24	1.67	0.64
Proposed Algorithm	1-4	± 0.29	0.61	0.23	99.4

^a MD: Mean (x,y) Drift (m). ^b MPE: Mean Position Error (%). ^c SD: Standard Deviation.

(a) Pose estimation in 2D

(b) Pose estimation in 3D

Fig. 13. Indoor flight - total distance: 19 m (Vision and VICON).

Finally, the average of the estimated computation time is 64.2 ms. This result proved that the using SIFT-FREAK combination is faster than classical SIFT calculation, which obtains 110 ms.

7.2.2. Waypoint Tracking

Ten outdoor and indoor experiments were tested; in order to evaluate and verify the performance of the control system on the UAV, to track the predefined waypoints. The occupancy grid maps for the scenarios are generated to the scale of the real dimensions, the grid cells are divided into one square meter (1 m²). All the static obstacles are considered in the mapping pro-

cess. Subsequently, the maps are loaded into the program in order to start the path planning algorithm and to obtain the final 3D waypoints; in order to achieve the autonomous navigation.

Outdoors Figure 15a shows the performance of the UAV navigating in a path of a total distance 48.8 m. The UAV follows a straight path with different range of yaw angles circumnavigating static obstacles, maintaining the safe distance to them.

In Figure 15b, the strategy of the waypoint switching is shown; once the UAV enters the tolerance area of the waypoint (wp_8), the control system receives the position of the new waypoint (wp_9), and starts changing the direction of the UAV and performs the maneuver before reaching the waypoint. This case demonstrates the concept of the path-following priority than the waypoint-precision.

The effect of the control system in order to maintain the UAV to be in the path is shown in Figure 15b. Once the UAV deviates from its predefined path (outside the tolerance area), the control system instantly corrects the UAV motion to be inside the tolerance area of the corresponding waypoint (wp_{12}).

Indoors Figure 16a shows the performance of the UAV in navigating an open ending square shape with total distance of 20.5 m. This experiment shows the accurate performance of the UAV control system to follow a loop shape where the starting point and final point are close to each other.

The situation of a missed waypoint is shown in Figure 16b. In this case the UAV was affected by the vortices that are reflected from the wall close to it. This effect pushes the UAV far from the waypoint. It is shown that the current x-position of the UAV is located beyond the waypoint (wp_5), then the tracking algorithm looks for the next waypoint (wp_6) instead of performing backward maneuver.

Fig. 14. End point (final position) and the maximum error.

Fig. 15. Waypoint Tracking Diamond-shape - total distance: 45.5 m.

The obtained results showed the high performance of the algorithm in following the theoretical path with total error in the final point of ± 30 cm in (x,y) -direction and ± 5 cm in the z -direction, in both outdoor and indoor scenarios.

Fig. 16. Waypoint Tracking Square-shape - total distance: 20.5 m.

The obtained results showed the high performance of the algorithm in following the theoretical path with total error in the final point of ± 30 cm in (x,y) -direction and ± 5 cm in the z -direction, in both outdoor and indoor scenarios.

8. Conclusion

This paper presents a guidance system based on visual information captured by on-board monocular camera; to provide a robust navigation system for indoor and outdoor applications. The proposed solutions deal with the problems of **pose estimation, waypoint tracking, and control system** of autonomous UAVs.

The obtained results proved the accuracy of the pose estimation algorithm is 99.4 % with an overall error of 40 cm and 30 cm on the (x,y) -axes in outdoors and 21 cm and 20 cm in indoors on the (x,y) -axes, respectively.

The second algorithm gives the UAV the ability to follow the waypoints generated from the path planning algorithm with minimal error. From the experiments, the total error in the final point of ± 30 cm and ± 5 cm in outdoors, and ± 30 cm and ± 5 cm in indoors, in (x,y) -direction and z -direction respectively.

Future researches will be focused on optimizing the parallel processing techniques to allow the processing of high-resolution images, and the fusion of the obtained visual information with the data provided by the other on-board sensors.

ACKNOWLEDGMENT

Research supported by the Spanish Government through the CICYT projects (TRA2015-63708-R and TRA2016-78886-C3-1-R), and the Comunidad de Madrid through SEGVAUTO-TRIES (S2013/MIT-2713). Also, we gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

References

- [1] D. Erdos, A. Erdos and S.E. Watkins, An experimental UAV system for search and rescue challenge, *IEEE Aerospace and Electronic Systems Magazine* **28**(5) (2013), 32–37.
- [2] F. Fraundorfer, L. Heng, D. Honegger, G.H. Lee, L. Meier, P. Tanskanen and M. Pollefeys, Vision-based autonomous mapping and exploration using a quadrotor MAV, in: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 4557–4564.
- [3] A. Al-Kaff, F.M. Moreno, L.J. San José, F. García, D. Martín, A. de la Escalera, A. Nieva and J.L.M. Garcéa, Vbii-uav: vision-based infrastructure inspection-uav, in: *World Conference on Information Systems and Technologies*, Springer, 2017, pp. 221–231.
- [4] H. Salo, V. Tirronen and F. Neri, Evolutionary regression machines for precision agriculture, in: *European Conference on the Applications of Evolutionary Computation*, Springer, 2012, pp. 356–365.
- [5] M.F. Bahat and T. Filik, GPS-based antenna tracking and signal beamforming system for small UAV platform, in: *Signal Processing and Communications Applications Conference (SIU), 2015 23th*, IEEE, 2015, pp. 1977–1980.
- [6] F. Bonin-Font, C.C. Gomila and G.O. Codina, Towards Visual Navigation of an Autonomous Underwater Vehicle in Areas with Posidonia Oceanica, *Revista Iberoamericana de Automática e Informática Industrial RIAI* **15**(1) (2017), 24–35.
- [7] C. Vincenzo Angelino, V.R. Baraniello and L. Cicala, High altitude UAV navigation using IMU, GPS and camera, in: *Information Fusion (FUSION), 2013 16th International Conference on*, IEEE, 2013, pp. 647–654.
- [8] A. Al-Kaff, D. Martín, F. García, A. de la Escalera and J.M. Armingol, Survey of computer vision algorithms and applications for unmanned aerial vehicles, *Expert Systems with Applications* **92** (2018), 447–463.
- [9] J.-Y. Baek, S.-H. Park, B.-S. Cho and M.-C. Lee, Position tracking system using single RGB-D Camera for evaluation of multi-rotor UAV control and self-localization, in: *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, IEEE, 2015, pp. 1283–1288.
- [10] H. Park, H. Lee, H. Adeli and I. Lee, A new approach for health monitoring of structures: terrestrial laser scanning, *Computer-Aided Civil and Infrastructure Engineering* **22**(1) (2007), 19–30.
- [11] S. Park, H. Park, J. Kim and H. Adeli, 3D displacement measurement model for health monitoring of structures using a motion capture system, *Measurement* **59** (2015), 352–362.
- [12] K. Park, M. Torbol and S. Kim, Vision-Based Natural Frequency Identification Using Laser Speckle Imaging and Parallel Computing, *Computer-Aided Civil and Infrastructure Engineering* **33**(1) (2018), 51–63.
- [13] R. Li, Y. Yuan, W. Zhang and Y. Yuan, Unified Vision-Based Methodology for Simultaneous Concrete Defect Detection and Geolocalization, *Computer-Aided Civil and Infrastructure Engineering* (2018).
- [14] S. Wu, G. Zhang, M. Zhu, T. Jiang and F. Neri, Geometry based three-dimensional image processing method for electronic cluster eye, *Integrated Computer-Aided Engineering* (2018), 1–16.
- [15] F. Martinez, E. Pissaloux and A. Carbone, Towards activity recognition from eye-movements using contextual temporal learning, *Integrated Computer-Aided Engineering* **24**(1) (2017), 1–16.
- [16] X. Kong and J. Li, Vision-Based Fatigue Crack Detection of Steel Structures Using Video Feature Tracking, *Computer-Aided Civil and Infrastructure Engineering* (2018).
- [17] A. Al-Kaff, A. de la Escalera and J.M. Armingol, Homography-based navigation system for unmanned aerial vehicles, in: *International Conference on Advanced Concepts for Intelligent Vision Systems*, Springer, 2017, pp. 288–300.
- [18] Q. Zeng, Y. Wang, J. Liu, R. Chen and X. Deng, Integrating monocular vision and laser point for indoor UAV SLAM, in: *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2014*, IEEE, 2014, pp. 170–179.
- [19] D. Nistér, O. Naroditsky and J. Bergen, Visual odometry for ground vehicle applications, *J. Field Robotics* **23**(1) (2006), 3–20.
- [20] J. Jeong, J. Mulligan and N. Correll, Trinocular visual odometry for divergent views with minimal overlap, in: *Robot Vision (WORV), 2013 IEEE Workshop on*, IEEE, 2013, pp. 229–236.

- [21] M. Bloesch, S. Weiss, D. Scaramuzza and R. Siegwart, Vision based MAV navigation in unknown and unstructured environments, in: *Robotics and automation (ICRA), 2010 IEEE international conference on*, 2010, pp. 21–28.
- [22] T. Krajník, M. Nitsche, S. Pedre, L. Preucil and M.E. Mejail, A simple visual navigation system for an UAV, in: *International Multi-Conference on Systems, Signals and Devices. Piscataway: IEEE*, 2012, p. 34.
- [23] V. More, H. Kumar, S. Kaingade, P. Gaidhani and N. Gupta, Visual odometry using optic flow for Unmanned Aerial Vehicles, in: *Cognitive Computing and Information Processing (CCIP), 2015 International Conference on*, IEEE, 2015, pp. 1–6.
- [24] B. Herissé, T. Hamel, R. Mahony and F. Russotto, Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow, *IEEE Transactions on Robotics* **28**(1) (2012), 77–89.
- [25] Y. Qu, S. Pandhiti, K. Bullard, W. Potter and K. Fezer, Development of a genetic fuzzy controller for an unmanned aerial vehicle, in: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Springer, 2011, pp. 328–335.
- [26] L.U. Emaletdinova and A.N. Kabirowa, Neural fuzzy controller to control the angle of heel and the course of the unmanned aerial vehicle, in: *Dynamics of Systems, Mechanisms and Machines (Dynamics)*, 2016, IEEE, 2016, pp. 1–5.
- [27] E. Applebaum, Fuzzy gain scheduling for flutter suppression in unmanned aerial vehicles, in: *Fuzzy Information Processing Society, 2003. NAFIPS 2003. 22nd International Conference of the North American*, IEEE, 2003, pp. 323–328.
- [28] S. Kurnaz, O. Cetin and O. Kaynak, Fuzzy logic based approach to design of flight control and navigation tasks for autonomous unmanned aerial vehicles, *Journal of Intelligent and Robotic Systems* **54**(1–3) (2009), 229–244.
- [29] J. Estevez, M. Graña and J.M. Lopez-Guede, Online fuzzy modulated adaptive PD control for cooperative aerial transportation of deformable linear objects, *Integrated Computer-Aided Engineering* **24**(1) (2017), 41–55.
- [30] A. Khalifa and M. Fanni, A new quadrotor manipulation system: Modeling and point-to-point task space control, *International Journal of Control, Automation and Systems* (2017), 1–13.
- [31] A. Al-Kaff, F. García, D. Martín, A. De La Escalera and J.M. Armingol, Obstacle detection and avoidance system based on monocular camera and size expansion algorithm for UAVs, *Sensors* **17**(5) (2017), 1061.
- [32] A. Al-Kaff, A. de la Escalera and J.M. Armingol, SIFT and SURF Performance Evaluation and the Effect of FREAK Descriptor in the Context of Visual Odometry for Unmanned Aerial Vehicles, in: *International Conference on Computer Aided Systems Theory*, Springer, 2015, pp. 739–747.
- [33] M.A. Fischler and R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM* **24**(6) (1981), 381–395.
- [34] E.H. Mamdani and S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Man-Machine Studies* **7**(1) (1975), 1–13.
- [35] C. Wang, T. Wang, J. Liang, Y. Chen and Y. Wu, Monocular vision and IMU based navigation for a small unmanned helicopter, in: *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2012, pp. 1694–1699.
- [36] R. Strydom, S. Thurrowgood and M. Srinivasan, Visual odometry: autonomous uav navigation using optic flow and stereo, in: *Proceedings of Australasian Conference on Robotics and Automation*, 2014.
- [37] V. Grabe, H.H. Bulthoff and P. Robuffo Giordano, Robust optical-flow based self-motion estimation for a quadrotor UAV, in: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 2153–2159.
- [38] J. Zhang, S. Singh and G. Kantor, Robust Monocular Visual Odometry for a Ground Vehicle in Undulating Terrain, in: *Field and Service Robotics*, Springer, 2014, pp. 311–326.