VISIÓN ELECTRÓNICA

Preparación de Artículos revista VISIÓN ELECTRÓNICA: algo más que un estado sólido
Fecha de envío: 06 de octubre de 2019
Fecha de recepción: 17 de diciembre de 2019
Fecha de aceptación: 09 de enero de 2020

A research vision

Autonomous trajectory following for an UAV based on computer vision

Seguimiento autónomo de trayectoria para un UAV basado en visión artificial

Jorge Daniel Gallo-Sanabria [1], Paula Andrea Mozuca-Tamayo [2], Rafael Iván Rincón-Fonseca [3]

Cite this article as: J. D. Gallo-Sanabria, P. A. Mozuca-Tamayo and R. I. Rincón-Fonseca, "Autonomous trajectory following for an UAV based on computer vision", *Visión electrónica, algo más que un estado sólido*, vol. 14, no. 1, pp. xx, january-june 2020. doi: xxx

Abstract: The trajectory following performed by unmanned aerial vehicles has several advantages that can be taken to several applications, going from package delivery to agriculture. However, this involves several challenges depending on the way the following is performed, particularly in the case of trajectory following by using computer vision. In here we will show the design, the simulation and the implementation of a simple algorithm for trajectory following by using computer vision, this algorithm will be executed on a drone that will arrive into a wished point.

Keywords: Computer vision, control law, thresholding, trajectory following, UAV.

Resumen: El seguimiento de trayectorias por parte de vehículos aéreos no tripulados trae consigo varias ventajas que pueden llevarse a aplicaciones que van desde la mensajería hasta

[1] BSc. in Mechatronics engineering, Universidad Nacional de Colombia, Colombia. E-mail: jdgallos@unal.edu.co ORCID: https://orcid.org/0000-0003-2531-0868

[2] BSc. in Mechatronics engineering, Universidad Nacional de Colombia, Colombia. E-mail: pamozucat@unal.edu.co ORCID: https://orcid.org/0000-0002-5659-7956

[3] BSc. in Mechatronics engineering, Universidad Nacional de Colombia, Colombia. E-mail: ririnconf@unal.edu.co ORCID: https://orcid.org/0000-0002-1053-3148

Preparación de Artículos revista VISIÓN ELECTRÓNICA: algo más que un estado sólido
Fecha de envío: 06 de octubre de 2019
Fecha de recepción: 17 de diciembre de 2019
Fecha de aceptación: 09 de enero de 2020

la agricultura. Sin embargo, esto involucra diferentes desafíos dependiendo de la forma como se realice; particularmente en el caso del seguimiento de trayectorias haciendo uso de la visión artificial. Este artículo describe el diseño, la simulación y la implementación de un algoritmo simple para el seguimiento de una trayectoria a través de la visión artificial, que permite el seguimiento de un dron y su aterrizaje en un punto deseado.

Palabras clave: Visión artificial, ley de control, umbral, seguimiento de trayectoria, UAV.

1. Introduction

The automation of different processes has had a high impact in the industry over the last years, however, recently the automation has been extended to different areas where new technologies such as drones are being adopted. In this particular case, several companies on a national and international scale are using drones to accomplish different tasks, in order to do so, it is needed that those drones follow trajectories in an autonomous way. Initially the drones where guided only by GPS and their trajectories where planned by using their location, nevertheless it was necessary the constant supervision of a person during the route due to the inability of the drone to recognize possible obstacles. Now days, drone have a camera that grants them the capability of acquiring constantly images of their environment, allowing the use of computer vision in order to avoid obstacles and following trajectories without having the need of a constant supervision. In recent years, MathWorks designed a toolbox that allows to program the behavior of the drone Parrot MAMBO with the aim of simulating and implementing different trajectory following algorithms, with this toolbox it is possible to read the values from the sensors of the drone and getting the images taken by the camera located in the lower part of the drone, it is possible to control the motors as well with this toolbox as expected. Considering the fact that the drone has

Fecha de envío: 06 de octubre de 2019
Fecha de recepción: 17 de diciembre de 2019
Fecha de aceptación: 09 de enero de 2020

a very small size and considering as well that the drone will be operated in restricted environments, the aim is to design a simple algorithm for performing the trajectory following, including the landing the effect of the jerk in a desired point.

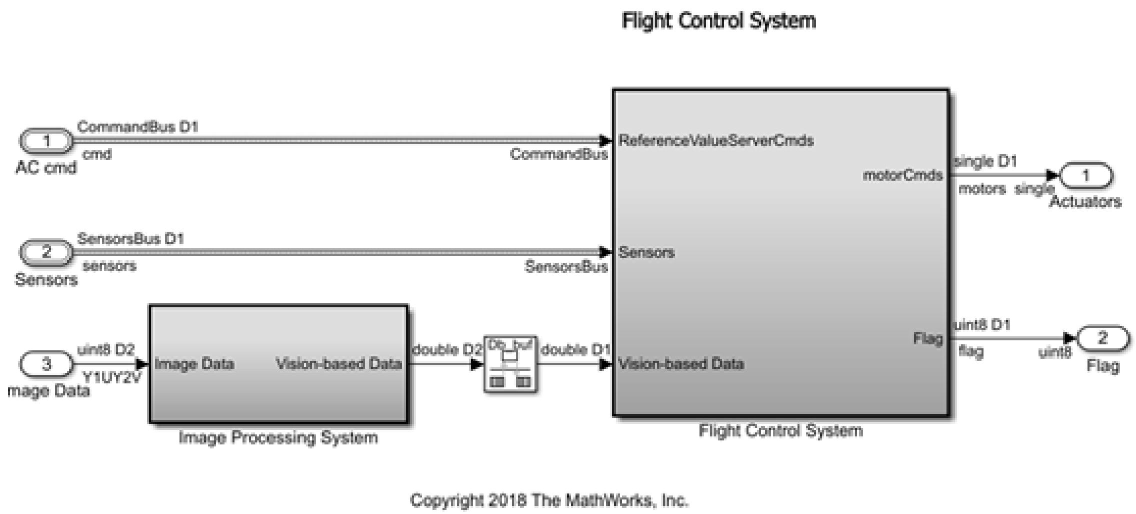
Initially resources used for this paper (both in hardware and software) will be introduced, after that, the methodology for acquiring and processing the image will be discussed alongside the computer vision and control algorithm. Finally, the obtained results will be shown as well as a brief discussion and the conclusions.

2. Methodology

During the experiment the drone “Parrot MAMBO” was used, this drone has a gyroscope-accelerometer which eases the reading of data regarding the acceleration and inclination of the drone in the three main axis, in addition the drone has an ultrasound sensor for applications involving the proximity to some object or surface, finally the drone has an inferior camera with a resolution of 120x160 that allows the collection of images in real time that are processed by the computer vision algorithm. On a software level Matlab was used, specifically the toolbox designed by MathWorks called “Simulink Support Package for PARROT Minidrones”, this toolbox allows the communication between the drone and Simulink giving the user the option of programing the computer vision and control algorithms by using Simulink blocks, [1].

In Figure 1 the control system is showed, this system is contained in a block where the flying programming is performed. As expected, that block collects the reading of all the sensors, the data coming from the camera and a bus of inner signals of the drone, such as the clock. The output of the control system goes into the actuators.

Figure 1. Flight control system, [2].



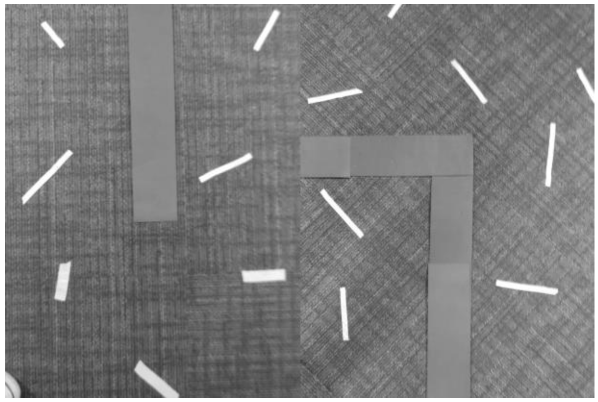
2.1. Acquisition of treating of images

In first place it is necessary to process the data from the camera, in order to do so it must be considered that the initial format of the data is YCbCr and has to be converted into RGB format with the aim of getting an easier manipulation and comprehension of the images (Any format could have been used without major problem). After this initial conversion, it is possible to appreciate the images obtained by the drone in the simulation and during the flight as it can be seen in the Figures 2 and 3 respectively.

Figure 2. Screenshot of the simulation of the drone.

Source: own.

Figure 3. Pictures taken with the camera integrated in the Parrot minidrone.



Source: own.

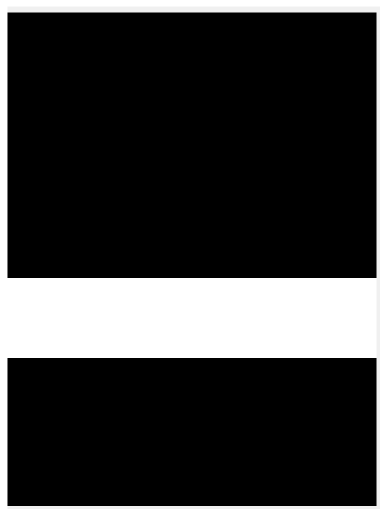
Once the data is put in an adequate representation, it is necessary to ensure that the drone is processing the appropriate information, this means the information corresponding to the track. For this purpose, a thresholding mask is implemented in order to separate the track from the rest of the surface. The mask needs a desired thresholding value in order to split the data, this means, getting a "seed" value to stablish the thresholding and eventually performing the posterior processing of the image. Even though the determination of this number could be done through the manual obtention of the color value in a manual calibration process, however the goal is to automate this process and algorithm is implemented in order to get the thresholding value. This algorithm takes the value of the RGB matrix (or HSV if the lightening conditions are very unstable) in a position where it is knows with certainty that the in the initial moments of the flight, there will be a color corresponding to the track, this will be the seed. It is necessary to clarify that this is made from the assumption that the drone will hold a steady position during the first seconds of the flight. Once the seed is determined, an interval of acceptable values to be considered as part of the track (due to the possible variations) needs to be considered, in order to do so, a range of upper and lower values from the initial seed value is considered.

Fecha de envío: 06 de octubre de 2019
Fecha de recepción: 17 de diciembre de 2019
Fecha de aceptación: 09 de enero de 2020

Finally, the thresholding is used to perform a binarization in order to ease the analysis of the track.

The simulation of the thresholding algorithm can be seen in Figure 4.

Figure 4. Image capture of the simulated drone after passing through the mask and seed algorithm.

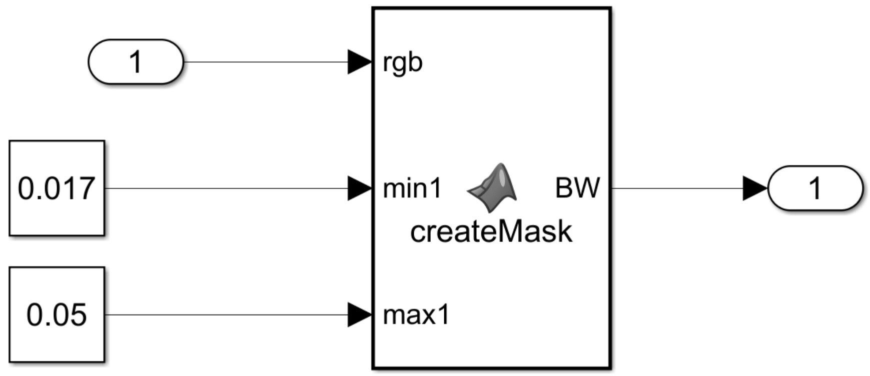


Source: own.

For implementing the mask on the Parrot, it was necessary to use the RGB values of the seed due to possible variations in the lightening during the flight as stated before. In Figure 5 it is possible to see the block diagram that was used, in this diagram the inputs are the image taken by the camera of the drone and the values that stablish the range of filtering that will be used during the route. The output is a Boolean image (black and white) with the applied mask.

In Figure 6 it is possible to appreciate twos shots from the camera of the Parrot after the filtering process, there is an image at the star of the track (left) and a curve forward into the track (right). During the processing of the images it is important to obtain as well as the values of area, perimeter and orientation. To do so, a block of region analysis is employed (Blob Analysis), this allows to get the necessary parameters from Boolean images.

Figure 5. Block diagram to create the mask and filter the image.



Source: own.

Figure 6. Image capture of the Parrot camera after going through the mask and seed algorithm.

Source: own.

During the first part of the image processing, a concatenation was performed with the R, G and B matrices in order to obtain a single work matrix RGB. After that process, the matrix is rotated 90° as seen in Figure 4. This is made because Simulink calculates the orientation from the vertical axis of the image, the issue with it is that when the track is oriented in a vertical way, the values of the angle can be around 179° and 180° instead of negative angles. After performing the process of thresholding and filtering the image, the region analysis is performed.

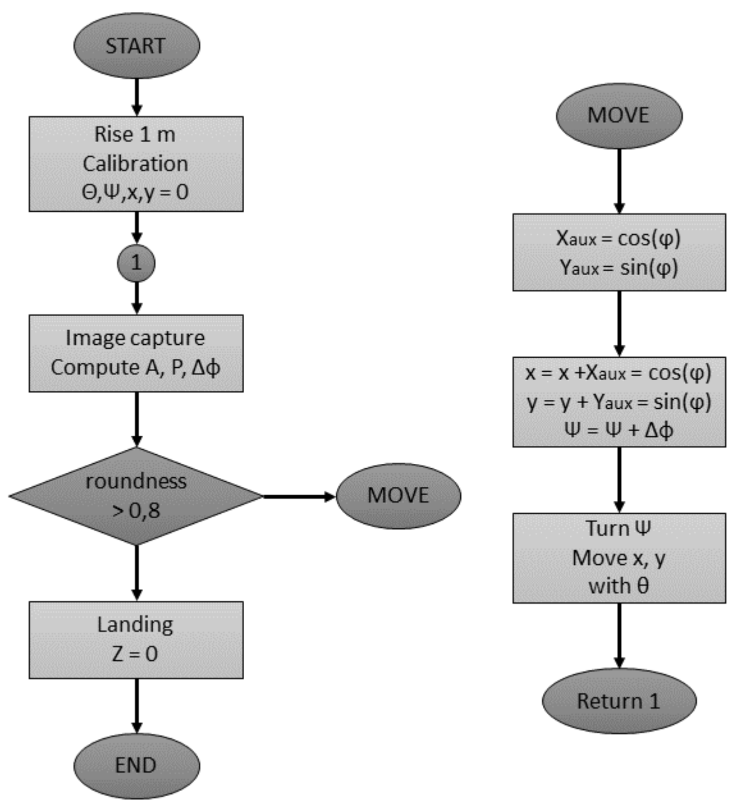
Fecha de envío: 06 de octubre de 2019
Fecha de recepción: 17 de diciembre de 2019
Fecha de aceptación: 09 de enero de 2020

For calculating the orientation an ellipse is drawn, this ellipse must contain all the white pixels in the image in order to calculate the angle between the major axis of the ellipse and the X axis. The area is the number of white pixels in the image and the perimeter is the number of white pixels in the edges of each object.

2.2. Algorithm of trajectory following

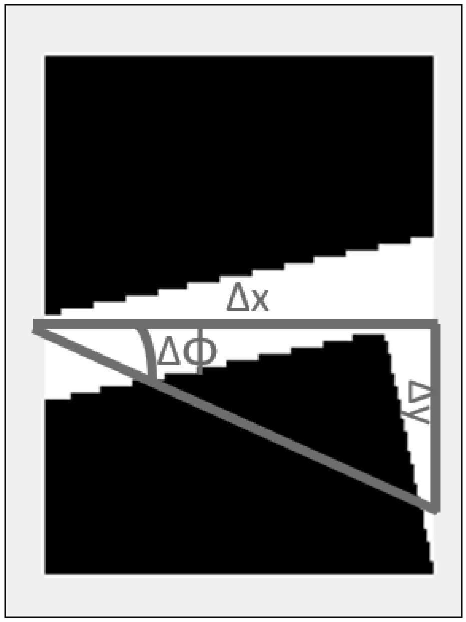
The design of the algorithm is based in the three data obtained from the acquisition and processing of the image, the route of the drone consists in following a track in order to arrive to a landing point marked as a circle, in this point the drone will descend in a smooth way avoiding the effects of jerk. The algorithm can be seen in Figure 7.

Figure 7. Flight Control Algorithm.



Source: own.

Figure 8. Obtaining orientation and position variations.

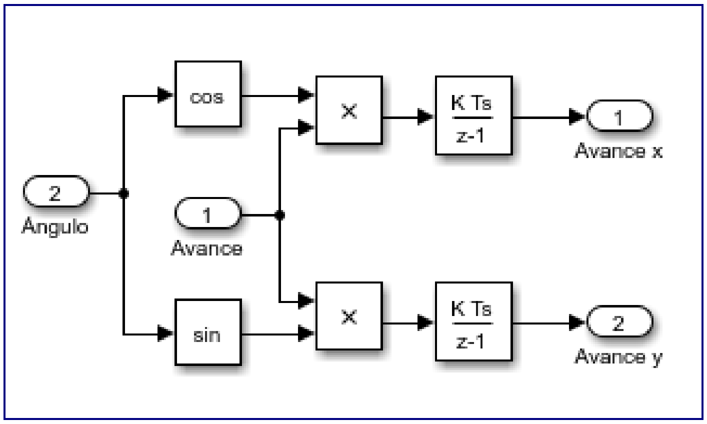


Source: own.

The flight of the drone is based on the orientation acquired by computer vision, considering that this is a 2D image, the movement can be restricted to only two axis, in this case *pith* and *yaw*, the system has a fixed coordinate system located in the takeoff point of the drone, the X axis is the horizontal axis of the image and the Y axis is the vertical one. The data obtained from orientation ($\Delta\varphi$) corresponds to the angle that the drone should rotate to stay in the same trajectory as the track, however, due to the fact that the system is based on a fixed reference system, it is necessary that the angle is taken to that system, this is made by obtaining the angle progressively from the image when the drone rotates. The drone must be told how to advance in the coordinates X and Y, this is made by a *pitch* movement, the coordinates are calculated with the angle previously obtained from the orientation by applying the *sin* and *cos* functions. Just like in the previous case, the components (Δx, Δy) must be progressively added due to the fixed reference system.

Fecha de envío: 06 de octubre de 2019
Fecha de recepción: 17 de diciembre de 2019
Fecha de aceptación: 09 de enero de 2020

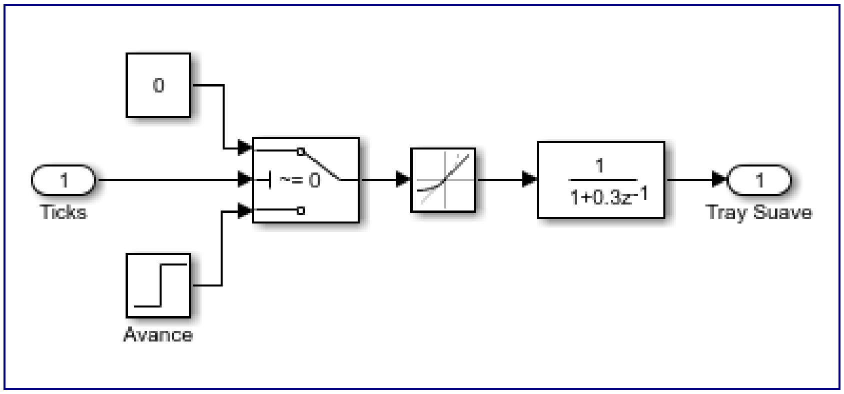
Figure 9. Move in Simulation.



Source: own.

During the trajectory a smooth movement is expected without important changes in the acceleration, preventing in this way significant effects from the jerk, in order to avoid this a *rate limiter* is used together with a discrete filter, avoiding in this way that the slope between the changes of angle and position gets to be high, smoothening the movement.

Figure 10. Smoothing of the paths to be performed.



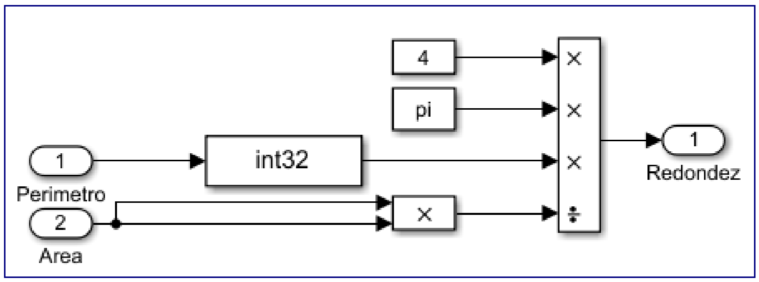
Source: own.

In order to verify the presence of a circle in the track (the mark of the landing point), the property of roundness is used. This is a relation between the area and the perimeter previously obtained, equation (1), [3].

$$round = \frac{4\pi A}{p^2} \quad (1)$$

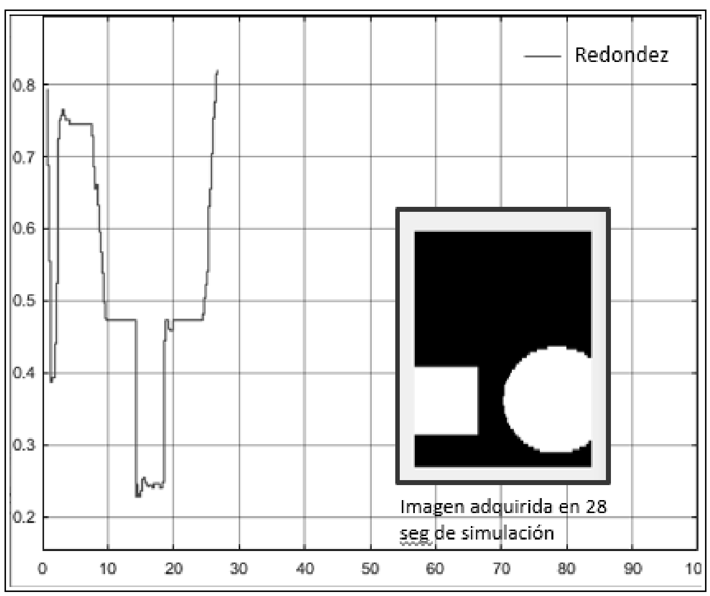
Being round the roundness, A the area and p the perimeter.

Figure 11. Roundness calculation in simulation.



Source: own.

Figure 12. Roundness result throughout the simulation.



Source: own.

When the roundness takes a high value, in this case over 0.8, means that there is a circle in the track and the drone will land, if this is not the case the drone will continue its route. The landing is expected to follow a soft trajectory, avoiding again a notorious jerk, a rate limiter is added in order to limit the slope during the route.

Fecha de envío: 06 de octubre de 2019
Fecha de recepción: 17 de diciembre de 2019
Fecha de aceptación: 09 de enero de 2020

3. Results

By verifying the appropriate functioning of the algorithm in the simulation, the drone achieved the desired goals following the trajectory and landing in a smoothly in the desired points, with this results the toolbox is used to implement the algorithm. This toolbox generates a code in C and programs the microcontroller of the drone. In order to verify the appropriate functioning during the implementation, a track with three linear segments is used alongside a circle for indicating the landing point. The illumination conditions are controlled in order to ensure a small variation in the calibration values of the thresholding mask. When the algorithm was implemented, it was necessary to make and adjustment over the roundness condition because the illumination causes the appearing of certain curves due to the shadows creating a roundness value higher than expected, taking the drone into landing in the wrong position.

During the trajectory following and the landing it was possible to observe smooth trajectories thanks to the digital filter and the implemented *rate limiter*.

Figure 13. Photograph of the track and the Parrot minidrone following it.

Source: own.

Fecha de envío: 06 de octubre de 2019
Fecha de recepción: 17 de diciembre de 2019
Fecha de aceptación: 09 de enero de 2020

4. Conclusions

Thanks to the computer vision algorithm that performs the segmentation of the images, the track is successfully followed and an automated flight is achieved, including the trajectory following and the landing.

By limiting the slope in the change of position (meaning, limiting the speed and filtering it), the effects of the jerk are avoided not only in the landing but in the movement throughout the different axis.

5. Discussion

For future works it would be expected to implement an automated mask, avoiding the manual calibration process, this would help as well to have and adequate flight without the need of a fully controlled illumination.

Finally, it was noted that during the simulation and the implementation, the drone needed a servovisual control system capable of maintaining centered and steady trajectory regarding the captured images. This was necessary because when the drone rotated, a mismatch in the position was seen and eventually increased due to the action of the integrator.

Acknowledgements

To MathWorks for allowing us the use of their toolbox and taking charge of the National Minidrone Competition 2018.

To Maitreyee Mordekar, member of the MathWorks India team for her advices for improving the simulation and achieving and adequate implementation of our Algorithm.

Fecha de envío: 06 de octubre de 2019
Fecha de recepción: 17 de diciembre de 2019
Fecha de aceptación: 09 de enero de 2020

To Ricardo Ramirez, teacher of the Universidad Nacional de Colombia for encouraging us to participate in the contest, promoting the formation of our investigation group and getting the necessary tools to promote the researching process.

To Flavio Prieto and German Ramos, teachers of the Universidad Nacional de Colombia, whose knowledge and advices helped us during the creation of the algorithm.

References

[1] MathWorks, “PARROT Minidrones Support form Simulink”, 2019. [Online]. Available at: https://www.mathworks.com/hardware-support/parrot-minidrones.html

[2] MathWorks, “Compute statistics for labeled regions”, 2019. [Online]. Available at: https://www.mathworks.com/help/vision/ref/blobanalysis.html

[3] M. A. Arias-Marta and J. A. Sierra-Ruiz, “Procesamiento de imágenes para la clasificación de café verde”, thesis, Pontificia Universidad Javeriana, Colombia, 2016.