Self-guided AR Drone using LabVIEW

Rodica Holonec, Romul Copindean, Florin Dragan, László Rápolti

Faculty of Electrical Engineering, Technical University of Cluj-Napoca, Romania

Abstract *– The Unmanned Aerial Vehicles (UAVs) can be used for a rather large number of applications. One of the advantages of drones is that they can be autonomous, being able to keep track of the environment that they are in, while carrying out their designated mission. This paper studies the possibilities to implement a self-guided drone with the help of image processing and acquired data from its onboard sensors. The applications were realized in LabVIEW environment by using a predefined pallet intended for the communication between a PC and the AR Drone 2.0.*

Keywords *– Self-guided drones, image processing, filtering, virtual instrumentation system*

1. INTRODUCTION

Unmanned aerial vehicles became very used in the past few years. The military, for example uses them frequently for different applications, like for example surveillance missions. Drones can be very small and can go to places too risky for humans to go to. The AR Drone was selected for the purposes of this experiment because it is relatively cheap compared to other drones and can be easily accessible for anybody willing to test it [1].

The drone has an onboard processor called ARM Cortex A8 which is running a Linux operating system. It can take off, land and hover automatically, these features being preprogramed in its operating system. It also has a multitude of sensors on board, for example it has an accelerometer, gyroscope, magnetometer, pressure sensor, ultrasonic sensor and it is equipped with two cameras. One high resolution wide angle camera in front of it, as seen in figure 1, and one low resolution camera at the bottom [2].

Fig. 1 Forward facing cameras angle

The communication between the PC and the drone is based on WI-FI. Because the quality of the signal drops rapidly with the distance, the AR Drone is not capable of traveling to great distances than 100 meters without signal amplification. The tests explained in the paper were conducted in a laboratory so it was not necessary to boost the signal. However, the quadcopter can also be equipped with GPS communication by connecting a compatible GPS module to its USB port. This module can help it travel farther distances without being connected to a user by programing it to go to a specific coordinate to carry out a specific task (e.g. gathering images, weather data, and so on). [3]

The drone is also equipped with four brushless DC motors, each of them controlled by its own logic board that constantly checks the speed of the rotor and if necessary adjusts them per the angle and task it is carrying out.

Onboard the drone it can be also found a Li-Po battery with a capacity of 1500 mAh at 11.1V, that ensures a 15-minute of flight. The battery percentage is constantly verified and when it drops below 10% it alerts the user and returns home, or if it drops below 5% the drone automatically lands safely.

Due to safety hazards in an indoor laboratory and for avoiding any injuries the drone has an external housing which is made of a durable and flexible expanded polypropylene material. To protect the internal components from shocks, they are protected by a special foam material that is placed upon the carbon fiber structure. For the protection against water the exposed ultrasonic sensor is coated with a water-repellent material [4].

Because of the lightweight materials and of its powerful motors, the drone can perform a number of preprogramed animations like backflips, “dancing”, salute and so on. All of these functions can be implemented with the help of the AR Drone LabVIEW Toolkit.

In this paper we experimented some applications based on image processing in order to have a self-controlled drone. The algorithms were based on data received from quadcopter. These values were the three rotational components: roll, pitch, and yaw. They represent the angle of the rotational movement on the x-axis, the y-axis, and z-axis (figure 2).

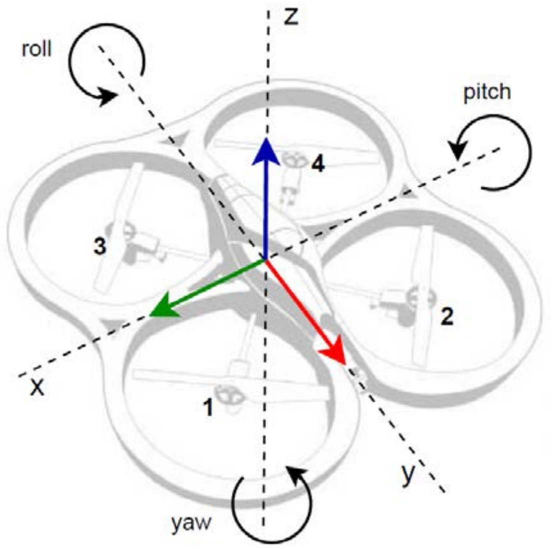

Fig. 2 The quadcopter orientation defined by the roll, the pitch, and the yaw angles [5]

2. SYSTEM DESCRIPTION

The system is based on Wi-Fi communication. The drones' processor gathers the data acquired from the sensors and processes them according to the software that's running. For stabilization, it will process the data from the gyroscope and accelerometer even if it is running another program parallel on it. After the data was processed it will be sent to the drivers, which will decide to speed up or down the rotation of the blades. Every bit of data that is acquired by the sensors will be transmitted via the wi-fi signal and then it can be processed by a second user as it can be observed on the diagram below (figure 3) [6] [10].

In our system, LabVIEW was used for data gathering and processing. The images taken by the drone were sent to the laptop in a H265 coded format being processed and decoded with the help of the AR Drone Toolkit. After the signal was acquired by the software, it was processed and then the calculated data (roll, pitch and yaw) where sent back to the drone in order to execute the needed maneuvers. Meanwhile the data from the other sensors were also processed and correlated with the calculations of the quadcopter orientation angles.

With the help of the AR Drone toolkit the data can be easily extracted, containing the decoding process and then the data can be processed and sent back by the user [7] [11].

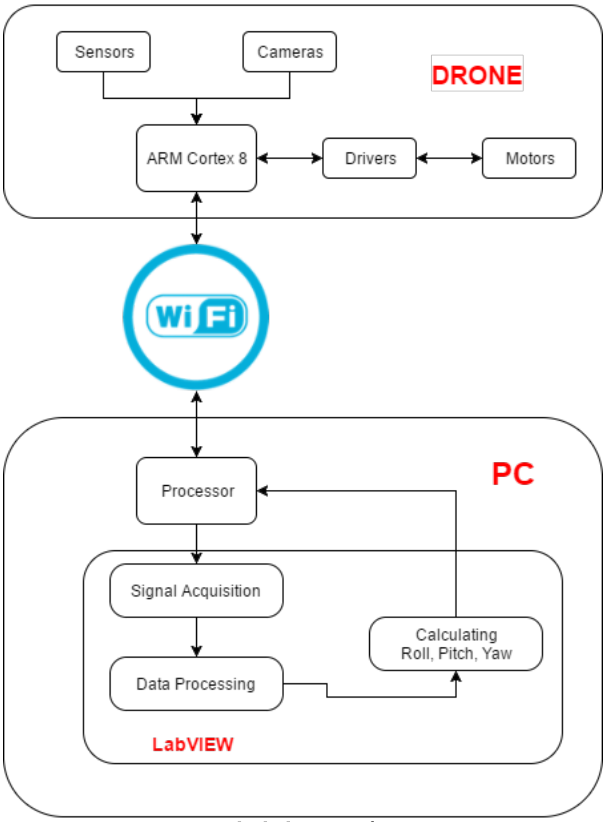

Fig. 3 Block-diagram of system

2.1. Image Processing

Image processing is a process of analyzing signals of which the input is an image, a series of images, or a video; the output could either be a different image or a set of characteristics or parameters related to the original image. Most image processing techniques treat an image as a bi-dimensional signal and apply a multitude of signal processing techniques to it by using different mathematical operations.

In the proposed application, the video fed from the drone was used to search for matching a specific pattern and color that we could easily distinguish from the surrounding area. Different type of objects were experimented and the conclusion was that for the drone bottom camera the one from figure 4 was the best option to be recognized with high scores.


Fig.4 The explored object by the bottom camera

The first few experiments were carried out trying to detect a white paper, and after concluding that it was not good enough for this application it was tried the object from figure 2.

The image acquired from the drone was converted to a pixmap from which a color pattern matching algorithm was used so the object was detected and localized by its coordinates.

In the second experiment, the front camera was used to try to cover the movement of the drone on all three axes. Because the drone had been designed for a variety of games and the drone is equipped with specific tags like in figure 5, the decision was to use one of this colored pattern objects to control the drone.

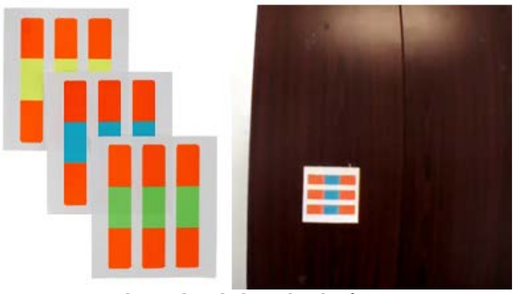

Fig 5. The explored object by the front camera

2.2. Calculating roll, pitch and yaw

The front panel of the implemented VI (figure 6) contains: the acquired image, the number of detected objects that have matching scores over a certain level, the current position (in pixels) of the detected objects, the information about the bounding box, the speed on the two axes and the current height position of the drone. The user can also control the drone by using five different buttons: one for hovering, one for takeoff and landing, one as a fail-safe in case something goes wrong which works by cutting the power supply of the motors, one for the prediction of the drones' position and finally one for stopping the program.

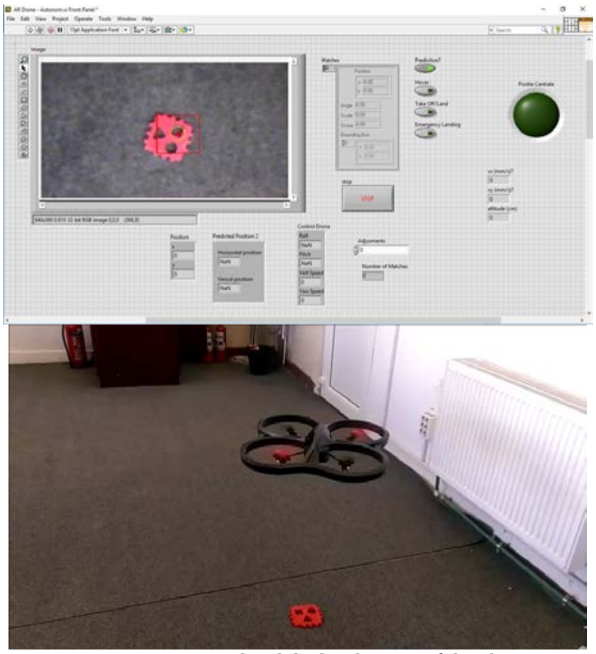

Fig 6. VI Front Panel and the localization of the object

After the drone gets the signal for takeoff it will read the data from the sensors and sent them to the pc. The application used the speed on x and y axes named vx and vy both being measured in mm/s. They have been converted then in cm/s. The altitude was also measured in pixels/cm. The drones' displacement limits where set for the half of the image which was 25 pixels. For keeping the drone above an object, we used a subVI called "Keep Above Object.Vi" which will only activate in case it detects an object otherwise the Hover function will be activated

On the other hand, if the VI finds an object it will calculate with the help of a SubVI called "Calc Pitch&Roll.vi" the exact values needed for the drone to move above the object. The exact position calculated from the image in pixels will be used and this data will be converted in such a way as to create an interval from 0 to 1, if the drone is in a positive quadrant and if it is in a negative quadrant the interval will be multiplied with -1, If the drone is right above the object it will enter Hover mode as seen in figure 7.

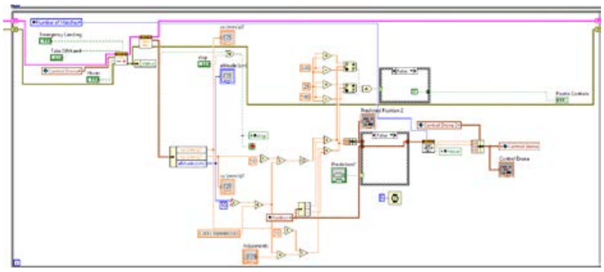

Fig. 7 Front camera experiment. Application block diagram

For the front camera, the same principles were implemented. By using the proper colored object the drone was able to follow the tag. The yaw and roll functions were implemented.

The experiment can be seen in figure 8.

Fig. 8 Experiment with the front camera

2.3. Perimeter control

The quadcopter could also be used for security reasons, like for example it could patrol an area that is set by the user. Because of this an algorithm was implemented in order to impose a specific path to be followed by the drone.

In the proposed experiment the drone was made to constantly go forwards and backwards every time pulling a bit to the right until it finds the object. When this was found, the drone was imposed to stop right above it.

The algorithm was implemented with the help of a case structure as seen in figure 9 which sent four specific moves to the drone. The only problem was to compensate for the speed when changing directions, otherwise it would fly continuously in the other direction or even crash in a wall. To prevent that, the automatic hover function was used.

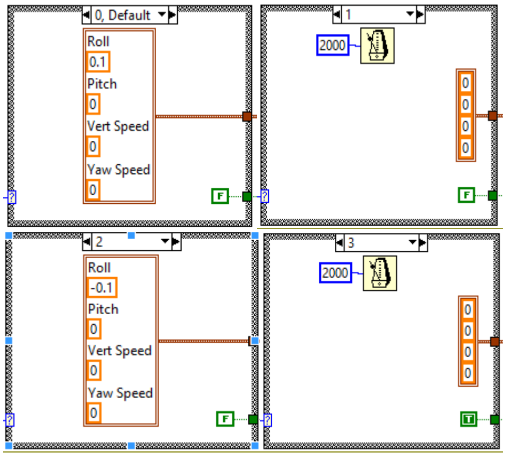

Fig. 9. Case structure used for patrol function

3. CONCLUSIONS

With this application, we managed, in laboratory conditions, to implement a self-guided AR drone in order to find an object in a limited area and follow it. The algorithms can be improved and extended in order have a fast and accurate drone behaviour.

For future experiments, we want to realise a software that can recognise a multitude of objects with the help of its cameras, and after this is implemented, with the help of GPS positioning send the drone to automatically fly to a designated area to scan.

With the help of this kind of algorithm there are several applications that could be done, like for example it could be used to go through a forest and in case it detects waste lying on the grass or anywhere else, it would pinpoint the location on a map via its GPS. This software also means that the drone must have object avoidance capabilities to avoid any trees or branches that could be in its way.

Another application for the quadcopter is to be used in agriculture, to go between isles of crops or any kind of plants and take measurements of the humidity temperature brightness and taking picture of the plants determining its age the health of them by measuring the greenness of the leaves or determine its size.

It also could be used as a security system to detect intruders on a perimeter and send a signal to the owner when something changes in the picture.

REFERENCES

1. Cai, Guowei, Chen, Ben M., Lee, Tong Heng. “Unmanned rotorcraft systems” Advances in Industrial Control, 2011, Springer.
2. Stephane Piskorski, Nicolas Brulez, “AR.Drone Developer Guide”SDK 1.6, 2011
3. Carlos Cambra Baseca; Juan R. Diaz; Jaime Loret, “Communication Ad Hoc Protocol for Intelligent Video Sensing Using AR Drones”, The IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks, 2013
4. Krajnik T, Vonasek V, Fiser D, Faigl J. “AR-Drone as a Platform for Robotic Research and Education”. Research and Education in Robotics :EUROBOT. Heidelberg. 2011
5. Jakob Julian Engel, “Autonomous Camera-Based Navigation of a Quadrocopter”, Master’s Thesis in Informatics, Technical University of Munich, December 15, 2011
6. Pedro Castillo, R. L. (2005). “Modelling and control of mini-flying machines: with 126 figures”. London : Springer.
7. National Instruments- “AR Drone Toolkit for LabVIEW”
8. Béla Lantos, Lőrinc Márton, “Nonlinear control of vehicles and robots”, Advances in Industrial Control, 2011, Springer.
9. Autoren:Lucas Vago Santana, Alexandre Santos Brandão, Mário Sarcinelli-Filho, “Navigation and Cooperative Control Using the AR.Drone Quadrotor”, Journal of Intelligent & Robotic Systems, 2016.
10. Gabriela Tonţ, Radu Adrian Munteanu, Dan George Tonţ, Dan Iudean „Aspects Regarding the Unidirectional Two-Port Circuits Implemented by Means of Electronic Gyrators”, Journal of Advances in Electrical and Computer Engineering, Volume 16, Number 1, 2016
11. R Munteanu, D Iudean, C Muresan, V Petean, „Access system in restricted areas based on programmable logic controller and GSM modem”, The 13th International Conference on Engineering of Modern Electric Systems (EMES), 2015, Publisher IEEE

Rodica Holonec
Faculty of Electrical Engineering, Technical University of Cluj-Napoca, 26-28, G. Bariţiu st., Cluj-Napoca, Romania
Rodica.Holonec@ethm.utcluj.ro

Romul Copîndean
Faculty of Electrical Engineering, Technical University of Cluj-Napoca, 26-28, G. Bariţiu st., Cluj-Napoca, Romania
Romul.Copindean@ethm.utcluj.ro

Florin Drăgan
Faculty of Electrical Engineering, Technical University of Cluj-Napoca, 26-28, G. Bariţiu st., Cluj-Napoca, Romania
Florin.Dragan@ethm.utcluj.ro

László Rápolti
Faculty of Electrical Engineering, Technical University of Cluj-Napoca, 26-28, G. Bariţiu st., Cluj-Napoca, Romania
Laszlo.RAPOLTI@student.utcluj.ro