

# Towards Image Mosaicking with Aerial Images for Monitoring Rice Crops

Juan Rojas<sup>1</sup>(✉), Carol Martinez<sup>2</sup>(✉),  
Ivan Mondragon<sup>2</sup>(✉), and Julian Colorado<sup>1</sup>(✉)

<sup>1</sup> Department of Electronic Engineering, School of Engineering,  
Pontificia Universidad Javeriana, Bogotá, Colombia  
{rojas-jp, carolmartinez}@javeriana.edu.co

<sup>2</sup> Department of Industrial Engineering, School of Engineering,  
Pontificia Universidad Javeriana, Bogotá, Colombia  
{coloradoj, imondragon}@javeriana.edu.co

**Abstract.** Around 8 to 10 million Ton of rice are required in the following years to be able to supply the demand of the overall population. Analysis and monitoring of rice crops becomes nowadays very important issue for farmers, for ensuring a rice production level to cope this demand. This paper presents simulation results of an algorithm that allows to plan and create 2D maps using the technique of image mosaicking with multiple geo-referenced aerial images (multispectral images in the scope of the project). The planning algorithm is called Image Capture algorithm. It takes into account the area the UAV has to cover, the camera configuration, and the state of the UAV in order to define where to take the pictures to build the mosaic. The algorithm presented in this paper was developed in ROS (Indigo) and simulated in Gazebo. The results of this first approach to the 2D mapping of a rice crop allows to see that using the proposed algorithm, it is possible to automate the process of acquiring the pictures for creating the mosaic, ensuring that all the area of interest is covered. By using this algorithm, pictures will be acquired only in specific areas. Therefore, keeping the storage capacity on-board, under control.

**Keywords:** Homography · Stitching · Unmanned Aerial Vehicles · Computer vision · Image mosaicking

## 1 Introduction

The rice is one of the most important cereals. It represents the 25% of the calories consumed by humans in a day, and it is one of most cultivated cereals in the world [1]. There are four types of rice cultivated around the world, the first and the most important is the Indica type. This type represents the 75% of the world trade. The second type is the India Basmati. This type is aromatic and represents the 13% of the global trade. In third position with the 10% of the

global trade is the japonica type. Finally, the fourth type is the glutinous type which is cultivated in some parts of Asia [2].

The most cultivated type of rice in Colombia is the Paddy Verde [3]. This specie of rice represents the rice core trade of the country, due to its fast growth, and resistance to plague and environmental changes. Colombia has two methods of growing rice. The mechanized and the manual methods. The mechanized can be divided in irrigated and upland rice. The difference between them is the source of the water. In the irrigated method the water proceeds of water districts, whereas in the upland method the rice is irrigated with rainwater. The mechanized method is used in the 94% growing rice fields, but the 70% of rice production is related with the upland method.

Colombia has many areas where the rice can be cultivated, but due to the cost of crop care, chemist and workforce [4], Colombia has began to import rice. In other countries such as China and USA, these problems have been reduced thanks to the use of technologies to monitoring the state of the crop (using for example drones, sensor networks, and satellite photos).

Image mosaicking is a computer vision technique used in many applications to generate a wider view of the scene, than the one obtained with only one image taken with a normal camera. This technique consists on aligning multiple overlapping images captured from a moving camera. One of the advantages of using image mosaicking in precision agriculture is to be able to capture in one image all the information of the crop for further analysis. Therefore, by analysing only that image, it is possible to extract information about the state of the crop without damaging the crop in the field. Therefore, it is considered a non-invasive technique for analysing the state of crops.

In this paper, we present results of the COLCIENCIAS 120371551916 project: "Aerial sensing and monitoring of rice crop fields applying precision agriculture techniques". The scope of this project is to develop a testbed for evaluating rice varieties in the field, by using sensors located on-board an aerial vehicles that allows to monitor in a non destructive way rice crops.

From previous results [5] at Pontificia Universidad Javeriana, a multispectral camera and the ASCTEC-Pelican were used to generate a multispectral mosaic that describes a specific terrain. the Normalized Difference Vegetation Index NDVI was used to estimate the healthy of the plants and the density of the vegetation. However, in this paper we focus on the image capture problem for creating the mosaic. The planning algorithm is called Image Capture algorithm. It takes into account the area the UAV has to cover, the camera configuration, and the state of the UAV in order to define where to take the pictures to build the mosaic. We present simulation results of the algorithm that allows to plan and create 2D maps using the technique of image mosaicking with multiple geo-referenced aerial images (multispectral images in the scope of the project). In the paper, the pictures to build the mosaic are taken in defined intervals during flight, in a specific altitude, and with a predefined overlap, to ensure the quality of the mosaic (in this paper 30% overlap is tested). The above values can be measured assuming the system has a predefined trajectory, a specific linear

velocity and feedback from the sensors on-board the UAV (GPS and IMU). The algorithm was developed in ROS (Indigo) and simulated in Gazebo.

The paper is organized as follows. First, Sect. 2 presents related work on precision agriculture and image mosaicking. In Sect. 3, the UAV system is presented. In that section, the simulated UAV and its relation to the real drone are discussed. Section 4 explains the image capture algorithm and the image mosaicking algorithm. Finally, Sect. 5 presents results of the image capture and image mosaicking algorithms; and Sect. 6 presents the conclusions and direction of future work.

## 2 Related Work

Precision farming has become an essential tool to improve agricultural production and productivity. It includes techniques for crop monitoring and management. The crop management essentially depends on the irrigation type, nitrogen levels, and chemists varieties used in it. In a specific site, crop management requires a fast diagnosis method that allows the farmer to have the measurement and analysis of the crop field in real time. This is to avoid crop losses, increase crop quality, and optimize agriculture supplies. The latter will impact the production cost (reducing it), and will also help protecting the environment. Remote sensing and monitoring applications based on unmanned aerial platforms, equipped with a set of specific sensors and instruments have been widely developed on the last few years, as a rapid deployment tool for field applications [6].

Field studies in Colombian rice zones have demonstrated that the biomass dynamics is related to a higher performance of the crop, when the conditions of the field are the best [7]. It is important to notice that the temperature and low solar radiation also affect the biomass dynamics.

To take the best behavior of the crop, many techniques and technologies have been developed that give a measurement of the field and allow the farmer to have solid information of the crop. An example of the developed technologies can be seen in [8], where a depth camera, a Laser, and a UAV are used to estimate the status of the soil. In [9], a sensor network is implemented to measure and classify the percentage of arsenic, ammonium, and other chemical substances in the field, with the aim of finding their origin and reduce their impact to the field. In [10], a satellite IKONOS is used to take several multispectral images of a grape crop, with the aim of analyzing the state of the crop; and to provide the farmers information to improve the crop care. To address this problem, the Normalized Difference Vegetation Index NDVI is used to classify segments of the crop. The size of the leaves is also measured to estimate their growth. On the other hand, in [11] a method to estimate crop area coverage is presented. This method uses a UAV and a spatial sampling method, based on Moderate Spatial Resolution (MSR) image classification results, to estimate the area.

It is important to mention that the use of UAVs in combination with image processing is a low cost tool compared with the analysis of satellite images or

the implementation of network sensors. Additionally, UAVs allow to have a full view of the field quickly, speeding up the process of measurement and analysis of the crop variables. In the work presented in [12], authors present a method to classify rice quality, based on high resolution images taken by a UAV flying at low altitudes.

Digital image mosaicking is a technique that has increased importance over time in many fields of the industry (military intelligence [13], image processing [14, 15] and biomedical approaches [16], etc.), especially in precision agriculture [17–19]. There are different approaches that allow to generate an image mosaic. Some are focused on improving the motion estimation problem, and others focused on post-processing the mosaic for solving problems related to drift.

In the work of Tom Botterill et al. [20] an aerial vehicle was used to capture multiple images of a specific area. The main idea was to provide the operator with an image that has a larger field-of-view. For feature extraction and matching they use Bag-of-Words, which allow to find, in an efficient way, the matching of points with wide-baseline. A seam-placement algorithm to de-noise and rendering the mosaic was also implemented.

Moussa [21] proposed an algorithm that allows to manipulate several images in an efficient way. SIFT (Scale-invariant Feature Transform) features were used; and the correlation between images was used for feature matching. Then, the algorithm estimates the motion of the images, warps them, and generates the mosaic.

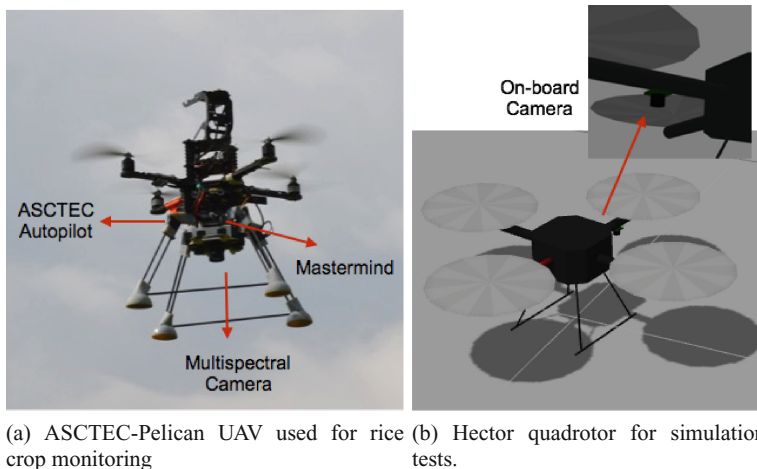
In [22], three algorithms for mosaic generation, were compared: controlled, uncontrolled and semi-controlled mosaics. The images are taken from two different test flies and has several geo-referenced points, based on GIS (Geographic Information System) and GCPs (Ground Control Points), that allow the algorithm to generated the stitching between images using these points as references.

As a general conclusion from the woks presented above is that the most important step in image stitching is the selection of the feature extraction and matching algorithm. In this sense, to detect growing areas it is possible to implement different techniques of image processing that segment the blocks of crops and other elements, as shown in [23].

For UAV-based image mosaicking, a mission planning algorithm has to be carefully designed in order to ensure the coverage of the terrain, taking into account the UAV autonomy. The research presented in [24] presents a solution to the Coverage Path Planing (CPP) problem. The proposed algorithm allows to reduce the consumption of energy; and to ensure the image resolution, the speed and acceleration of the drone. In the work of [25], a new path planning method is proposed. It allows to optimize the drone task, path length, and coverage lost. The algorithm is based on three planers, Route Planer (RP), path planer (PT), and coverage planer (CP). All the called planers are based on the modified versions of Genetic Algorithms and Dijkstra Algorithm ( $A^*$ ).

### 3 UAV System

The drone that is going to be used in the project is the ASCTEC-Pelican from Ascending Technologies, shown in Fig. 1a. The system is composed by a ground station (a standard PC), and the drone which is equipped with a Mastermind board (for image processing and data acquisition); the ASCTEC autopilot; and cameras: an RGB camera SONY FCB-EH6500, and a multispectral camera ADC Lite from Tetracam. The multispectral camera captures visible light wavelengths longer than 520 nm and near-infrared wavelengths up to 920 nm. It is located looking-downwards in the bottom of the drone, and aligned with the center of mass.

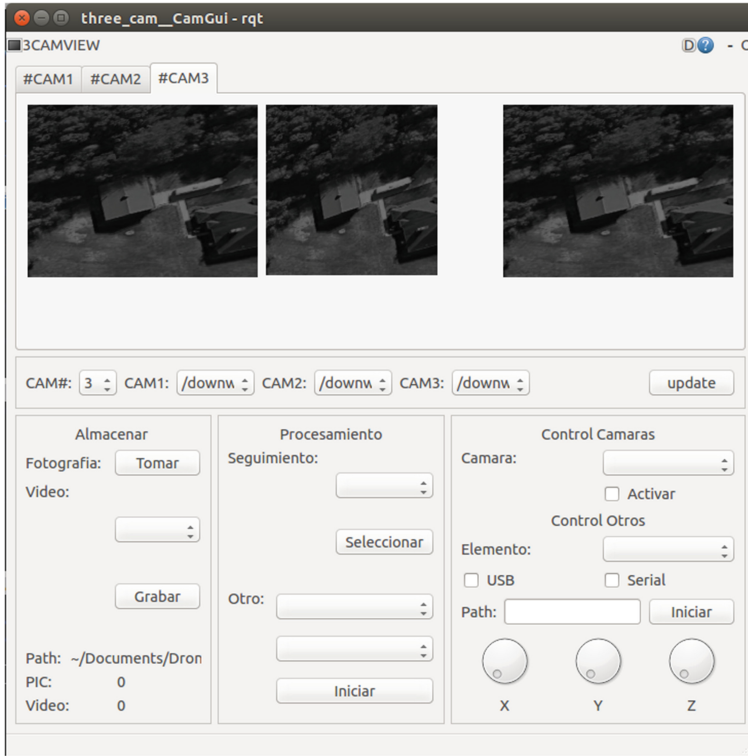


**Fig. 1.** UAVs used for rice crop monitoring. Figure 1a shows the ASCTEC-Pelican test-bed used in the project. It is equipped with a multispectral camera, the ASCTEC autopilot, and a Mastermind (on-board computer). Figure 1b shows the simulated drone used for software development.

The ASCTEC-Pelican comes with a GPS waypoint navigation strategy with  $\pm 1$  m of accuracy. Data from sensors are sent via 2.4 GHz XBee link to the ground station. In the ground station there are two computers, one for running the Graphical User Interface that comes with the ASCTEC-Pelican, that allows to control the drone (autonomous take-off and landing; and waypoint navigation) and provides information about the state of the drone.

The second computer runs a customized GUI, design in Qt with Ubuntu 14.04, shown in Fig. 2. This GUI allows to start the different processes that run in the Mastermind board, and as shown in the figure, allows to have visual information about the different cameras on-board the UAV.

On the other hand, the Mastermind computer on-board the UAV, runs Ubuntu 14.04 and the framework used to control the autopilot from the mastermind was

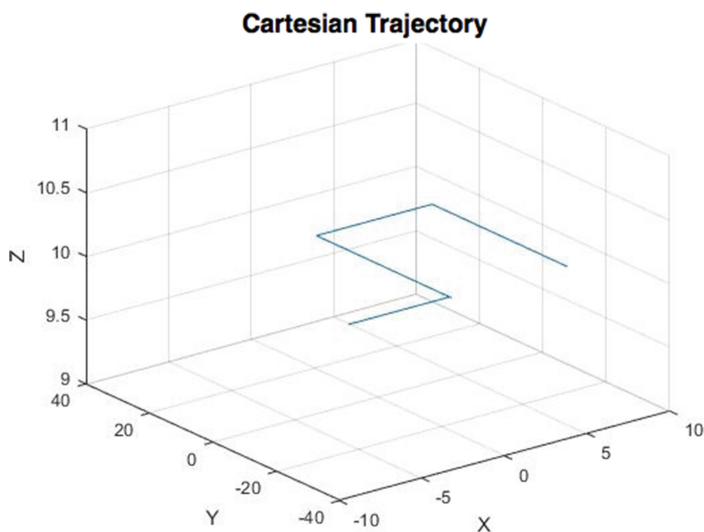


**Fig. 2.** GUI designed for controlling the cameras and running the on-board computer vision algorithms

ROS (Robot Operating System). Under this framework, the trajectory planning algorithm, the position and linear velocity control are independent nodes, and all of them run on-board, but are monitored from the ground station.

### 3.1 Simulation Environment

In order to develop the different algorithms required by this project, a simulated environment, based on ROS, was created. The drone model used was the Hector Quadrotor, a ROS package that allows to simulated and control a UAV of type quadrotor [26]. The model of this drone runs in the Gazebosim [27], a simulation environment that allows to integrate multiple sensors like IMUs, cameras, pressure sensors, thermal cameras, among other sensors. The drone model communicates with ROS under specific topics related to the multiple elements that are placed on-board the simulated drone (e.g. sensors). For the case of this project, an RGB camera that points to the ground, an IMU, and a GPS were used. Figure 1b shows the Hector quadrotor with the hardware that was configured.



**Fig. 3.** Example of the generated trajectory points, when using the trapezoidal move profile, used in the simulation tests.

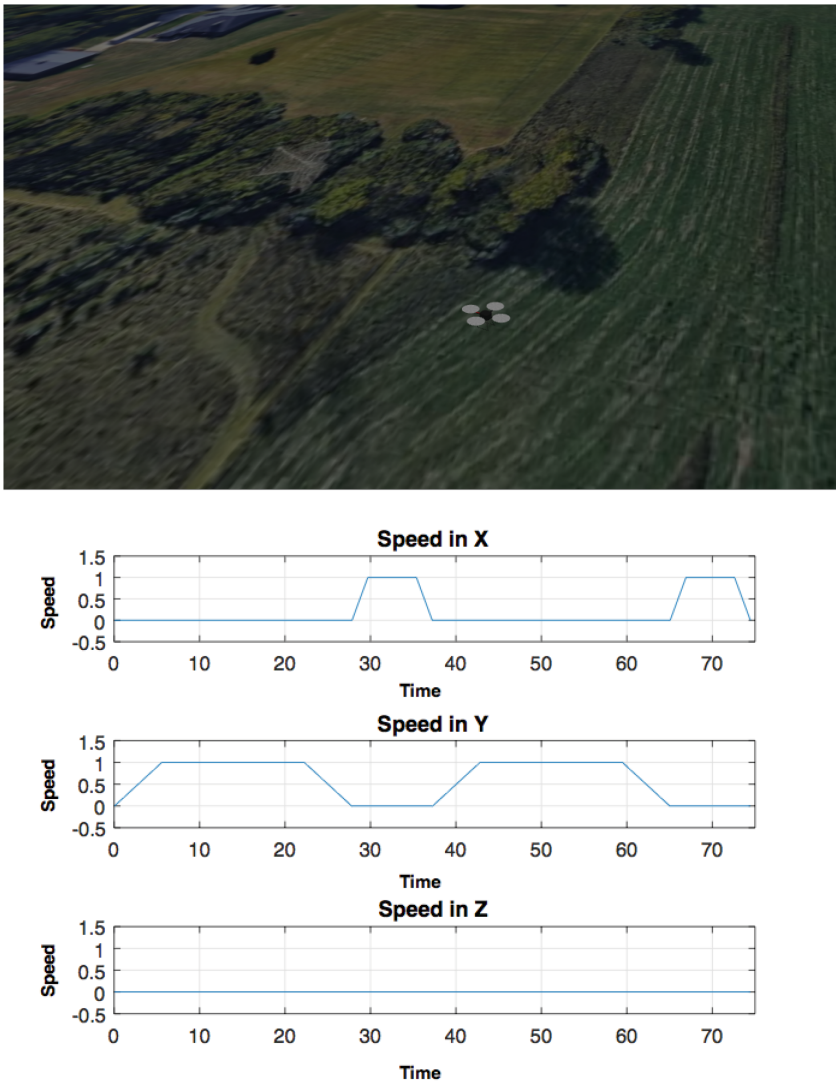
The predefined trajectory model can be seen in Fig. 3. This trajectory was generated using a trapezoidal move profile. This model gives the cartesian points, the linear velocities, and the acceleration that the drone needs to follow, in every moment, to cover the working area.

To simulate the environment a satellite image of a rice crop field was taken from Google Earth, and a digital elevation model was taken from Google Sketchup. All the hardware place on-board the simulated UAV, and the physics variables are scaled and related with the world definition file from Gazebo. Once the world model is created in gazebo-sim, the drone is loaded, and the simulated system is ready to work. Figure 4 presents a picture of the environment and the trajectory followed by the UAV.

## 4 Image Mosaicking

The main objective of the COLCIENCIAS project is to analyze the state of the rice field. Assuming the area of the rice crop is known, the UAV should be able to plan the proper trajectory to follow, in order to take the pictures required to create an image mosaic of the field. This image mosaic will be the one used to analyze the state of the field.

The Image Capture algorithm presented in this paper will take into account the characteristics of the camera, the field area, and information from the state of the UAV, to define the places where the images from the crop will be acquired. This algorithm will ensure that the captured images will cover the flight area with the enough overlap to create the mosaic of the crop area. Once image are



**Fig. 4.** Simulation environment. A satellite image of a rice crop field was taken from Google Earth and a digital elevation model was taken from Google Sketchup. Once the world model is created in gazebosim, the drone is loaded, and the simulated system is ready to work.

captured, the image mosaicking function (stitching) from OpenCV is used to generated the mosaic.

#### 4.1 Image Capture Algorithm

The implemented algorithm assumes that the intrinsic camera parameters (the camera calibration matrix  $\mathbf{K}$  Eq. 1), the size of the camera sensor  $s_x$  and  $s_y$ , and the altitude of the drone, are known. This information is used by the algorithm to estimate the area covered by the camera.

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where  $f_x$  and  $f_y$  are the focal length, in pixel units, in the  $x$  and  $y$  axes; and  $c_x$  and  $c_y$  are the coordinates of the optical center.

From the focal length ( $f_x$  and  $f_y$ ), the size of the camera sensor ( $s_x$  and  $s_y$ ), and the number of pixels per unit distance ( $m_x, m_y$ ), it is possible to estimate the field of view (FOV) of the camera, as shown in Eq. 2; where  $\theta_x$  and  $\theta_y$  correspond to the FOV in each axis.

$$\theta_x = 2 * \tan\left(\frac{s_x/2}{f_x/m_x}\right); \theta_y = 2 * \tan\left(\frac{s_y/2}{f_y/m_y}\right) \quad (2)$$

To estimate the distance covered by the camera in the  $x$  and  $y$  axes ( $\mathbf{d}_{\text{cam}} = \{d_{\text{cam}_x}, d_{\text{cam}_y}\}$ ), Eq. 3 is used. This equation relates the height of the UAV ( $h_{\text{UAV}}$ ) and the FOV of the lens calculated in previous equation ( $\theta_x$  and  $\theta_y$ ), to estimate the covered area, as shown in Fig. 5.

$$\mathbf{d}_{\text{cam}} = 2 * h_{\text{UAV}} * \begin{bmatrix} \tan(\theta_x) \\ \tan(\theta_y) \end{bmatrix} \quad (3)$$

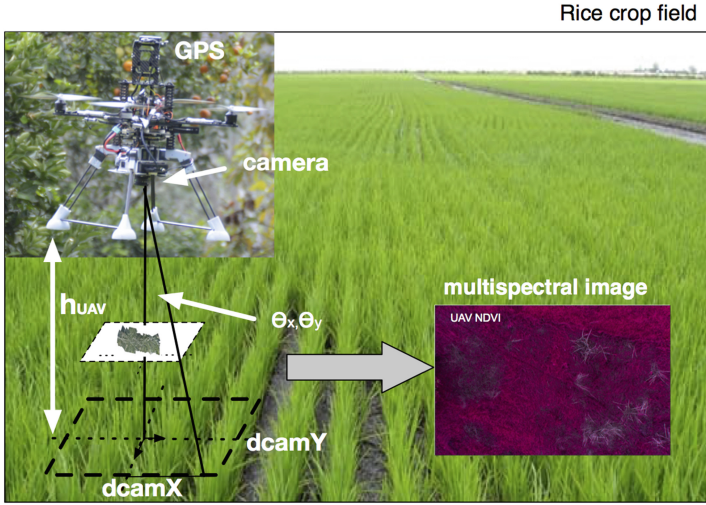
The distance  $\mathbf{d}_{\text{cam}}$  gives the measurement in meters; and the area covered by the camera is given by:

$$a_{\text{cam}} = d_{\text{cam}_x} * d_{\text{cam}_y} \quad (4)$$

In order to calculate the number of pictures required to cover the rice field; and the point where those pictures has to be taken, it is required to know the percentage of the desired overlap between images ( $p_{ov}$ ); the distance covered by the camera, given by Eq. 3; and the area of the crop field, given by the width ( $w_{\text{crop}}$ ) and height ( $h_{\text{crop}}$ ) of the crop field. With these values, Eq. 5 estimates the number of pictures in  $x$  and  $y$  axes that has to be taken in order to cover the crop area.

$$n_x = \frac{w_{\text{crop}}}{d_{\text{cam}_x} - (d_{\text{cam}_x} * p_{ov})} \quad n_y = \frac{h_{\text{crop}}}{d_{\text{cam}_y} - (d_{\text{cam}_y} * p_{ov})} \quad (5)$$

In this application, the percentage of overlapped  $p_{ov}$  tested was 30% in order to ensure that consecutive images have enough common features to create the mosaic.



**Fig. 5.** Area covered by the camera. The height of the UAV and the lens field of view is used to estimate  $d_{cam_x}$  and  $d_{cam_y}$ , see Eqs. 3 and 4

To calculate the places where the images have to be taken ( $\mathbf{x}_{pic}^T = [x_{pic}, y_{pic}]$ ), an iterative model is used. This model takes into account that the origin of the crop coordinate system is at the center of the crop area (the working area); and that the UAV starts in the lower right corner of the working area and moves towards the lower right corner (UAV initial position  $\mathbf{x}_{UAV}(0) = [w_{crop}/2, -h_{crop}/2]$ ).

The initial iteration is:

$$\mathbf{x}_{pic}^T(0) = \mathbf{x}_{UAV}^T(0) + \mathbf{d}_{cam} \begin{bmatrix} -1 + p_{ov} & 0 \\ 0 & 1 - p_{ov} \end{bmatrix} \quad (6)$$

and when iterating, the  $x_{pic}$  or  $y_{pic}$  position is updated as follows:

$$\begin{aligned} x_{pic}(i+1) &= x_{pic}(i) + d_{cam_x} * (-1 + p_{ov}) \\ y_{pic}(j+1) &= y_{pic}(j) + d_{cam_y} * (-1 + p_{ov}) \end{aligned} \quad (7)$$

Once Eq. 7 estimates the coordinates where pictures have to be taken. The UAV is ready to fly and to acquire the images required to generate the mosaic.

During the flight, the only communication with the UAV is to request GPS points and IMU data to integrate, and estimate the position of the UAV related to the starting point of the algorithm. When the UAV is close to the places where the pictures have to be acquired, a flag is activated and the image is acquired.

When the fly ends a node with the OpenCV stitching algorithm is launched and begin the process of generating the mosaic (feature extraction, feature matching, motion estimation and image warping).

## 4.2 Mosaic Generation

To generate the mosaic, a function that uses the stitching class of OpenCV is used. This class contains different functionalities that allows the user to estimate the parameters that relate multiple images. In this particular case, default parameters were used to create the mosaic.

The stitching function resizes the images, finds features, matches them, and finally returns the blend of the images to generate a unique images (the mosaic), that contains all the given images.

The stitching class of OpenCV uses SIFT (Scale-invariant feature transform) algorithm for feature extraction and matching. Feature matching is improve with Bayes's rule. On the other hand, RANSAC is used for adjusting the best homography that aligns the images, and the model of Burt and Adelson [28] is used for blending the mosaic. OpenCV class allows to select different algorithm for the different stages required to create the mosaic.

## 4.3 Algorithm

The developed system for mosaic generation can be summarized in the algorithm shown in Algorithm 1. The algorithm is divided in three sections, two offline and one online.

In the first part (offline), the Image Capture algorithm estimates the positions where the images have to be captured. This is based on information about the crop size, the camera, and the planned UAV's height. The second part occurs online. The UAV moves following a predefined trajectory and when it reaches the positions were images have to be captured, the image is stored. Finally, after the flight has finished, the OpenCV image stitching algorithm is used to generate the mosaic, using the stored images.

# 5 Results

In order to test the image capture algorithm and the mosaic generation for rice crops, different tests were conducted, using the simulation environment create in Gazebo and the simulated drone: Hector quadrotor.

## 5.1 Image Capture Algorithm

The Image Capture algorithm allows estimate how many images have to be taken with some percentage of overlap, in a specific area; and where they have to be taken in order to cover the working area. To test the algorithm, the simulated quadrotor was commanded to fly an area of 50 m<sup>2</sup>, at two different heights 10 m and 15 m. This was done with the aim of analyzing how the image capture algorithm covers the working area.

The camera parameters (see Eq. 1) used in the tests were: a focal length of 40;  $c_x$  and  $c_y$  correspond to the center of the image ( $320 \times 240$ );  $s_x = 39.3$  and

**Offline Image Capture planner (Before Flying);**

**input** : Crop size ( $w_{crop}$  and  $h_{crop}$ ), Camera Sensor Size ( $s_x$  and  $s_y$ ), Drone Height ( $h_{UAV}$ )

**output**: Array of cartesian world points

1. **Compute** Area covered by the camera  $\mathbf{d}_{cam}$ , Eq. 3
2. **Compute** Number of photos  $\mathbf{n}$  in each axis, Eq. 5
3. **Compute** Initial pose  $\mathbf{x}_{pic}^T(0) = [x_{pic}(0), y_{pic}(0)]$ , Eq. 6

```

for  $i \leftarrow 1$  to  $n_x$  do
  for  $j \leftarrow 1$  to  $n_y$  do
    Update  $y_{pic}$  position, Eq. 7;
    Save  $\mathbf{x}_{pic}^T = [x_{pic}(i), y_{pic}(j)]$ ;
  end
  Update position  $x_{pic}$  position, Eq. 7;
  Save  $\mathbf{x}_{pic}^T = [x_{pic}(i), y_{pic}(j)]$ ;

```

**end**

**Online Image Capture (During Flight)**

**input** : initialize motors and sensors reading. Read file with  $\mathbf{x}_{pic}^T$  coordinates

**output**: Stored mages

Take off;

```

for  $i \leftarrow 1$  to  $n_x * n_y$  do
  Move to next pose;
  Read current UAV position  $\mathbf{x}_{uav}^T$  ;
  if if  $x$  and  $y$  coordinates of  $\mathbf{x}_{uav}^T \approx \mathbf{x}_{pic}^T(i)$  then
    Store image;
  else
    Save current pose  $\mathbf{x}_{uav}^T$ 
  end

```

**end**

**Offline Mosaic Generation;**

**input** : Stored images

**output**: Image Mosaic

Initialize  $nPic = n_x * n_y$ ;

```

for  $i \leftarrow 1$  to  $nPic$  do
  Read stored image;
  Append images in an array;

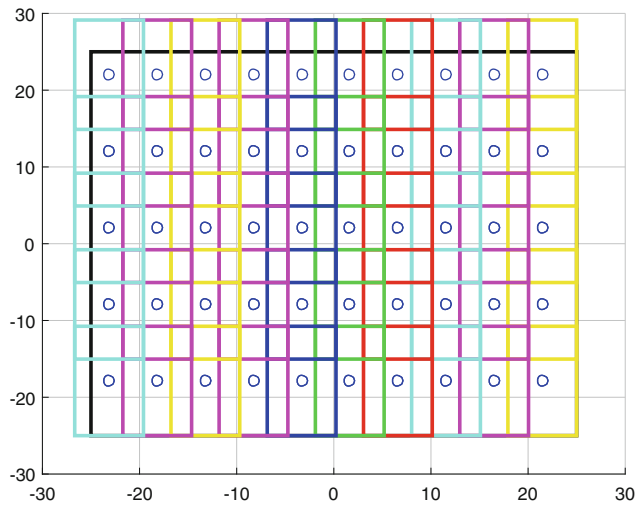
```

**end**

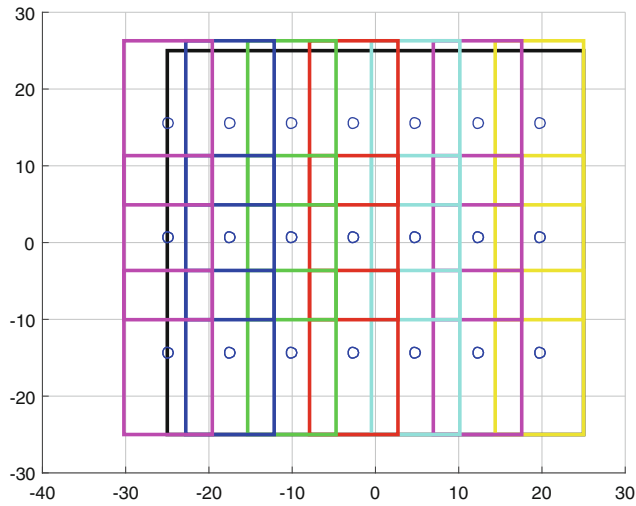
Call OpenCV stitching function;

**Algorithm 1.** General Algorithm. In the first stage (offline), the places where images have to be captured are estimated. The second stage (online) focuses on capturing and storing the images; and the third stage (offline), deals with the creation of the mosaic.

$s_y = 44.5$ ; and  $m_x = 16$  and  $m_y = 10$ . With these parameters, when the UAV flies at 10 m, the algorithm has to take 5 pictures in the Y axis, and 10 in the X axis, in order to cover the working area (see Fig. 6). However, when the UAV flies at  $h = 15$  m, it requires to take less pictures to cover the working area: 3



**Fig. 6.** Mission planing test at 10 m. Working area 50 m<sup>2</sup>



**Fig. 7.** Mission planing test at 15 m. Working area 50 m<sup>2</sup>

in the Y axis and 7 in the X axis (when being higher, the field of view of the camera covers more area).

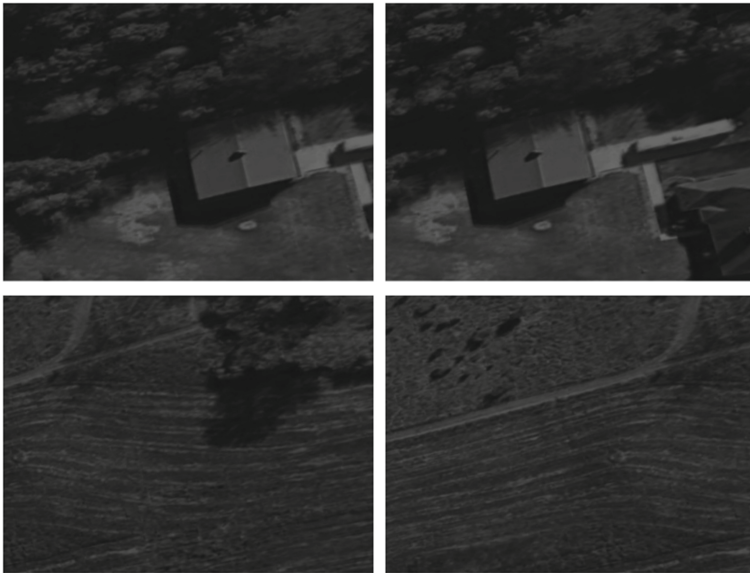
Figures 6 and 7 also show how the percentage of overlap defined (30%) is preserved, and it is possible to see the different grids created by the algorithm. The circles represent the coordinates where images have to be captured, which are used during flight for capturing the different images required to build the mosaic.

## 5.2 Stitching

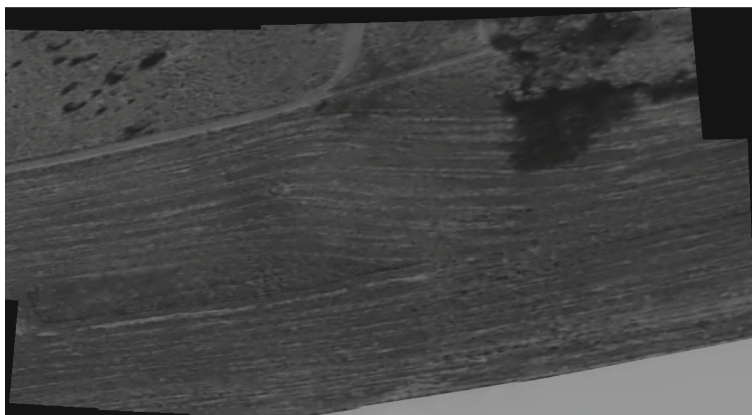
Once the image capture algorithm has decided the places to take pictures, and the UAV has acquired them. The second step is the generation of the mosaic. This is a critical process, especially if it is required to run online on-board the UAV (this is not the case), due to the processing time (high) and the required storage capacity on-board the UAV. For this reason, the image capture algorithm is very important for the system, because with it, it is possible to reduce, in a significant way, the number of images that need to be taken in order to ensure a mosaic that covers the working area. For the tests conducted in this paper, the Stitching class of the OpenCV libraries was used to deal with the mosaic generation problem.

Figure 8 shows some of the images taken by the image capture algorithm, when the UAV flew at 10 m.

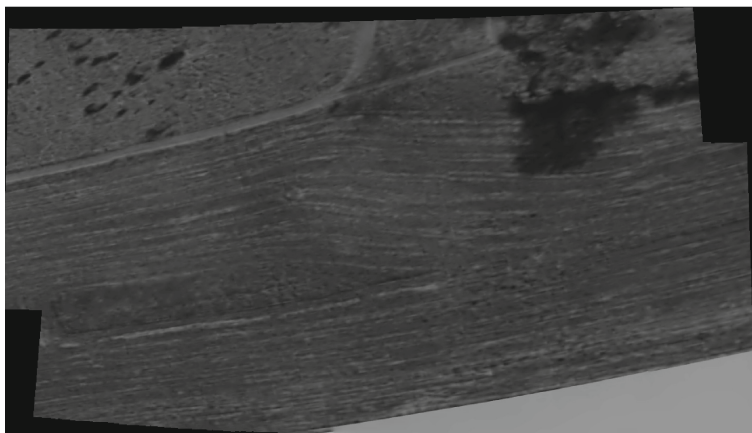
Figure 9 compares two mosaics. One created using images acquired by the image capture algorithm, see Fig. 9a; and the other one created using all the images captured while flying, see Fig. 9b. In the figures, it is possible to see that both mosaics look similarly. However, when using the image capture algorithm, the mosaic shown in Fig. 9a was created using only 6 images and it took 5 min to create the mosaic. However, the second mosaic was created with 19 images (manually captured) and it took 30 min to create the mosaic. Therefore, using the image capture algorithm less pictures are required to cover the working area, and therefore the image stitching process is faster.



**Fig. 8.** Representative images taken using the mission planning algorithm, when flying at 10 m.



(a) With mission planning algorithm. Images 6, processing time 5 min.



(b) Without mission planning algorithm. Images 19, processing time 30 min.

**Fig. 9.** Comparison of two different mosaics created. Using the mission planning algorithm less pictures are required to cover the working area, and therefore the image stitching process is faster.

Finally, Fig. 10 shows another mosaic created using the OpenCV stitching function and the simulation environment created for the project. For this mosaic, the UAV took 10 min to fly around the area, and the stitching algorithm took 85 min to create the mosaic. From this figure, it is possible to see the quality of the mosaics that can be generated with the proposed strategy.

It is important to mention that in this process the GPU was not used; and that depending on the resolution of the images, it is possible to require or not the use of additional filters to eliminate some noise due to the overlapping process.



**Fig. 10.** Mosaic generated after using the image capture algorithm, when flying at 10 m. Processing time 85 min.

## 6 Conclusions

In this paper we have presented an image capture algorithm for ensuring the generation of image mosaics of rice crop fields.

Different tests were conducted using a simulation environment created in ROS. From the tests we have shown the importance of using the image capture algorithm, which allows to reduce, significantly, the number of images required to cover the area of interest, and to reduce the processing time required to generate the mosaic. In the area of UAVs, this is a very important feature taking into account the limited payload capacity on-board these vehicles. It has also been shown that by reducing the amount of images required to cover a specific area, the quality of the results is not affected.

It is important to highlight that with the proposed algorithm, it is possible to generate a mosaic that describes the environment with a small amount of images, but ensuring the resolution and overlap between them required by the image stitching algorithm.

The tests have also shown that with 30% of overlap the stability of the image stitching algorithm was preserved (with this value there were common features in consecutive images to ensure the feature matching process). Nevertheless, current work in the project is focused on conducting real-flight tests to analyze the behavior of the algorithm using multispectral images, and to define with more exhaustive tests the different parameters required by the algorithm, such as the required flight height and the appropriate percentage of overlap. Future work will focus on analyzing the state of the rice crop, based on the multispectral mosaic.

**Acknowledgements.** This work has been funded and sponsored by the research project: Desarrollo de una herramienta para la agricultura de precision en los cultivos de arroz: sensado del estado de crecimiento y de nutricion de las plantas usando un dron autonomo (Aerial sensing and monitoring of rice crop fields applying precision agriculture techniques). Funded by: COLCIENCIAS. ID 120371551916.

## References

1. C. for the Common Organisation of Agricultural Markets, "Rice market," European Commission; AGRICULTURE AND RURAL DEVELOPMENT, Technical report, November 2016
2. Childs, N., Burdett, A.: The U.S. rice export market; rice situation and outlook. Economic Research Service - USDA, Technical report (2000)
3. Chauhan, B., Jabran, K., Mahajan, G.: Rice Production Worldwide, 1st edn. Springer, New York (2016)
4. Mazuera, C.A. Neira, J.D.: ANALISIS DE LOS COSTOS DE PRODUCCION DE ARROZ, Oriza sativa L., EN EL MUNICIPIO DE SALDAÑA, TOLIMA. METODO PULVER VS METODO TRADICIONAL DE MANEJO. Universidad De La Salle Facultad De Administracion De Empresas Agropecuarias, p. 44 (2009)
5. Patino, D., Colorado, J., Navia, J., Mondragon, I.: Multispectral mapping in agriculture: terrain mosaic using an autonomous quadcopter UAV. In: International Conference on Unmanned Aircraft Systems, ICUAS, pp. 1351–1358 (2016)
6. Pajares, G.: Overview and current status of remote sensing applications based on unmanned aerial vehicles (UAVs). *Photogram. Eng. Rem. Sens.* **81**(4), 281–329 (2015). <http://www.sciencedirect.com/science/article/pii/S0099111215300793>
7. Takai, T., Matsuura, S., Nishio, T., Ohsumi, A., Shiraiwa, T., Horie, T.: Rice yield potential is closely related to crop growth rate during late reproductive period. *Field Crops Res.* **96**(2–3), 328–335 (2006)
8. Tripicchio, P., Satler, M., Dabisias, G., Ruffaldi, E., Avizzano, C.A.: Towards smart farming and sustainable agriculture with drones. In: Proceedings of the International Conference on Intelligent Environments, IE, pp. 140–143 (2015)
9. Ramanathan, N., Balzano, L., Burt, M., Estrin, D., Harmon, T., Harvey, C., Jay, J., Koheler, E., Rothenberg, S., Sirvastava, M.: Rapid deployment with confidence: calibration and fault detection in environmental sensor networks. *Commun. ACM* **47**(6), 34–40 (2006). <http://ieeexplore.ieee.org/xpls/absall.jsp?arnumber=1180542>
10. Johnson, L.F., Roczen, D.E., Youkhana, S.K., Nemani, R.R., Bosch, D.F.: Mapping vineyard leaf area with multispectral satellite imagery. *Comput. Electron. Agric.* **38**(1), 33–44 (2003)
11. Shen, K., Li, W., Pei, Z., Fei, W., Sun, G., Zhang, X., Chen, X., Ma, S.: Crop area estimation from UAV transect and MSR image data using spatial sampling method. *Procedia Environ. Sci.* **26**, 95–100 (2015). <http://www.sciencedirect.com/science/article/pii/S1878029615001747>
12. Tri, N.C., Hoai, T., Duong, H.N., Trong, N.T., Vinh, V., Snasel, V.: A novel framework based on deep learning and unmanned aerial vehicles to assess the quality of rice fields. In: Akagi, M., Nguyen, T.-T., Vu, D.-T., Phung, T.-N., Huynh, V.-N. (eds.) ICTA 2016. AISC, vol. 538, pp. 84–93. Springer, Cham (2017). doi:[10.1007/978-3-319-49073-1\\_11](https://doi.org/10.1007/978-3-319-49073-1_11)
13. Goodman, R., I.U. Ltd.: Too many sensors not enough receptors. *The Ranger J. Defence Surveyor Assoc.* **1**(1) (2011)
14. Kang, Z., Zhang, L., Zlatanova, S.: An automatic mosaicking method for building facade texture mapping. *ISPRS*, vol. XXXVI (2014)
15. Shum, H.Y., Szeliski, R.: Construction of panoramic image mosaics with global and local alignment. *IJVC* **36**, 101–130 (1998)
16. Ulman, V.: Mosaicking of high resolution biomedical images. Ph.D. dissertation, Masaryk University, Brno (2005)

17. Guo, T., Kujirai, T., Watanabe, T.: Mapping crop status from an unmanned aerial vehicle for precision agriculture applications. In: ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXIX-B1, August 2012
18. Valente, J., Guillen, A., Manrique, O., Arenal, M.: On the acquisition of high resolution maps with open source software and commercials off the shelf quadrotors. In: Gonzalez, P., Ribeiro, A. (eds.) Second International Conference on Robotics and Associated High Technologies and Equipment for Agriculture and Forestry New Trends in Mobile Robotics Perception and Actuation for Agriculture and Forestry (2014)
19. Primicerio, J., Gennaro, S.F.D., Fiorillo, E., Genesio, L., Lugato, E., Matese, A., Vaccari, F.P.: A flexible unmanned aerial vehicle for precision agriculture. *Prec. Agric.* **13**(10), 517–523 (2012)
20. Botterill, T., Mills, S., Green, R.: Real-time aerial image mosaicing. In: International Conference on Image and Vision Computing, New Zealand (2010)
21. Moussa, A., Sheimy, N.E.: A fast approach for stitching of aerial images. In: ISPRS, vol. XLIB3, July 2016
22. Hussain, R.Y.: Production of digital mosaics from aerial images. *J. Eng. Devel.* **17**(4) (2013)
23. Claudia, A., Porto, S.M.: Classification of crop-shelter coverage by RGB aerial images a compendium of experiences and findings. *J. Agric. Eng.* **41**(3), 1–11 (2010)
24. Di Franco, C., Buttazzo, G.: Coverage path planning for UAVs photogrammetry with energy and resolution constraints. *J. Intell. Rob. Syst. Theory Appl.* **83**(3–4), 445–462 (2016)
25. Hawary, A.F., Chipperfield, A.J.: Routeing strategy for coverage path planning in agricultural monitoring activity using UAV. In: International Congress on Recent Development in Engineering and Technology, RDET (2016)
26. Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., Stryk, O.: Comprehensive simulation of quadrotor UAVs using ROS and Gazebo. In: Noda, I., Ando, N., Brugali, D., Kuffner, J.J. (eds.) SIMPAR 2012. LNCS (LNAI), vol. 7628, pp. 400–411. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-34327-8\\_36](https://doi.org/10.1007/978-3-642-34327-8_36)
27. Gazebo, “GazeboSim.org.” <http://gazebo.org/>. Accessed: 15 Sep 2016
28. Adelson, E.H., Burt, P.J.: A multiresolution spline with application to image mosaics. *ACM Trans. Graph.* **2**(4), 217–236 (1983). <http://dl.acm.org/citation.cfm?id=245.247>