International Journal of Research in Engineering, Science and Management
Volume 4, Issue 4, April 2021
https://www.ijresm.com | ISSN (Online): 2581-5792

Aircraft Visual Damage Inspection with Autonomous Drone and Image Processing

C. Vimal Jothi[1], S. Gokulnath[2], K. Vetrivel[3], H. Balaji[4*], D. P. Sujay Iswarian Issac[5]
[1,2,3,4,5] *Department of Aeronautical Engineering, GKM College of Engineering and Technology, Chennai, India*

***Abstract*: The drone-based aircraft visual inspection system is optimized for inspecting the surface of the whole aircraft. The drone is equipped with ultrasonic sensors capable of detecting obstacles and halting the inspection if needed. The drone is capable of flying with and without the need for remote piloting. The drone follows a predefined inspection path and the autonomous drone captures all the required images with its on board camera. The implementation of the drone will reduce the overall inspection time. High-quality pictures are then transferred to a pc database for detailed analysis using a software system. It measures visual damage on an aircraft's surface by image processing. Finally, the software generates an inspection report.**

***Keywords*: Autonomous drone, damage, image-processing, inspection, sensors.**

1. Introduction

The international air transport association safety found that approximately 20 percent of aircraft accidents are caused by machine failures and 80 percent of aircraft accidents are due to human error (pilots, air traffic controller mechanics visual inspectors et cetera).

Normal aircraft visual inspection has been accomplished from the ground and it could last up to a day. But this drone-based aircraft visual inspection takes approximately 2 to 3 hours to inspect a whole aircraft. The collection of captured images is then sent to a computer and processed through image processing. The image processing is carried out by python-open CV (for crack detection) and mat-lab (for dents detection) and the output has been verified.

Generally, aircraft surface experience cracks (stressed corrosive cracks, brittle cracks et cetera) dents, scratches, etc. In these damages, the cracks are initiated by stressed corrosive cracks and brittle cracks in which SCC is induced and developed with the combination of tensile stress and corrosion in a particular part of the aircraft, in the brittle crack the aircraft surface is exposed in low temperature for a particular period of time. It is a combination of concentrated stresses and cold environment.

2. Selection and Assembly of Drone Parts

The selection of drone parts plays an important role as the drone is used for aircraft visual inspection. So, it must be kept in mind while selecting the parts. Initially, the size of the drone is selected as per our convenience. Thus F450 quadcopter frame was selected. And respective Motors, ESC (Electronic Speed Controllers), PM (Power Module), Propellers, Battery, Transmitter, and Receiver were selected. The APM 2.8 flight controller (in-build compass) is used for the inspection drone as it can be used with and without a remote pilot (with the help of a mission planner and GPS).

Initially, the assembling is started by soldering the power module with the PDB (Power Distribution Board) and soldering the ESC with PDB, next the respective motors are soldered with ESC as per clockwise and counter-clockwise orientation. And the PDB is attached with the F450 quadcopter frame. Following that, APM 2.8 flight controlled is mounted at the center of the frame, the GPS is connected with the APM 2.8 flight controller.

The APM 2.8 flight controller is mounted at the center of the drone and the Receiver, ultrasonic sensor, GPS is connected to the APM 2.8 flight controller. Then the FPV camera is clamped at the center bottom of the drone.

3. Mission Planner and Initial Setup

After integrating the component with the frame the APM 2.8 flight controller is connected to a laptop (to install firmware) via USB. In mission planner software install the corresponding firmware (quad rotor) to the flight controller following that Acceleration calibration, Compass, Radio calibration ESC calibration is done. In order to fly the drone automatically, coordinates of the flying path must be fed into the APM 2.8 flight controller via mission planner.

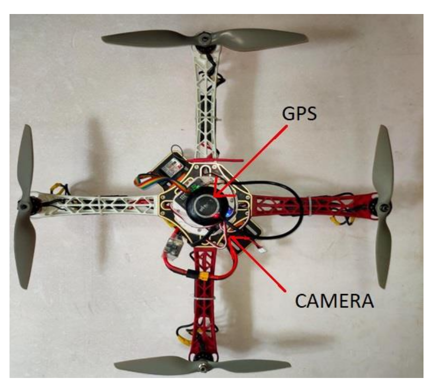

Fig. 1. Damage visual inspection Drone

*Corresponding author: balaji.hk007@gmail.com

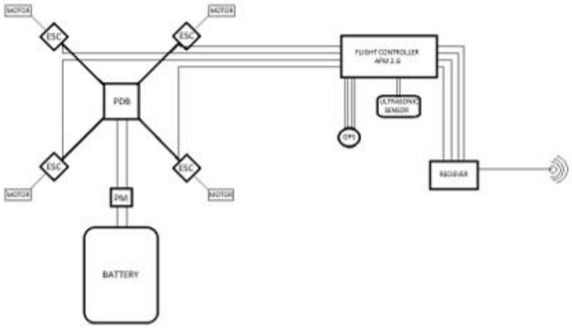

Fig. 2. Typical diagram of drone

4. Image Capturing

After feeding the predefined path coordinates of the flight to the drone, the drone is ready to fly. The drone automatically flies in the predefined path and captures the image which is needed to be inspected. After that, the drone transmits the images to a computer database. During the capturing process, the drone avoids the obstacles with the help of ultrasonic sensors. After finishing the process, the drone automatically comes to its landing position.

5. Image Processing

Image processing is implied in many fields such as gamma-ray imaging, x-ray imaging, detection of cracks and dent et cetera.

It is carried out by two software:

- Python OpenCV
- Matlab

1) Python Open CV

Open CV is a computer vision library that can be used with python for image processing machine learning, deep learning. Therefore, the python Open CV is used in this image processing to find crack detection in aircraft surface

Python Opencv Program:

```
import cv2
import math
import numpy as np
import scipy.ndimage
def orientated_non_max_suppression(mag, ang):
    ang_quant = np.round(ang / (np.pi/4)) % 4
    winE = np.array([[0, 0, 0],[1, 1, 1], [0, 0, 0]])
    winSE = np.array([[1, 0, 0], [0, 1, 0], [0, 0, 1]])
    winS = np.array([[0, 1, 0], [0, 1, 0], [0, 1, 0]])
    winSW = np.array([[0, 0, 1], [0, 1, 0], [1, 0, 0]])
    magE = non_max_suppression(mag, winE)
    magSE = non_max_suppression(mag, winSE)
    magS = non_max_suppression(mag, winS)
    magSW = non_max_suppression(mag, winSW)
    mag[ang_quant == 0] = magE[ang_quant == 0]
    mag[ang_quant == 1] = magSE[ang_quant == 1]
    mag[ang_quant == 2] = magS[ang_quant == 2]
    mag[ang_quant == 3] = magSW[ang_quant == 3]
    return mag
def non_max_suppression(data, win):
        data_max = scipy.ndimage.filters.maximum_filter(data, footprint=win, mode='constant')
        data_max[data != data_max] = 0
        return data_max
Start Calculation
gray_image = cv2.imread(r' INSERT FILE LOCATION PATH ', 0)
with_nmsup = True (Apply Non-Maximal Suppression)
fudgefactor = 1.3 (For Error or uncertainty)
sigma = 21 (For Gaussian Kernel)
kernel = 2*math.ceil(2*sigma)+1 (Kernel Size)
gray_image = gray_image/255.0
blur = cv2.GaussianBlur(gray_image, (kernel, kernel), sigma)
gray_image = cv2.subtract(gray_image, blur)
(Compute Soble Response)
sobelx = cv2.Sobel(gray_image, cv2.CV_64F, 1, 0, ksize=3)
sobely = cv2.Sobel(gray_image, cv2.CV_64F, 0, 1, ksize=3)
mag = np.hypot(sobelx, sobely)
ang = np.arctan2(sobely, sobelx)
(Threshold)
threshold = 4 * fudgefactor * np.mean(mag)
mag[mag < threshold] = 0
(Either Get Edges Directly)
if with_nmsup is False:
    mag = cv2.normalize(mag, 0, 255, cv2.NORM_MINMAX)
    kernel = np.ones((5,5),np.uint8)
    result = cv2.morphologyEx(mag, cv2.MORPH_CLOSE, kernel)
    cv2.imshow('im', result)
    cv2.waitKey()
(Or Apply A Non-Maximal Suppression)
else:
    (Non-Maximal Suppression)
    mag = orientated_non_max_suppression(mag, ang)
    # create mask
    mag[mag > 0] = 255
    mag = mag.astype(np.uint8)
    kernel = np.ones((5,5),np.uint8)
    result = cv2.morphologyEx(mag, cv2.MORPH_CLOSE, kernel)
    cv2.imshow('im', result)
    cv2.waitKey()
```

Steps Involved in Crack Detection Python Program:

At first, remove the in-homogeneous background illumination. Then apply a big Gaussian blur and subtract the original image from the blurred one. After subtracting the image apply Sobel edge detection and morphological close. Next, remove the noise in the image using blob analysis. To get better results you can apply a non-maximal suppression after Sobel.

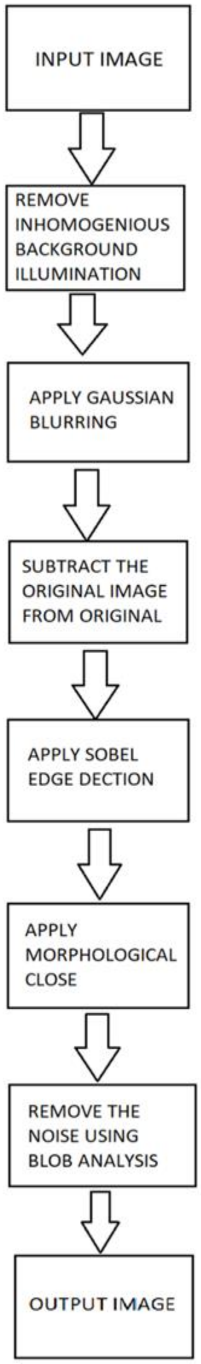

Fig. 3. Python program flowchart

6. Results

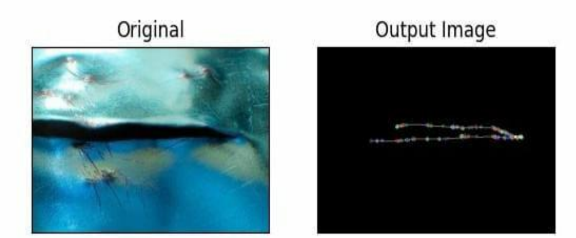

Fig. 4. Crack detection using python (Result 1)

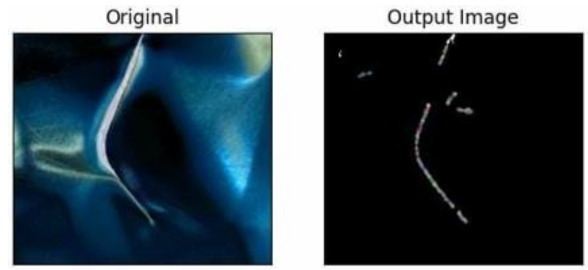

Fig. 5. Crack detection using python (Result 2)

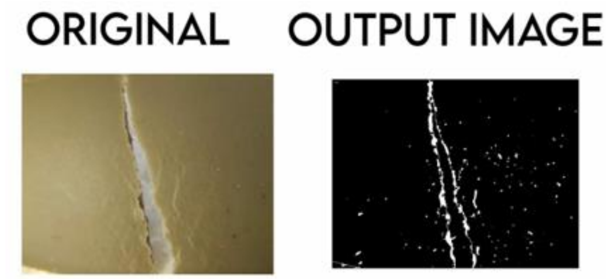

Fig. 6. Crack detection using python in fiber surface (Result 1)

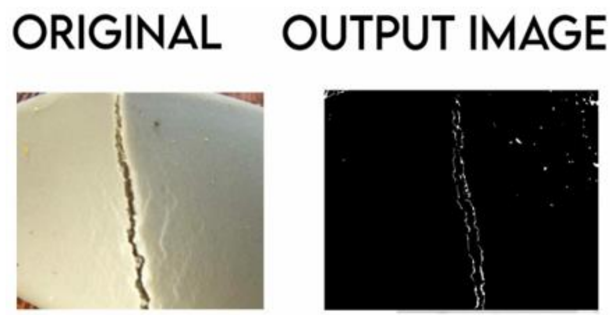

Fig. 7. Crack detection with python in fibre surface (Result 2)

A. MATLAB

Matlab, a truncation for 'Network Research Facility,' is a stage for taking care of numerical and logical issues. It is a restrictive programming language created by math works, permitting grid controls, capacities and information plotting, calculation execution, UI creation and interfacing with programs sent in programming dialects like c, c++, java, etc.

It very well may be utilized to perform picture division, picture upgrade, commotion decrease, mathematical changes, picture enrollment, and 3d picture handling activities. Large numbers of the IPT capacities support c/c++ code age for work area prototyping and inserted vision framework arrangement.

It gives a complete arrangement of reference-standard calculations and work process applications for Picture Handling, Examination, Perception, and Calculation Advancement. Here this software is used to detect dents and scratches on aircraft surfaces.

Steps Involved in Matlab Program:

- The given program study the input image and converts RGB to gray level.
- Then it computes the image intensity.
- It evaluates the value of pixel along y coordinate and computes the pixel value along x coordinate.
- And returns the gradient magnitude value and displays

the magnitude value set and returns the visibility of the object and save.

- After that, the program reads the input image and deletes the filename.
- It returns the array and column values of pixels.
- Then it returns the minimum and maximum values of the array.
- Following that, the program specifies the crop region and resizes it to a two-dimension image.
- Displays the image and returns the visibility of the object and appending the blur in file.
- Next, it reads the input of the image and deletes the file name, and removes connected.
- Thereafter it fills the holes in the input binary image.
- And returns the row and column coordinate of border pixel and displays the output and determines the size of the boundary.
- For loop declaration, declaring boundary values and plots the values having lines width.
- Then assesses the convex hull of all objects and returns the label matrix l that contains labels for the 8-connected objects.
- Following that returns the centroid of the array.
- In the end, it plots the value for y for every value of x.

B. Result

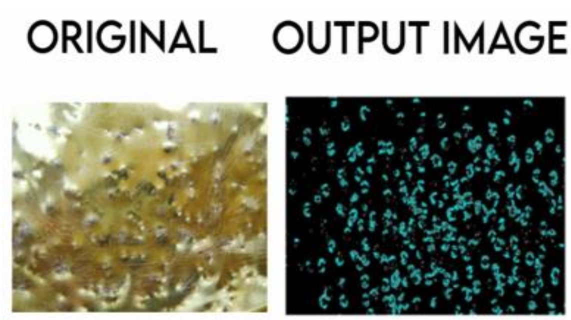

Fig. 8. Dent detection using Matlab

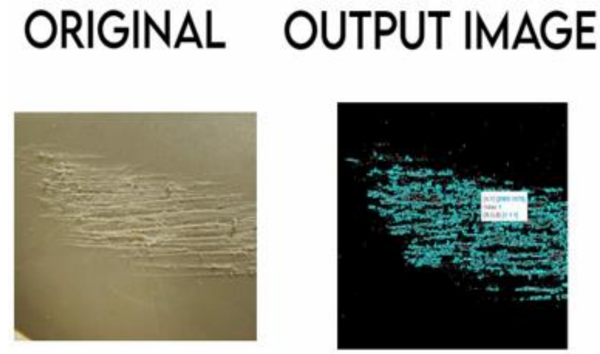

Fig. 9. Scratch detection using Matlab

7. Conclusion

The aircraft visual inspection with the inspection drone and image processing reduces the time when comparing with manual inspection. As it also reduces the manpower and workload. The overall time of the inspection takes approximately up to three to 4 hours. While the manual inspection takes nearly a day. Hence this process is more convenient for this operation. However, at some time deep learning and machine learning can be implemented in the form of image processing which can fasten and facilitates the quality of the inspection.

References

[1] Henk Freimuth, and Markus Konig, "Planning and Executing Construction Inspevtion with Unmanned Aerial Vehicles", in *Automation in Construction*, vol. 96, pp. 540-553, Dec. 2018.

[2] Junwon Seo, Luis Duque, Jim Wacker, "Drone-Enabled Bridge Inspection Methodology and Application", in *Automation in Construction*, vol. 94, pp.112-126, Oct. 2018.

[3] Yinan Miao, Jun Young Jeon, Gyuhae Park, "An Image Processing-Based Crack Detection Technique for Pressed Panel Product", in Journal of Manufacturing System, vol. 57, pp. 287-297, Oct. 2020.

[4] Tarek Rakha, Alice Gorodetsky, "Review of unmanned Aerial System (UAS) Application in The Built Environment: Towards Automated Building Inspection Procedures Using Drones", in *Automation in Construction*, vol. 93, pp. 252-264, Sept. 2018.

[5] Gerard Gibbs, Huamin Jia, And Irfan Madani, "Obstacle Detection with Ultrasonic and Signal Analysis Metrics", in *Transportation Research Procedia*, vol. 28, pp. 173-182, 2017.

[6] Juan S. Guerrero Guerrero, Aldo F. Contreras Gonzalez, Jose I. Herandez Vega, And Leticia A. Neira Tovar, "Instrumentation of an Array of Ultrasonic Sensors and Data Processing for Unmanned Aerial Vehicle (UAV) For Teaching the Application of the Kalman Filter", in *Procedia Computer Science*, vol. 75, pp. 375-380, 2015.

[7] Ling Li, Meng Gong, Y. H. Chui, And Marc Schneider, "A Matlab Based Image Processing Algorithm for Analyzing Cupping Profiles of Two-Layer Laminated Wood Products" in *Measurement*, vol. 53, pp. 234-239, July 2014.

[8] Yun Wang, Ju Yong Zhang, Jing Xin Liu, Yin Zhang, Zhi Ping Chen, Chun Guang Li, and Rui Bin Yan, "Research On Crack Detection Algorithm of the Concrete Bridge Based On Image Processing," in *Procedia Computer Science*, vol. 154, pp. 610-616, 2019.