VISION-BASED OBJECT TRACKING FOR AN UAV ROTORCRAFT USING ROBOT OPERATING SYSTEM (ROS)

Fatimah Mohd Nizam, Norsinnira Zainul Azlan*

Department of Mechatronics,
Faculty of Engineering,
International Islamic University Malaysia

fatimahnizam96@gmail.com, sinnira@iium.edu.my

Article history

Received: 10 May 2024

Received in revised form: 25 June 2024

Accepted: 27 June 2024

Published online: 30 June 2024

*Corresponding author: sinnira@iium.edu.my

ABSTRACT

In agricultural activities, it is difficult to monitor and protect the crops in a large open space. This leads to the need of a high number of workforces to perform the task. This problem may be alleviated with the application UAVs with machine vision feature to track any intruders, including thieves and any animals that may destroy the crops such as wild boars. This paper presents the implementation of Robot Operating System (ROS) for object tracking in Unmanned Aerial Vehicles (UAV) rotorcraft. Six object trackers in the OpenCV library including BOOSTING Tracker, Multiple Instant Learning (MIL) Tracker, Kernelized Correlation Filter (KCF) Tracker, Tracking, Learning and Detection (TLD) Tracker, Medianflow Tracker and Minimum Output Sum of Squared Error (MOSSE) Tracker, are tested and compared to determine the best object tracker. The efficiency of the object trackers is determined by its frame per second (FPS) and its ability to keep on tracking the object continuously. The results show that MOSSE provides the best performance and it has been programmed on the Raspberry Pi and then attached to the drone to perform the object tracking task. The outcome verifies that the UAV is able to perform the object tracking task as desired using Robot Operating System (ROS).

***Keywords**: Unmanned Aerial Vehicles (UAV), Drone, Robot Operating System (ROS),Object Tracker*

1.0 INTRODUCTION

UAV rotorcraft or drone is a cluster of small aircraft technology that operates in the absence of pilot onboard. UAVs are going through a phase of a global boom because of their growing accessibility and functionalities in recent years. UAV rotorcraft could be applicable from agricultural sector to the hazardous sector such as the military and defense. UAV rotorcraft are mainly used for inspection and monitoring, mostly in an impossible and inaccessible place for human to act and in a continuous time-set action. Positioning an UAV relative to another object is a difficult task to execute.

One of the ways to enable a UAV rotorcraft to track and maintain a desired relative location is by using monocular vision system [1]. Monocular vision system is utilized to track a stable and static object. Monocular vision in terms of human sight means both eyes are used separately. In UAV, the monocular vision is used to obtain continuous surrounding images. The on-board vision system camera plays an essential role in executing the monocular vision tracking task. The Connected Component Labeling algorithm is used for the object detection. It is a very efficient blob finding algorithm with the utilization of OpenCV library. HSV color coordinated are considered in this method as they are robust to lighting changes. Monocular vision could also be used in SLAM-based navigation [2]. Another method for UAV object tracking is by using real-time motion sensing [3]. The position of the mobile object is estimated using VICON motion capture camera and the information obtained is processed by a locally developed MATLAB Simulink code. The VICON system is composed of 12 infrared cameras that are allocated around the object. It is used for motion tracking of reflective tapes. The VICON tracker designed constructs and triangulates the 3D position of the object by reading the capture cues received from the camera. Hough Transform Computer Vision algorithm could also be implemented for object tracking [4]. The limitation of using this algorithm is that it can detect and track an object with a certain shape only. A specific tag is also used and identified in advance by the UAV to ease the UAV tracking task.

DJI Phantom 3 is an example of the existing UAV technology, equipped UAV aerodynamic, circuit boards, chipset and software [5]. ROS has been applied to the UAV. ROS has a framework that is capable to gather different software tools with different type of software languages, applying them into one robot without reinventing the language wheel [6]. ROS is a robotic open source middleware that is designed to be compatible with other writing robotic software [7]. It is developed to ease the programming of the robots by allowing all different types of robotic software to transfer and receive data by communicating with each other. The principle of nodes is used in this operating system. Nodes are all combined into a graph and communicate with one another using streaming topics. The nodes can be created by using Python, C++, Java Script and other programming languages. The topics are stream of messages, which the nodes choose while classifying them as publishers or subscribers of the topics. It manages the program, the speed and whom is receiving (subscriber) and transferring (publisher) the information. All nodes could work as both publishers and subscribers at once.

Object tracking is a process of allocating an object in successive frames of video with accuracy and speed detection during the process. OpenCV is used for object tracking in the research. The object trackers that are available in the OpenCV library [8] include:

1. BOOSTING Tracker
2. Multiple Instant Learning (MIL) Tracker
3. Kernelized Correlation Filter (KCF) Tracker
4. Tracking, Learning and Detection (TLD) Tracker
5. Medianflow Tracker
6. Goturn Tracker

PERINTIS eJournal, 2024, Vol. 14, No. 1, pp. 25-37

7. Minimum Output Sum of Squared Error (MOSSE) Tracker
8. Discriminative Correlation Filter with Channel and Spatial Reliability (DCF-CSR)

In this study, only six object trackers will be tested: KCF, MOSSE, Medianflow, MIL, TLD and BOOSTING trackers. Mission Planner software is used to set up the initial setting of the UAV such as the installing the correct firmware, speed and compass. It is used as the ground station for the UAV [9].

The rest of the paper is organized as follows. The UAV rotorcraft mechanical design and circuit, ROS architecture and serial communication between Raspberry Pi and Ardupilot are described in Section 2. Section 3 presents the results obtained and discussion. Finally, conclusion is drawn in Section 4.

2.0 UAV ROTORCRAFT

2.1 Mechanical Design

The design of the UAV or quadcopter is shown as in Figures 1 and 2. It consists of a single stand base that will support the Raspberry Pi, Ardupilot Mega APM 2.8 board and the Lipo battery. Brushless motor with an ESC controller is used to uplift the UAV. The Raspberry Pi is powered up separately from the Ardupilot Mega 2.8 board and the brushless motors to avoid high current drawn from the brushless motors. Four landing gears are used to stabilize the UAV before the take off and for a smooth landing.

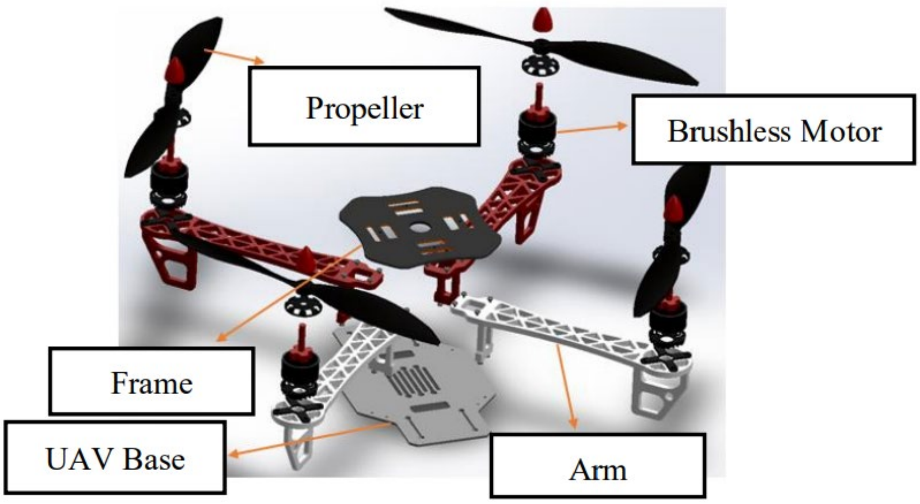

Figure 1. Mechanical design of quadcopter UAV

PERINTIS eJournal, 2024, Vol. 14, No. 1, pp. 25-37

Figure 2. Quadcopter mechanical design

The direction of each propeller of the UAV is illustrated as in Figure 3. Each of the propeller must be interact with its opposite propeller by rotating in the opposite direction. Clockwise propeller's direction is known as pushers and counterclockwise propeller's direction is known as pullers. A and C are the front propellers while B and D are the back propellers, which act in the directions that are against to each other.

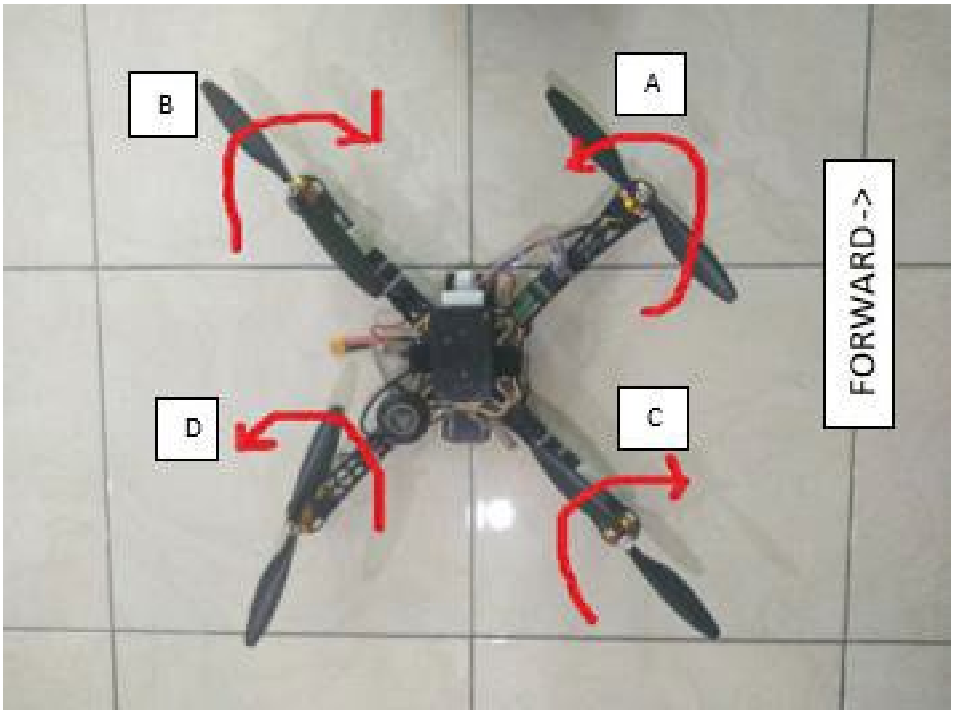

Figure 3. Indication of each propeller's direction

2.2 Electrical Design

The UAV electrical components and their connection are shown in Figures 4 and 5. Ardupilot Mega 2.8 board is used as the computing board to control the brushless motors through ESC. The brushless motors are connected parallel by soldering them on the main frame of the quadcopter (UAV). The internal compass inside the Ardupilot Mega 2.8 board is used to indicates the orientation of the UAV. The raspberry pi is utilized for the computer vision task. A Picamera is attached to perform the object tracking process. An external GPS module is used to determine the location of the UAV (quadcopter). This plays a role when path planning task is executed. A receiver is used to obtain signal from the FlySky controller to control the UAV manually for emergency and other necessary tasks.

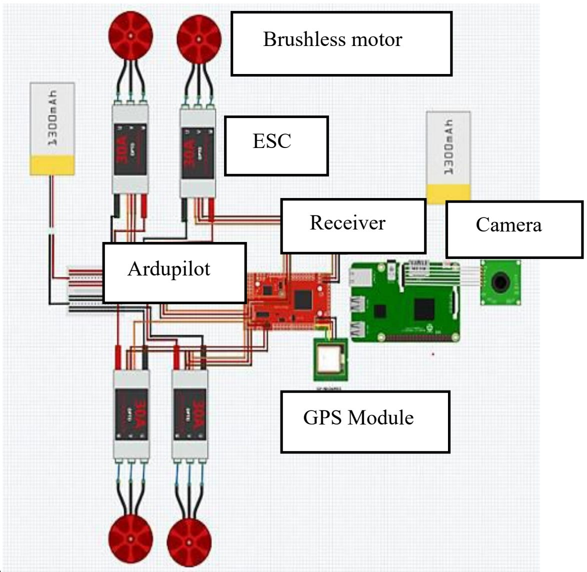

Figure 4. Electrical components connection in UAV

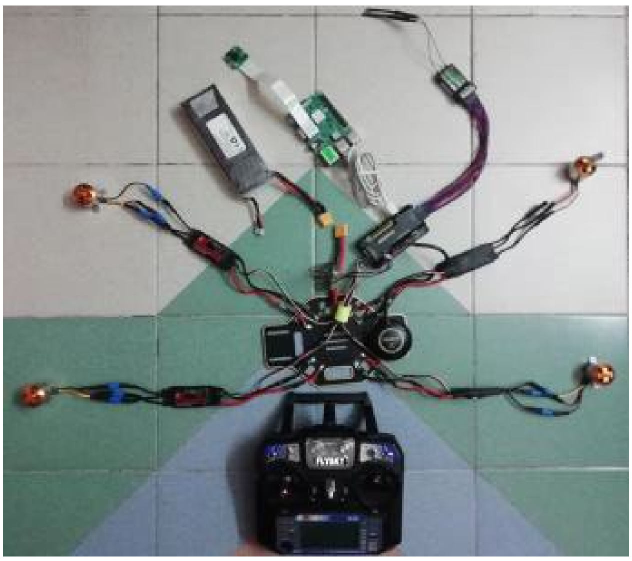

Figure 5. Electrical components in the UAV

2.3 Software, Libraries and Operating System

ROS is used as the middleware to communicate between different type of software languages such as Python and C++. The OpenCV library is used for the object tracking. Ubiquity Robotics image with Ubuntu 16.04 operating system (OS) and ROS pre-installed is used for the Raspberry Pi. This image is flashed in a Micro SD card before inserted in the Raspberry Pi.

2.4 Basic ROS Architecture

Figure 6 shows the basic ROS Architecture. The first layer of ROS consists of a package. The package consists of nodes that consists of topics, which will determine whether the nodes will be a publisher or a subscriber. It is executable even when the nodes are written in different type of software languages.

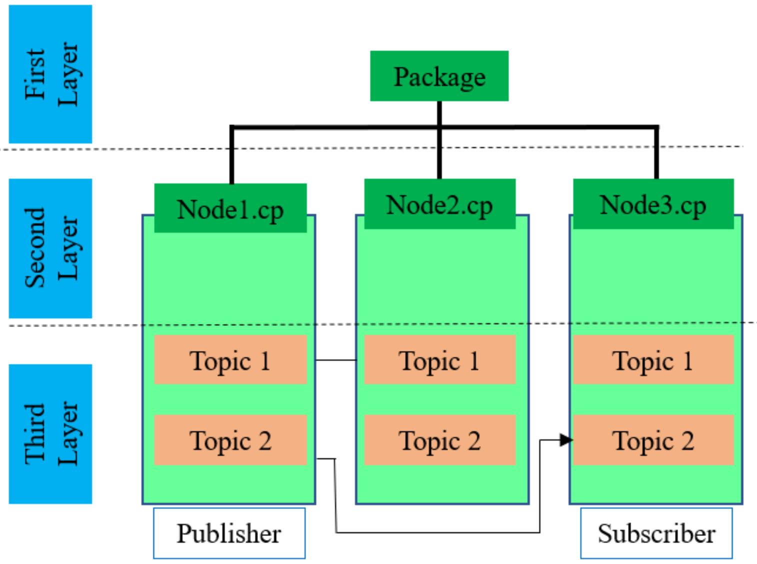

Figure 6. Basic ROS architecture

2.5 UAV ROS Architecture

Figure 7 shows the UAV ROS Architecture. The opencv_object_tracking node contains the calls the MOSSE OpenCV object tracker to create a bounding box around the desired object. The bounding box will then detect and follow the desired object while the frame per second is measured in real time. The opencv_object_tracking node subscribes the topic inside MAVROS node to utilize the instructions used to control the brushless motors' directions. The MAVROS node will publish the topic to other nodes available.

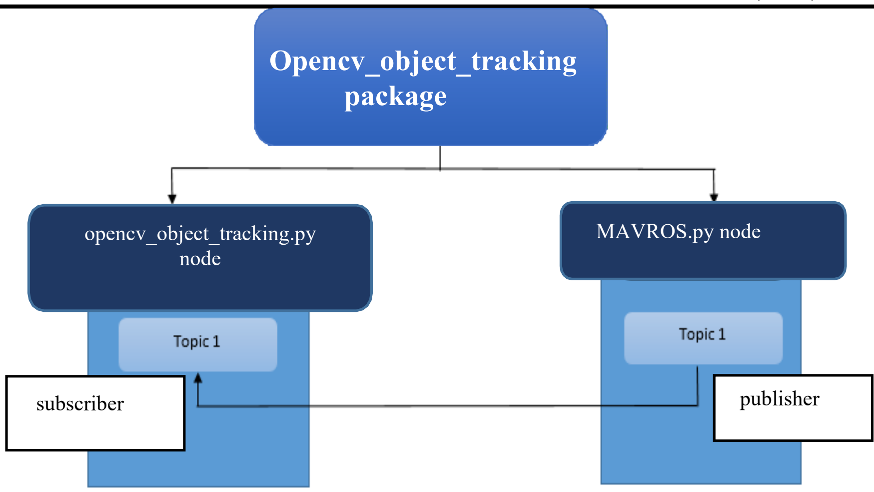

Figure 7 . UAV ROS architecture

2.6 Serial Communication between Raspberry Pi and Ardupilot using MAVROS (MAVlink protocol)

MAVROS is a ROS UAV software package that allows ROS nodes to communicate with the Ardupilot based on MAVLink based systems. MAVlink stands for Micro Air Vehicle Link. It is the protocol that is used for communication with unmanned vehicle. The connection between the raspberry pi and Ardupilot Mega 2.8 board is shown in Figures 8 and 9.

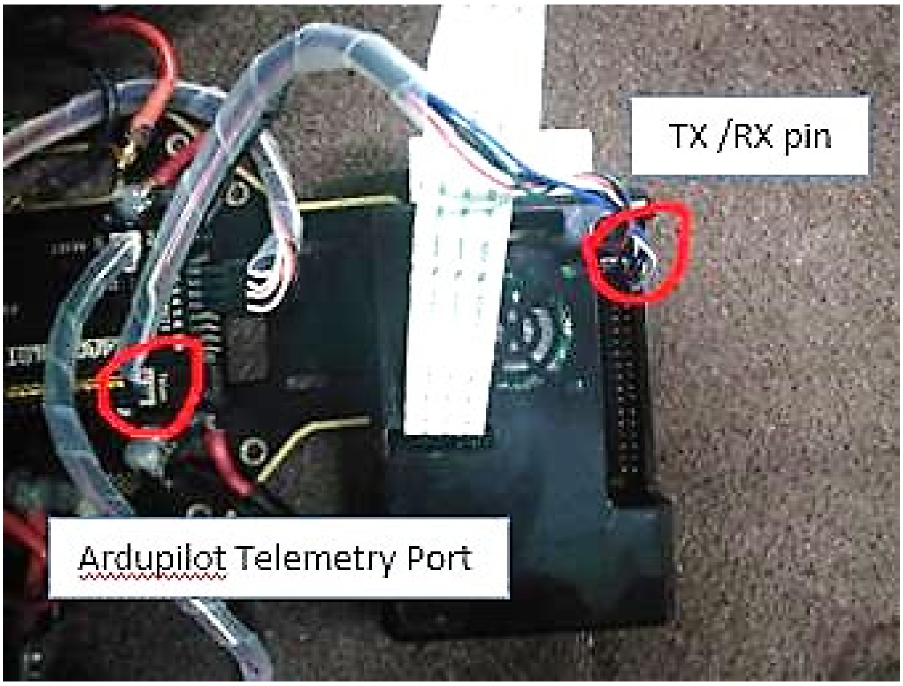

Figure 8. Connection between the Ardupilot Mega 2.8 board and the Raspberry Pi

PERINTIS eJournal, 2024, Vol. 14, No. 1, pp. 25-37

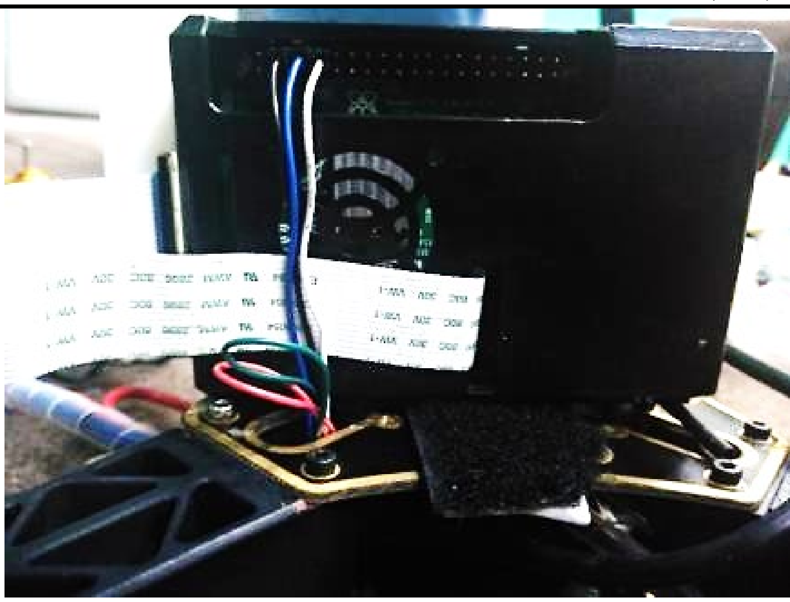

Figure 9. GPIO pin connection on the Raspberry Pi

The telemetry port at the Ardupilot APM 2.8 board is connected to the Raspberry Pi GPIO pin. The 5V port is disconnected for the MAVlink protocol to operate. The Raspberry Pi and the Ardupilot need to have a separate power supply respectively.

When the connection is successful, the Raspberry Pi will have control over the brushless motors. There will be no need for the transmitter, the FlySky controller to send signals to the receiver to operate the brushless motors. It can give instructions to the brushless motors such as arming the throttle, take-off and landing. The example is shown in Figures 10 and 11.

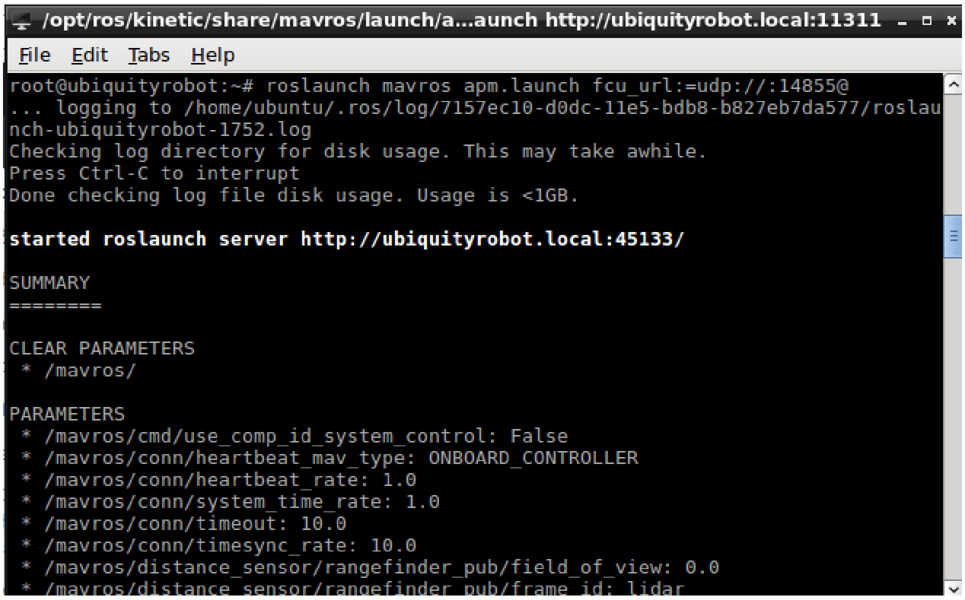

Figure 10. MAVROS node is launched

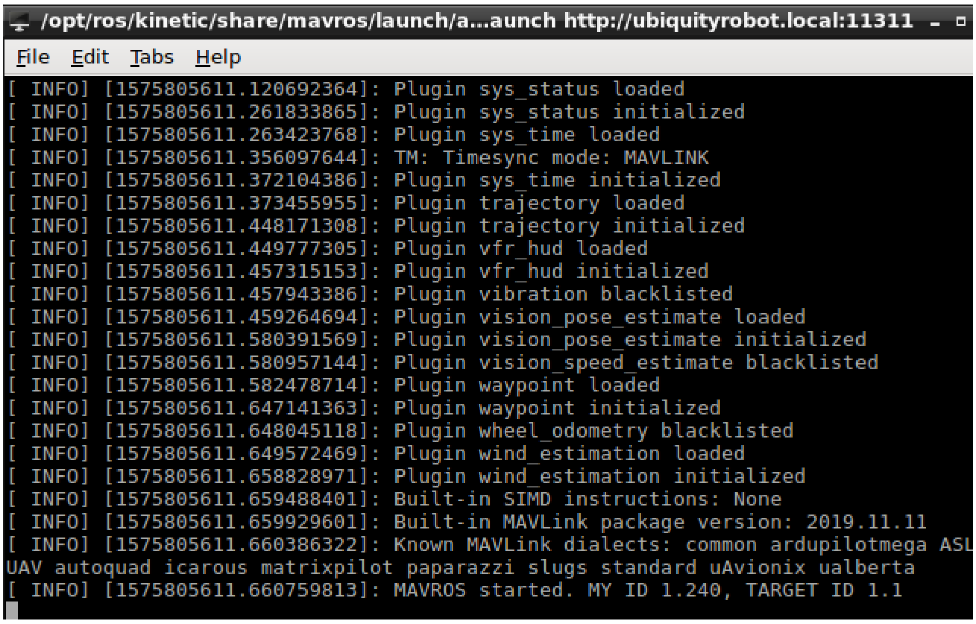

Figure 11. MAVROS node is successfully initialized

2.7 Object Tracking: Minimum Output Sum of Squared Error (MOSSE)

Robust filters are needed for visual tracking activities. MOSSE filter is one of the stable filters [11]. It is known as a filter-based tracker that model the structure of objects using filters trained on example images. In the beginning, a small window is set at the center of the object in the first frame so that the tracker can be initialized. In the in the next frame, the targeted object is tracked by correlating the filter over a search window. The new position of the targeted object will be indicated as the location with the highest correlation value.

The correlation in MOSSES is calculated based Fast Fourier Transform (FFT). The first step is to compute the 2D Fourier transform of the input image: $F = \mathcal{F}(f)$, and the filter: $H = \mathcal{F}(h)$. Based on convolution theorem, [11]

$$G = F \, \Theta \, H^* \quad (1)$$

where G is the trained output, Θ is the correlation or element-wise multiplication in the Fourier domain and * indicates the complex conjugate of the equation. In MOSSE filter, the training images, F_i and the trained outputs, G_i are generously considered. From Equation (2) [11],

$$H_i^* = \frac{G_i}{F_i} \quad (2)$$

MOSSE implements a filter, H to minimize the sum of squared error between the convolution's real and the required outputs. The minimization process takes in the form of [11]:

$$H = {}^{min}{}_{H} \sum_i |F_i \odot H * - G_i|^2 \quad (3)$$

All the elements in H can be optimized independently since the correlation in the Fourier domain is element-wise multiplication, [11]. The stable point is detected to obtain a single optimum. It is determined by setting the derivative of the real valued, positive and convex function equals to 0 and solving the function for the variable of interest in terms of H_{WV}^{*} and $H_{WV}^{\square}$. [11].

$$0 = \frac{\partial}{\partial H_{WV}^{*}} \sum_i |F_{iwv} H_{WV}^{*} - G_{iwv}|^2 \tag{4}$$

Equation (4) leads to:

$$0 = \frac{\partial}{\partial H_{WV}^{*}} \sum_i (F_{iwv} H_{WV}^{*} - G_{iwv})(F_{iwv} H_{WV}^{*} - G_{iwv})^{*} \tag{5}$$

$$0 = \frac{\partial}{\partial H_{WV}^{*}} \sum_i [(F_{iwv} H_{WV}^{*} - G_{iwv})(F_{iwv} H_{WV}^{*} - G_{iwv})]^{*} - (F_{iwv} H_{WV}^{*}) G_{iwv}^{*} - G_{iwv}^{\square} (F_{iwv} H_{WV}^{*})^{*} + G_{iwv}^{\square} G_{iwv}^{*} \tag{6}$$

$$0 = \frac{\partial}{\partial H_{WV}^{*}} \sum_i F_{iwv} F_{iwv}^{*} H_{wv}^{\square} H_{wv}^{*} - F_{iwv} G_{iwv}^{*} H_{wv}^{*\,\square} - F_{iwv}^{*} G_{iwv}^{\square} H_{wv}^{\square} + G_{iwv}^{\square} G_{iwv}^{*} \tag{7}$$

The partial derivative can be solved to be:

$$0 = \sum_i [F_{iwv} F_{iwv}^{*} H_{wv}^{\square} H_{wv}^{*} - F_{iwv} G_{iwv}^{*}]^{\square} \tag{8}$$

Solve $H_{wv}^{\square}$ [11]:

$$H_{wv}^{\square} = \frac{\sum_i F_{iwv} G_{iwv}^{*\,\square}}{\sum_i F_{iwv} F_{iwv}^{*\,\square}} \tag{9}$$

The expression in (9) can be rewritten as [11]:

$$H = \frac{\sum_i F_i \odot G_i^{*\,\square}}{\sum_i F_i \odot F_i^{*\,\square}} \tag{10}$$

Further explanation on the formulation can be found in [11].

3.0 RESULTS AND DISCUSSION

The comparison between the six object trackers has been done in terms of their frame per second (FPS). The object trackers are tested on both laptop and the Raspberry Pi (RPI). MP4 videos and live video stream are used as the medium for object tracking. The boundary box is used for the selected object. From the result obtained in Table 1, it can be concluded that MOSSE tracker produces a better FPS than all the object trackers for both Mp4 Video and Live stream, on both Laptop and RPI, except for Medianflow tracker, where it produces higher FPS for video live stream only. Even though Medianflow tracker produces a higher FPS for video live stream, it is unable to handle the high speed video frame and easily loses track. KCF Tracker can handle the high speed but the FPS produced is lower than the FPS expected and suitable for this work. In conclusion, MOSSE tracker the most ideal tracker to be applied on the UAV in this case. Figure 12 shows the result of the object tracking from the laptop. The MOSSE tracker operates successfully without any problems. The FPS is 43.49, which is enough for a live video.

Table 1. FPS result for each object tracking

Number	Object Tracker	Frame Per Second (FPS)			
		Mp4 Video		Video Live Stream	
		Laptop	RPI	Laptop	RPI
1.	KCF	22.48	6.10	11.66	0.17
2.	MOSSE	38.15	7.77	43.49	4.20
3.	MEDIANFLOW	37.63	6.29	55.26	7.36
4.	MIL	7.38	1.34	7.33	1.25
5.	TLD	8.00	1.73	11.48	1.76
6.	BOOSTING	12.86	2.23	9.04	1.17

Figure 12. Object tracked in a video live stream using laptop

Figure 13 shows the result of the object tracking while the UAV is taking off. The object tracking has been executed successfully in the beginning but because of network latency, the video live stream is not streamed in real-time. The video became stuck at certain frames and caused the window to be unable to update its next position in the consecutive frames. This happens when the remote desktop is used to display the video captured from the Raspberry Ri. The poor performance of the Raspberry Pi deters the operation of the UAV to follow the object being tracked while the UAV is taking off. It is suggested to replace the Raspberry Pi with a more powerful computing board such as Nvidia Jetson to improve the machine vision performance.

PERINTIS eJournal, 2024, Vol. 14, No. 1, pp. 25-37

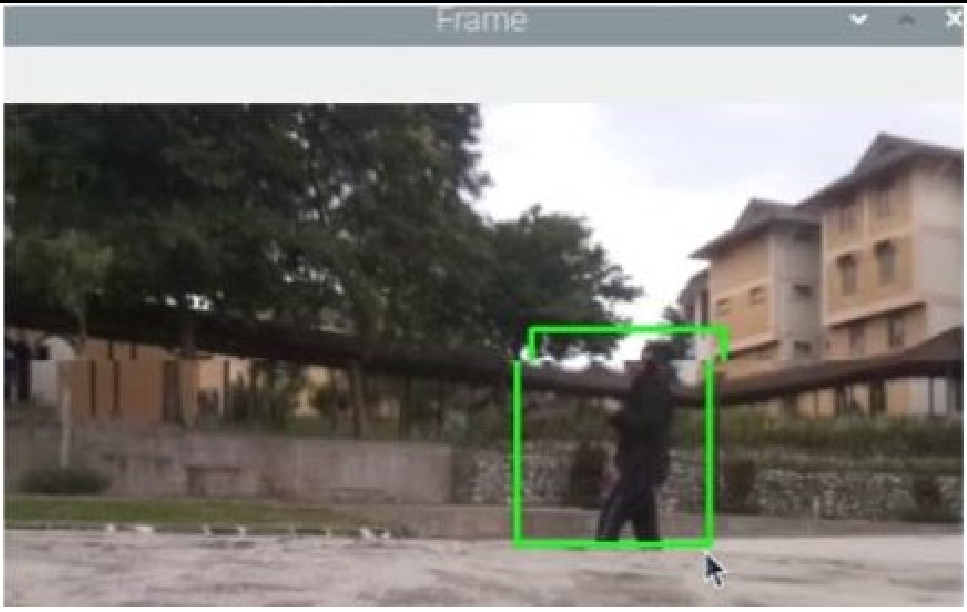

Figure 13. Object tracked in a video live stream using Raspberry Pi

4.0 CONCLUSION

Object tracking with Robot Operating System (ROS) has been implemented on the UAV rotorcraft or UAV in this study. Comparison has been done one object trackers available in the OpenCV library, which are the BOOSTING Tracker, Multiple Instant Learning (MIL) Tracker, Kernelized Correlation Filter (KCF) Tracker, Tracking, Learning and Detection (TLD) Tracker, Medianflow Tracker and Minimum Output Sum of Squared Error (MOSSE) Tracker. MOSSE gives the best performance compared to other techniques and has been applied on the UAV rotorcraft by programming it on the attached Raspberry Pi. The hardware experimental test result verifies that the proposed system is able to track the object while the UAV is taking off as desired. However, the system suffers from the limitation while UAV is taking off due to the Raspberry Pi and it is suggested to replace the device with a more powerful computing board such as Nvidia Jetson in future work. Nvidia Jetson provides a higher performance and more capable graphic processor. Future works also involves in testing the system in a real agricultural land to ensure that the system is practical for the actual situation.

REFERENCES

[1] A. G. Kendall, N. N. Salvapantula and K. A. Stol, "On-board object tracking control of a quadcopter with monocular vision," in 2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014 - Conference Proceedings, pp. 404-411,

2014.

[2] T. Yang, P. Li, H. Zhang, J. Li and Z. Li, "Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments," Electronics (Switzerland), vol. 7, no. 5, article no. 73, 2018.

[3] A. Mashood, A. Dirir, M. Hussein, H. Noura and F. Awwad, "Quadrotor object tracking using real- time motion sensing," in International Conference on Electronic Devices, Systems, and Applications, pp. 1-4, 2017.

[4] K. Boudjit and C. Larbes, "Detection and implementation autonomous target tracking with a quadrotor AR.Drone," in ICINCO 2015-12th International Conference on Informatics in Control, Automation and Robotics, Proceedings, pp. 223-230, 2015.

[5] F. Corrigan, "How do drones work and what is drone technology." 2020. Available: https://www.dronezon.com/learn-about-drones-quadcopters/what-is-drone-technology-or-how-does-drone-technology-work/. Last Accessed: 25 April 2024.

[6] W. D. S. Morgan Quigley, Brian Gerkey, "Robots with ROS", isbn: 9781449323899, 2015.

[7] P. Fankhauser, D. Jud, M. Wermelinger and M. Hutter, "Programming for robotics - Introduction to ROS," Lecture Notes. ETH Zurich, Switzerland, 2017.

[8] A. Rosebrock, "OpenCV object tracking", 2018. Available: https://www.pyimagesearch.com/2018/07/30/opencv-object-tracking/. Last Accessed: 25 April 2024.

[9] A. S. Putera and F. Ramdani, "Software testing by standard software metrics method; Study case "Mission Planner" as UAV ground station software," Journal of Telecommunication, Electronic and Computer Engineering, vol. 10, no. 1-8, pp. 123-128, 2018.

[10] D. S. Bolme, J. R. Beveridge, B. A. Draper and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2544-2550, 2010.