Terbit *online* pada laman web jurnal : http://metal.ft.unand.ac.id

METAL: Jurnal Sistem Mekanik dan Termal

| ISSN (Print) 2598-1137 | ISSN (Online) 2597-4483 |

Research article

Autonomous Quadcopter Image Processing for Simulated Search and Rescue Flights

Budi Hartono, Muhammad Rizki Zuhri, Citra Asti Rosalia, Nofrijal Fauzan

Politeknik Negeri Bandung, Jl. Gegerkalong Hilir, Ds. Ciwaruga, Bandung 40012, Indonesia

ARTICLE INFORMATION

Article History:
Received : 21 August 2023
Revised : 12 September 2023
Accepted : 27 October 2023

KEYWORDS

autonomous quadcopter

image processing

SAR flight simulations

CORRESPONDENCE

E-mail: buhar@polban.ac.id

A B S T R A C T

In Search and Rescue (SAR) operations, a designated search area is explored with aircraft and helicopters. After target identification, the flying vehicle continues landing and rescue procedures. This research uses a quadcopter to replicate a SAR flight simulation. Autonomous quadcopter operation includes takeoff and navigation between waypoints determined by Mission Planner software. On the way to the second waypoint, a camera-based image processing system scans the ground surface. If the marker is detected by the image processing system, the Raspberry Pi program will instruct control commands to the Pixhawk flight controller to ensure the quadcopter lands directly on the recognized marker. In the case when the quadcopter reaches the second waypoint but the system fails to identify the marker, the Mission Planner commands the quadcopter to autonomously return to the starting point and land automatically at the take-off location. An interesting aspect of this research concerns the application of a low-cost image processing system to ensure the quadcopter flies at a constant flight altitude above the ground surface, so that the quadcopter can perform simulated SAR flight missions and accurately identify landmarks. Research parameters include marker diameter, flight altitude, and quadcopter speed. The results show successful marker detection at a flying altitude of up to 3 meters above the ground and reaching a top speed of 3 m/s at a flying altitude of 2 meters.

1. BACKGROUND

In Search and Rescue (SAR) operations, the exploration of designated search areas with aircraft and helicopters is a crucial endeavor, where the identification of targets often leads to subsequent landing and rescue procedures. In this study, a quadcopter is employed to emulate a SAR flight simulation, emphasizing the significance of employing unmanned aerial vehicles in these critical missions.

The integration of image processing systems within autonomous quadcopters has gained substantial traction in recent research endeavors. Noteworthy examples include the work of Raj et al. [1], who devised an autonomous quadcopter employing image processing techniques for navigation. Similarly, Rezende et al. [2] seamlessly integrated image data from onboard cameras with Inertial Measurement Unit (IMU) data to facilitate navigation in a quadcopter platform. Concurrently, Sharma and Rajesh [3] achieved obstacle detection and avoidance capabilities in their quadcopter by

https://doi.org/ 10.25077/metal.7.2.8-17.2023

leveraging the OpenCV library for image processing.

The realm of autonomous quadcopter landing has also been a focal point of dedicated research initiatives. Tran et al. [4] employed an Arduino Uno microcontroller for flight control, coupled with a Raspberry Pi 3 to facilitate automatic landing through image processing. Respal et al. [5] harnessed image processing capabilities on the Parrot AR.Drone 2.0 to track and land on moving objects. Xuan-Mung et al. [6] introduced the DJI-F450 quadcopter, outfitted with an infrared sensor and a distance sensor utilizing laser technology for precise automatic landings.

Vivek et al. [7] contributed to the field by conducting research on quadcopters tailored for monitoring and surveillance, employing the Arducopter v2.8 flight control, a 1000 KV brushless motor, Ublox Neo-7 GPS for positioning, and IMU sensors to ensure orientation during automatic flights utilizing the Mission Planner software. In the autonomous quadcopter research domain, Gururajan and Bai [8] and [9] utilized the 3DR Pixhawk flight controller, a 924 KV brushless motor, and executed intricate figure-8 flight paths with varying waypoints, speeds, and altitudes. Lakshmanan et al. [10] developed an autonomous quadcopter employing an Ardupilot 2.8 Mega microcontroller, a 1,400 KV motor, and equipped with a 3000 mAh Lithium-Polymer battery, enhancing its versatility for extended missions. Additionally, Pluckter and Scherer [11] demonstrated the application of automatic landing techniques using a fisheye camera lens in conjunction with an Autel Robotics X-Star Premium Quadcopter, achieving a remarkable landing precision of 40 cm based on predefined takeoff patterns.

In the context of this research, the autonomous quadcopter undergoes comprehensive flight testing, encompassing takeoff and navigation between waypoints meticulously determined through Mission Planner software. While en route to the second waypoint, a camera-based image processing system meticulously scans the ground surface. Should the system successfully detect a ground marker, the Raspberry Pi program promptly issues instructions to the flight control system, orchestrating a precision landing directly above the recognized marker. Conversely, in scenarios where the quadcopter reaches the second waypoint without marker detection, the Mission Planner system autonomously directs the quadcopter to return to its initial departure point, initiating an automatic landing procedure at the take-off location.

An intriguing facet of this research pertains to the integration of an image processing system, pivotal in ensuring the quadcopter maintains a consistent flight height above the ground. This capability is instrumental in facilitating simulated SAR flight missions, optimizing the quadcopter's capacity to identify and respond to markers with a high degree of accuracy.

2. RESEARCH METHODOLOGY

The research encompassed the creation of an economical autonomous quadcopter design, featuring a 3D-printed frame for enhanced cost-efficiency. The quadcopter's ability to maintain flight at specific altitudes was realized through the integration of a LiDAR (Light Detection and Ranging) sensor. Complementing this, an image processing system was meticulously assembled, pairing a Raspberry Pi with a budget-conscious camera module. The comprehensive assessment regimen encompassed a series of autonomous flight trials, spanning takeoffs, navigation between predetermined waypoints, and precise automatic landings. The research scope was further extended to encompass simulated Search and Rescue (SAR) missions, introducing a dynamic element to the evaluation. The primary parameters under scrutiny encompassed marker dimensions, flight altitudes, and quadcopter velocities, collectively informing the research’s multifaceted investigation into autonomous flight control.

3. FINDING AND ANALYSIS

3.1. 3D-Printed Frame

The quadcopter's frame was crafted through the process of 3D-printing, with the design phase being orchestrated using Autodesk Fusion 360 software. The efficacy of Fusion 360 hinged on its capacity to simulate load placement intricacies and adhere to design constraints. The pinnacle outcome of the generative design analysis, as depicted in Fig.1, materialized through the utilization of a 3D-printing apparatus. Noteworthy aspects of this design encompass an impeccably distributed stress profile within the brushless motor mount and a frame structure that exhibits minimal stress. PLA (Polylactic Acid) filament emerged as the chosen printing material. The tangible manifestation of this effort can be observed in Fig.2, showcasing the resulting 3D-printed quadcopter frame, which is intricately linked with the budget-conscious autonomous quadcopter employed throughout this research, depicted in Fig.3.

Figure 1. Autodesk Fusion 360 simulation result of quadcopter frame generative design study

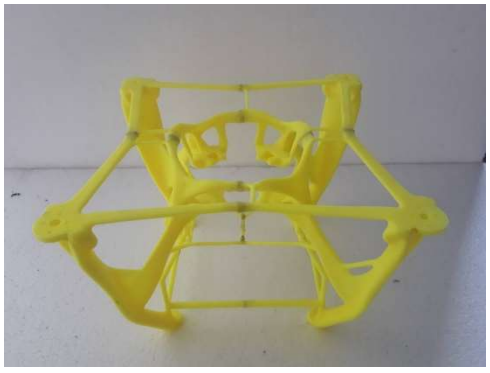

Figure 2. 3D-printed frame

Figure 3. Autonomous quadcopter with image processing system

3.2. Low-Budget Autonomous Quadcopter

The quadcopter utilized in this research is an autonomous configuration featuring a 1,300 gram take off mass, Neo M8N GPS unit, 920 KV brushless motor, and a 9x4.7-inch propeller.

The quadcopter's flight control system is orchestrated through the Pixhawk 2.4.8 platform. The image processing component seamlessly integrates a Raspberry Pi 3 B+ with an economical camera module boasting a 5 MegaPixel capacity and a frame resolution of 1280x720 pixels, as delineated in Figure 4. This camera module is strategically positioned to capture aerial imagery and transmits data to a ground station computer via a wireless interface facilitated by a WiFi Router. The images captured are conveniently stored on the Raspberry Pi's SD card.

Figure 4. Raspberry Pi and camera module connection

 https://doi.org/ 10.25077/metal.7.2.8-17.2023

The interconnections between the Pixhawk, Raspberry Pi, and laptop are elucidated in Figure 5. A local network, enabled by the WiFi router, facilitates communication between the Raspberry Pi and the laptop. Subsequently, access to the Raspberry Pi is established utilizing the PuTTY application; prior to this, the IP address of the Raspberry Pi must be ascertained, a process elucidated in Fig.6.

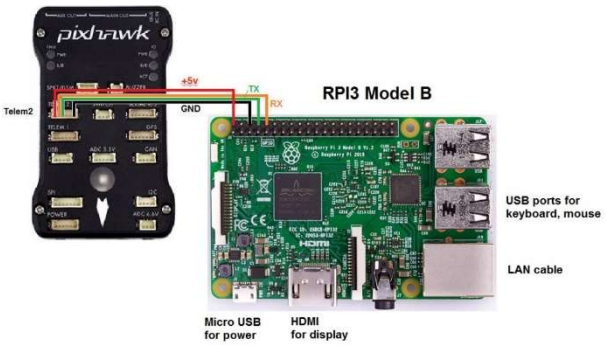

Figure 5. Pixhawk and Raspberry Pi connection (http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html)

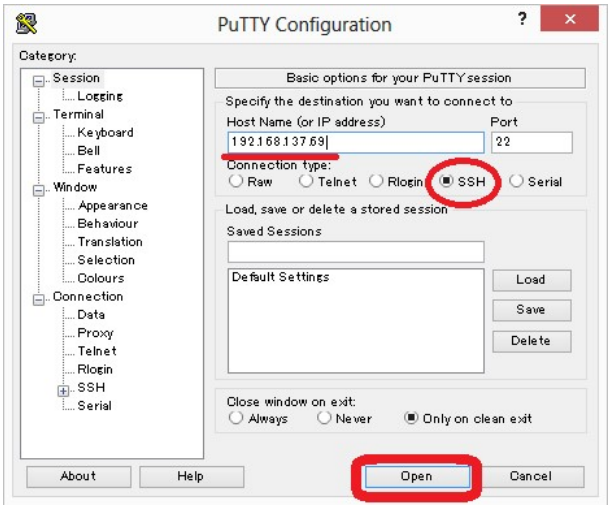

Figure 6. PuTTY configuration (http://ardupilot.org/dev/_images/RaspberryPi_Putty.png)

The camera module is meticulously situated in the central axis of the quadcopter frame, directed downward. Fig.7 provides a comprehensive overview of the spatial arrangement of the Pixhawk, Raspberry Pi, camera module, and additional components within the quadcopter.

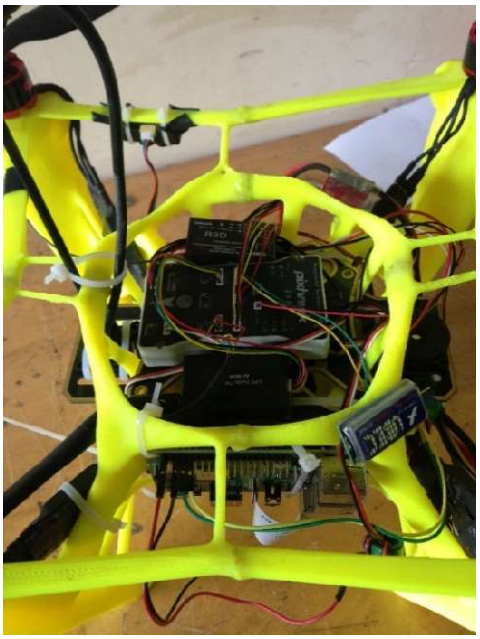

Figure 7. Quadcopter components placement

Functioning as a critical altitude sensor for quadcopter operations, LiDAR (Light Detection and Ranging) plays a pivotal role. The TF Mini LiDAR, showcased in Fig.8, is selected for its capacity to provide real-time, precise altitude data, and compatibility with the Pixhawk 2.4.8 platform. Notably, Pixhawk features a dedicated channel for seamless TF Mini LiDAR integration, obviating the need for intricate coding processes required by other flight controllers (APM).

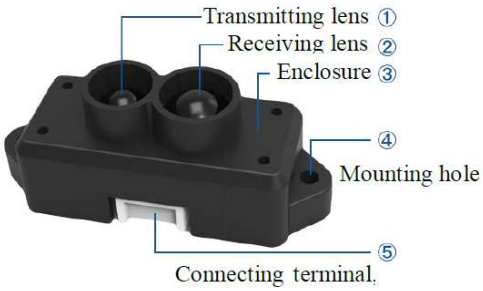

Figure 8. TF Mini LiDAR

The amalgamation of these essential components takes place within a 3D-printed quadcopter frame, an illustrative depiction of which is presented in the block diagram depicted in Fig.9.

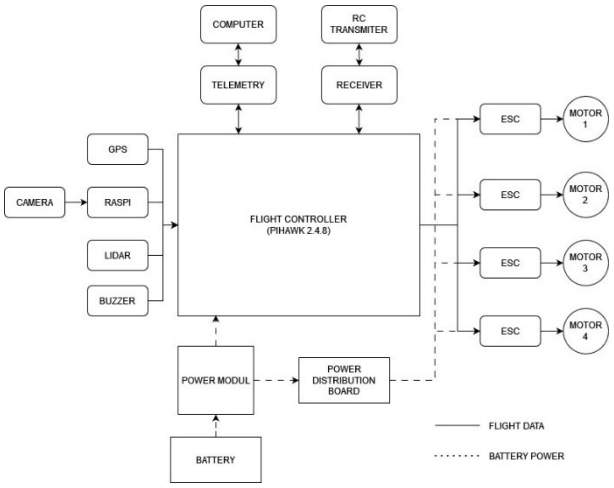

Figure 9. Block diagram of autonomous quadcopter system

Efficient information transmission from the Raspberry Pi to the Pixhawk is underpinned by programming languages such as Python and Dronekit. Python, being the default programming language for various Linux-based operating systems, including the Raspbian OS customized for Raspberry Pi devices, serves as a robust foundation for this purpose. Conversely, Dronekit is a purpose-built software library designed to streamline the command-giving process to flying crafts. This functionality empowers users to exercise control over the Pixhawk flight controller using the Python programming language, thus enhancing the versatility and operability of the autonomous quadcopter system.

3.3. Image Processing System

In this research, the Python programming language, coupled with the OpenCV library, is employed to facilitate real-time image processing, a fundamental component of the study. Within the OpenCV library, the Hue values span from 0 to 180, Saturation ranges from 0 to 255, and Value extends from 0 to 255, a structure detailed in Fig.10. To attain the requisite HSV (Hue, Saturation, Value) ranges for the specific task of scanning landing targets, precise adjustments are meticulously executed. The optimized HSV ranges, tailored for landing target detection, are meticulously defined as follows: Hue spanning 5 to 130, Saturation from 5 to 40, and Value encompassing 60 to 180. Fig.11 provides a visual representation of images captured by the camera, subjected to the predefined range of HSV values and subsequently converted into a binary black-and-white representation via thresholding.

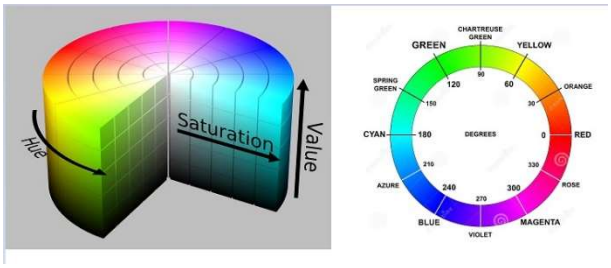

Figure 10. HSV color space in OpenCV library [12]

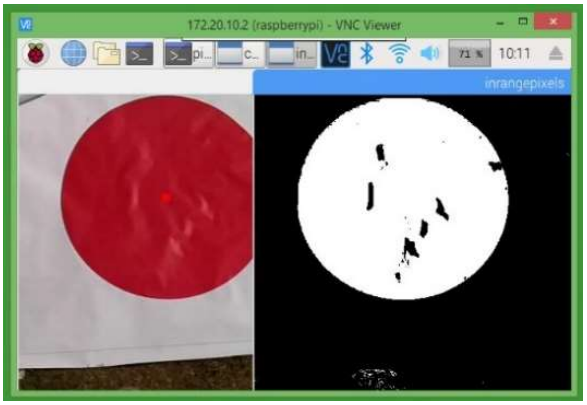

Figure 11. Image capturing from the camera and HSV threshold conversion

Subsequently, an in-depth analysis delves into the influence of HSV input values on color identification. Specifically, within the Hue range of 130 to 180, Saturation spanning 5 to 40, and Value ranging from 60 to 180, the camera's capacity to detect red color is rendered ineffectual, an evident outcome illustrated in Fig.12. In stark contrast, when adopting the value range encompassing Hue 5 to 130, Saturation spanning 40 to 255, and Value remaining within the interval of 60 to 180, the camera not only successfully identifies the red color but also exhibits the ability, as visually exemplified in Fig.13, to distinguish green grass. Opting for a value range that includes Hue values spanning 5 to 130, Saturation from 5 to 40, and Value ranging from 0 to 60 leads to the detection of the grass color, a discernible outcome showcased in Fig.14.

 https://doi.org/ 10.25077/metal.7.2.8-17.2023

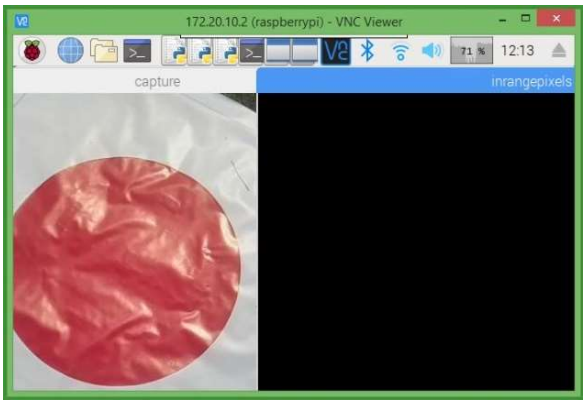

Figure 12. Result of Hue 130-180, Saturation 5-40, and Value 60-180 inputs

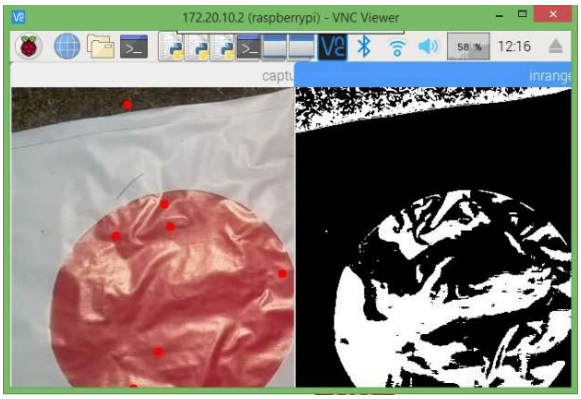

Figure 13. Result of Hue 5-130, Saturation 40-255, and Value 60-180 inputs

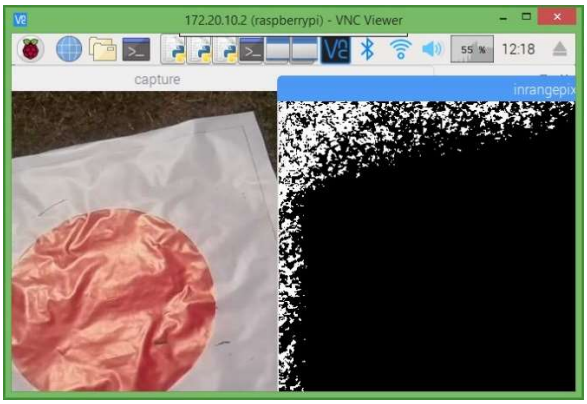

Figure 14. Result of Hue 5-130, Saturation 5-40, and Value 0-60 inputs

The image processing mechanism is strategically activated from the moment of the quadcopter's liftoff, diligently engaged in the pivotal task of detecting markers on the ground, discerned by their distinctive red circular configuration, representing designated landing targets. This process unfolds in accordance with the comprehensive flowchart presented in Fig.15. The primary action undertaken by the Raspberry Pi entails the transformation of color representation from the RGB color space to the HSV color space, subsequently specifying the precise ranges for Hue, Saturation, and Value parameters. Following this initial phase, the Raspberry Pi diligently proceeds to identify the largest contour, contingent upon a contour size threshold meticulously set at a minimum of 500 pixels. Should the discerned contour surpass this predefined threshold, it is unequivocally designated as a target. The pivotal juncture in this process occurs when both the most prominent contour and its central point are unmistakably identified as red. At this juncture, the Raspberry Pi issues a control directive to the Pixhawk flight controller, orchestrating a meticulously executed autonomous landing procedure, culminating in the precise alignment of the quadcopter with the identified marker on the ground.

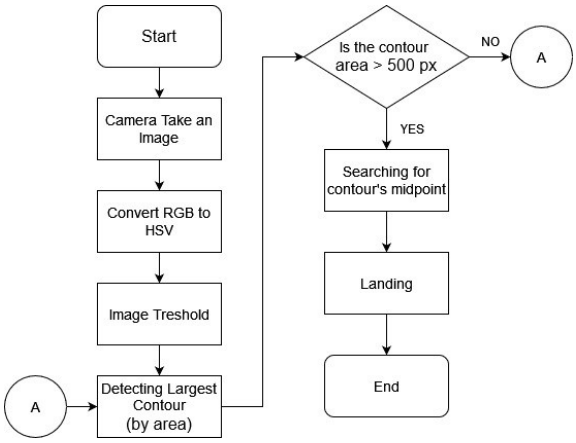

Figure 15. Image processing flow diagram

3.4. Autonomous Flight Test

A quadcopter, inherently unstable in its flight, necessitates the implementation of a robust flight control system to maintain stability. This control system continuously adjusts the rotational speeds of its four rotors. In the context of this research, the attainment of quadcopter stability was successfully achieved through the meticulous fine-tuning of the Proportional-Integral-Derivative (PID) flight control parameters, a process facilitated by the autotuning method.

Subsequently, a comprehensive evaluation of autonomous flight capabilities ensues, encompassing key flight phases such as autonomous takeoff, altitude maintenance (altitude hold), waypoint navigation from the initial to subsequent points, and the seamless execution of an automated landing. The incorporation of the LiDAR sensor within the quadcopter's architecture is a strategic move aimed at ensuring a consistent flight altitude.

To validate the reliability of the TF Mini LiDAR system, a flight test was meticulously conducted at a consistent flight altitude of 2 meters, precisely input into the Mission Planner software. During this evaluation, the Pixhawk system actively recorded flight altitude data based on the LiDAR sensor's measurements, while the flight controller's barometer sensor remained activated solely to monitor flight altitude data readings. Consequently, this approach allows for a meticulous comparison between flight altitude data derived from the LiDAR sensor and the barometer sensor. A total of 122 flight altitude data points were sampled and are visually depicted in Fig.16.

Analysis of the graph in Fig.16 reveals the quadcopter's consistent ability to maintain flight towards the target coordinates by effectively utilizing the LiDAR sensor data. In stark contrast, the flight altitude readings from the barometer sensor exhibited instability. Notably, the average flight altitude, as determined by the LiDAR sensor, equated to 1.78 meters, indicating a minor deviation of 0.22 meters from the predefined altitude set point.

Further analysis, as illustrated in Fig.17, demonstrates an average flight altitude of 5.22 meters with the LiDAR sensor, denoting a slight elevation of 0.22 meters from the designated set point. At a flight altitude of 7 meters, as depicted in Fig.18, the LiDAR sensor recorded an average flight altitude of 6.7 meters, reflecting a 0.30-meter reduction from the preset altitude. In contrast, the barometer sensor recorded an average altitude of 6.25 meters, signifying a 0.75-meter decrease. This comprehensive evaluation establishes that the altitude accuracy achieved by the LiDAR sensor during autonomous flight is particularly robust at a 7-meter altitude.

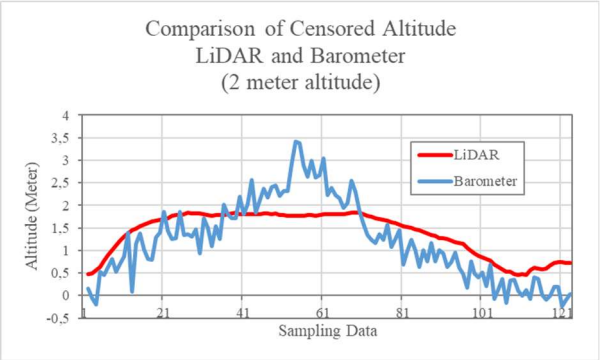

Figure 16. Comparison of altitude censored by LiDAR and barometer at 2-meter flight altitude

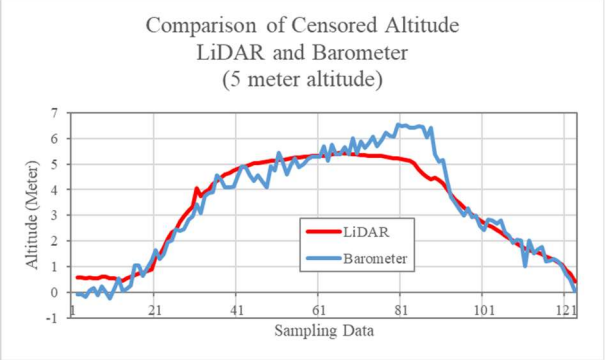

Figure 17. Comparison of altitude censored by LiDAR and barometer at 5-meter flight altitude

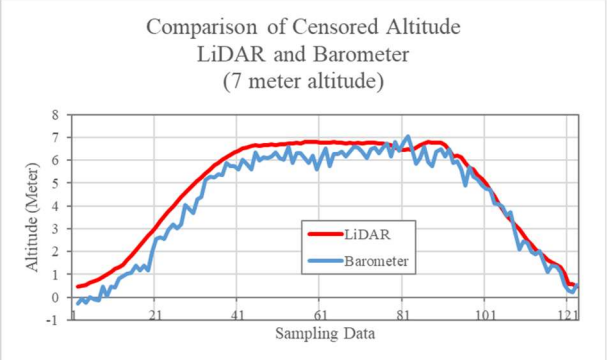

Figure 18. Comparison of altitude censored by LiDAR and barometer at 7-meter flight altitude

In conclusion, it is evident that the LiDAR sensor consistently delivers a higher level of altitude accuracy compared to the barometer sensor, particularly when deployed for autonomous flight operations at a 2-meter altitude from the ground.

 https://doi.org/ 10.25077/metal.7.2.8-17.2023

To further scrutinize autonomous flight capabilities, the quadcopter meticulously executed an autonomous flight test encompassing a flight path defined by five waypoints. These precise waypoint coordinates were meticulously input into the Mission Planner software, as visually indicated in Fig.19. Furthermore, the designed and constructed quadcopter was rigorously assessed for its Return to Base (RTB) flight functionality, a critical feature that enables the quadcopter to autonomously return to its initial take-off location.

Figure 19. Mission Planner display when setting waypoints for autonomous flight tests

3.5. SAR Simulation Flight Test

The search and rescue flight simulation, as illustrated in Fig.20, follows a structured protocol. The autonomous quadcopter commences its flight from takeoff, navigates between waypoints from the initial to the second navigation point, with the active image processing system continuously monitoring the ground surface. In the event that the system successfully detects the red marker on the ground, a command from the Raspberry Pi initiates a reduction in motor speed, facilitating the quadcopter's automatic landing on the designated marker. Conversely, if the quadcopter reaches the second navigation point without marker detection, the Pixhawk system orchestrates a return to base, culminating in an automated landing.

Autonomous flight experiments were rigorously conducted at altitudes of 2, 3, and 4 meters above ground level, as visually represented in Fig.21, with each scenario replicated thrice. Notably, landing targets were exclusively identified at altitudes of 2 and 3 meters, as the red circle marker remained undetectable at a 4-meter elevation. Under these circumstances, the quadcopter proceeded to the second waypoint and subsequently returned to the initial waypoint to execute an automated landing.

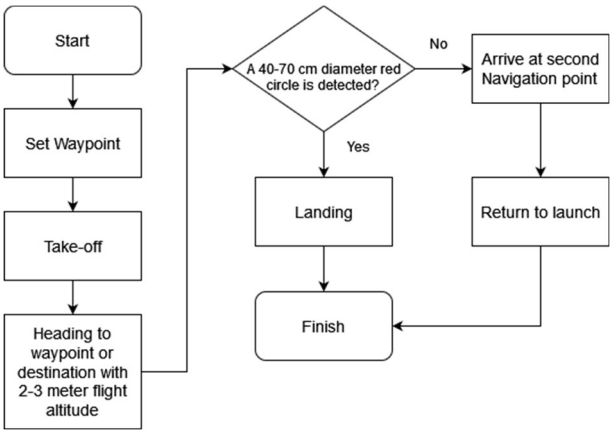

Figure 20. SAR flight simulation flowchart

Figure 21. SAR simulation flight test

Subsequent flight assessments entailed maintaining fixed flight altitudes of 2 and 3 meters while adhering to a constant flight speed of 1 m/s. Variations in the diameter of the red circle marker, ranging from 40 to 70 cm, were introduced, as outlined in Table 1.

Fig.22 visually portrays the Mission Planner interface during a SAR simulation flight test, alongside the measurement of the distance between the center of the landing target circle and the quadcopter's center point. Despite comprehensive analysis of the flight test data, no discernible correlation materialized between flight altitude, marker diameter, and landing point accuracy.

Tabel 1. Test flight at a flight speed of 1 m/s with variations in flight height and marker diameter

Flight height (meter)	Marker diameter (centimeter)	Distance from marker center to quadcopter center (centimeter)
2	40	33
		46
		35
	50	47
		97
		68
	60	20
		32
		140
	70	90
		114
		110
3	40	23
		95
		10
	50	20
		62
		49
	60	84
		120
		110
	70	145
		190
		190

Figure 22. Interface display during flight test simulation

The third flight test series involved maintaining flight altitudes of 2 and 3 meters, akin to the second stage, while systematically varying flight speeds from 1 m/s to 5 m/s. Throughout these trials, a consistent target landing diameter of 70 cm was sustained. This comprehensive examination confirmed that, at an altitude of 2 meters, the marker remains undetectable at flight speeds of 4 and 5 m/s. Similarly, at a 3-meter altitude, the landing target is solely identifiable at a flight speed of 1 m/s. The quadcopter seamlessly executes an automated landing on the designated marker in these instances. Conversely, in scenarios where the image processing system fails to detect the marker, the quadcopter proceeds to the second waypoint before initiating the Return to Base (RTB) maneuver, ultimately culminating in an automated landing at the take-off location.

Conclusively, the results derived from the flight tests spotlighted challenges associated with the low-cost camera specifications employed, notably manifested in considerable delays and latencies during the scanning process. Remarkably, the image processing system exhibited notable limitations in detecting the red circle marker when flight altitudes exceeded 4 meters. A similar limitation arose when flight altitudes remained at 3 meters and flight speeds exceeded 2 m/s.

4. CONCLUSION

The autonomous flight capabilities of the quadcopter, orchestrated through the Mission Planner software, encompass a comprehensive set of maneuvers. These include autonomous takeoff, the maintenance of a pre-defined altitude, and precise navigation between predetermined waypoints. Should the image processing system aboard the quadcopter fail to detect ground markers during its journey from the first waypoint to the second, an automated Return to Base (RTB) procedure is initiated, culminating in an autonomous landing back at the initial take-off position. Conversely, upon successfully identifying a marker, the quadcopter will seamlessly execute an automatic landing on the designated marker, thereby serving as the intended landing target.

The camera module at the core of this research boasts a 5 Mega Pixel specification, accompanied by a frame resolution of 1280x720 pixels. Despite these modest camera specifications, it's imperative to acknowledge inherent limitations within the

 https://doi.org/ 10.25077/metal.7.2.8-17.2023

image processing system's functionality. Specifically, this autonomous quadcopter exhibits the capability to discern landing targets exclusively when operating at altitudes no greater than 3 meters above ground level. In terms of speed, the quadcopter attains a maximum velocity of 3 m/s when positioned at an altitude of 2 meters, while its maximum speed drops to 1 m/s when navigating at a height of 3 meters. Furthermore, the system's minimal marker diameter detection threshold stands at 40 cm.

ACKNOWLEDGEMENT

The Authors would like to thank Politeknik Negeri Bandung that provided support and funding through P3M to this research article.

REFERENCES

[1] S. Raj, M. Dreyer, and S. Gururajan, "Autonomous quadcopter navigation using vision-based landmark recognition," in *2018 Aviation Technology, Integration, and Operations Conference*, 2018. doi: 10.2514/6.2018-4243.

[2] A. M. C. Rezende, V. R. F. Miranda, H. N. Machado, A. C. B. Chiella, V. M. Goncalves, and G. M. Freitas, "Autonomous system for a racing quadcopter," in *2019 19th International Conference on Advanced Robotics, ICAR 2019*, 2019. doi: 10.1109/ICAR46387.2019.8981660.

[3] V. N. V. A. Sharma and M. Rajesh, "Building a quadcopter: An approach for an Autonomous Quadcopter," in *2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018*, 2018. doi: 10.1109/ICACCI.2018.8554718.

[4] L. A. Tran, N. P. Le, T. D. Do, and M. H. Le, "A Vision-based Method for Autonomous Landing on a Target with a Quadcopter," in *Proceedings 2018 4th International Conference on Green Technology and Sustainable Development, GTSD 2018*, 2018. doi: 10.1109/GTSD.2018.8595521.

[5] V. M. Respall, S. Sellami, and I. Afanasyev, "Implementation of autonomous visual detection, tracking and landing for AR.Drone 2.0 quadcopter," in *Proceedings - International Conference on Developments in eSystems Engineering, DeSE*, 2019. doi: 10.1109/DeSE.2019.00093.

[6] N. Xuan-Mung, S. K. Hong, N. P. Nguyen, L. N. N. T. Ha, and T. L. Le, "Autonomous quadcopter precision landing onto a heaving platform: New method and experiment," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3022881.

[7] D. R. V. J, M. M, and M. J. R. M, "Autonomous Quadcopter for Surveillance and Monitoring," *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 7, no. 4, 2018.

[8] S. Gururajan and Y. Bai, "Autonomous 'figure-8' flights of a quadcopter: Experimental datasets," *Data*, vol. 4, no. 1, 2019, doi: 10.3390/data4010039.

[9] Y. Bai and S. Gururajan, "Evaluation of a baseline controller for autonomous 'figure-8' flights of a morphing geometry quadcopter: Flight performance," *Drones*, vol. 3, no. 3, 2019, doi: 10.3390/drones3030070.

[10] D. Lakshmanan, P. Saravanan, P. Vadivelu, D. Nivitha, and M. S. Yaswanth, "Performance Analysis of Medium Altitude Low-Cost Autonomous Quadcopter," in *IOP Conference Series: Materials Science and Engineering*, 2020. doi: 10.1088/1757-899X/764/1/012037.

[11] K. Pluckter and S. Scherer, "Precision UAV Landing in Unstructured Environments," in *Springer Proceedings in Advanced Robotics*, 2020. doi: 10.1007/978-3-030-33950-0_16.

[12] H. Singh, *Practical Machine Learning and Image Processing*. 2019. doi: 10.1007/978-1-4842-4149-3.

NOMENCLATURE

GPS	Global Positioning System
HSV	Hue Saturation Value
LiDAR	Light Detection and Ranging
PID	Proportional Integral Derivative
PLA	Polylactic Acid
RGB	Red Green Blue
RTB	Return to Base
SAR	Search and Rescue