International Journal of Marine Engineering Innovation and Research, Vol. 10(1), March. 2025. 40-46
(pISSN: 2541-5972, eISSN: 2548-1479)

Implementation of Waypoint Navigation and Computer Vision for Monitoring Markers on a Quadcopter Based on ROS (Robot Operating System)

Setyawan Ajie Sukarno[1] Hendy Rudiansyah[2], Ahsan Basyar [3]

(Received: 01 December 2024 / Revised: 23 December 2024 /Accepted: 08 January 2025 / Available Online: 21 March 2025)

***Abstract*— Indonesia shares borders with Papua New Guinea, Malaysia, and Timor Leste, where border markers often face displacement or disputes due to challenging and inaccessible terrain. This research develops a waypoint navigation system on a quadcopter, integrating computer vision to enhance the detection and monitoring of border markers. The system leverages the Robot Operating System (ROS) as middleware for seamless integration and control, while a camera detects ArUco markers placed on boundary markers. Image processing, implemented using OpenCV integrated with ROS, facilitates efficient data conversion. The quadcopter autonomously navigates to target coordinates based on marker detection, with an average percentage error of 3.3% for the X-axis and 2.5% for the Y-axis. Tests showed the system could detect a 40x40 cm marker from a height of 5 meters up to a distance of 14 meters, with an average position error of 3.75%. The communication range was effective up to 150 meters before timing out. Despite the computational limitations of the Raspberry Pi hardware, the system demonstrated efficiency, scalability, and ease of deployment. Future research will focus on hardware enhancements, the exploration of advanced image processing methodologies, improved camera resolutions, and the extension of communication networks to support deployment in national boundary monitoring operations.**

Keywords— *Border Marker, Surveillance, Quadcopter, Waypoint Navigation, Computer vision, ArUco Marker, ROS (Robot Operating System*)

I. INTRODUCTION

Boundary markers are fundamental components in defining a nation's sovereignty under international law [1]. The Law of the Republic of Indonesia Number 43 of 2008 explicitly delineates the country's territorial boundaries [2]. Indonesia shares direct borders with several neighbouring countries, thereby increasing the potential for territorial disputes, marker displacement, or even the loss of border markers [3]. Safeguarding sovereignty and national security necessitate prioritizing the monitoring and maintenance of these markers as part of broader national objectives [4]. However, challenges such as the inaccessibility of remote areas and limited human resources frequently hinder these efforts [5]. To address these challenges, quadcopter technology has emerged as a viable solution [6]. Quadcopters, a subset of Unmanned Aerial Vehicles (UAVs) or drones, have gained significant research interest over the past few years [7]. They are commonly employed in mapping, search-and-rescue missions, object detection, and particularly in surveillance operations [8].

Previous studies have primarily focused on GPS-based mission planning, drone stabilization for precise monitoring, and the use of cameras to capture critical details [9]. Quadcopters equipped with closed-loop control systems can perform tasks such as automatic hovering and directional rotation based on pre-set paths, enhancing monitoring precision [10]. Nugraha et al. focused on a quadcopter designed for object tracking using image processing, beginning with mechanical and electrical design, followed by the development of software utilizing C++ for image analysis. Data collection involved component-wise testing of the device [11]. Despite their utility, GPS does not always deliver accurate results under certain environmental conditions, leading to movement and landing inaccuracies [12]. As an alternative, computer vision technology has been explored for position tracking. For instance, a study by Tan et al. [13] utilized images from UAV cameras but faced challenges such as complex backgrounds, small targets, and occlusion, which resulted in detection errors. Similarly, ArUco markers have shown promise in detecting and tracking geographic boundaries [14]. By integrating computer vision with UAVs, M. Aly, in 2023, improved the accuracy of displacement estimation for bridges using ArUco marker.[15]. To mitigate the high costs of real-world drone testing [16], simulated environments such as Software in the Loop (SITL) are increasingly adopted, enabling comprehensive vehicle testing without the need for physical hardware [17][18].

Building upon prior research and advancements in quadcopters and the utilization of computer vision with ArUco markers, this study introduces a monitoring system capable of detecting and tracking national border

---

Setyawan Ajie Sukarno. Departement of Automation of Engineering, Politeknik Manufaktur Bandung, Bandung, 40135, Indonesia. E-mail: ajie@ae.polman-bandung.ac.id

Hendy Rudiansyah. Departement of Automation of Engineering, Politeknik Manufaktur Bandung, Bandung, 40135, Indonesia. E-mail: hendy_r@polman-bandung.ac.id

Ahsan Basyar. Departement of Automation of Engineering, Politeknik Manufaktur Bandung, Bandung, 40135, Indonesia. E-mail: basyarahsan@gmail.com

markers using a quadcopter. The integration of ROS (Robot Operating System) and image processing technologies within both SITL and real-world environments is expected to facilitate the surveillance of remote border areas that are otherwise difficult to access [19]. This research aspires to provide an innovative and cost-effective solution for safeguarding national border markers.

II. METHOD

This section presents a detailed description of the system architecture and workflow, accompanied by supporting diagrams.

A. System Design

The proposed system's primary objective is to control a quadcopter drone to navigate toward specified coordinate points, addressing precision and efficiency challenges [20]. Unlike conventional quadcopter control systems, this design incorporates advanced middleware integration and simulation capabilities to enhance adaptability and enable robust testing within virtual environments prior to real-world deployment. The system is designed to detect border markers using a camera integrated with the drone and employs ROS as middleware [21]. In this system, the workstation sends commands to the quadcopter drone through the communication layer, with PX4 as the drone flight controller. Software can replace this system In The Loop (SITL) to simulate drone flight without using hardware. SITL enables accurate and efficient flight simulation, allowing system testing and development in a virtual environment such as Gazebo [22]. It helps in virtually replicating drone behaviour, better-integrating hardware and software, and improving the understanding of drone flight dynamics and controllability [23].

Figure 1 illustrates the ROS-based quadcopter control system, which leverages various components for precise and efficient control. The user interacts with a workstation running ROS, enabling the development, testing, and control of the quadcopter through SITL simulation using Gazebo. Communication between the workstation and Raspberry Pi is established via a WiFi connection, where the Raspberry Pi functions as a companion computer that integrates data from the camera and controls the quadcopter through the PX4 Autopilot on the Pixhawk 2.4.8 FCU. The camera connected to the Raspberry Pi is used to detect ArUco Marker objects, which serve as markers. Additionally, the system includes the storage of relevant data in an SQLite Database, facilitating the storage and analysis of image processing results. This entire system ensures that the quadcopter can be effectively controlled and monitored, with the simulation in Gazebo assuring that all components are functioning correctly before real-world implementation [24].

Figure 2 highlights the integration of ROS and OpenCV within the system for image processing tasks such as detecting ArUco markers. CvBridge, a ROS library, facilitates seamless conversion between ROS image messages and OpenCV images (cv::Mat format). Images captured by the Raspberry Pi's camera are published as ROS image messages, converted to OpenCV objects for processing, and subsequently reconverted for further analysis and action [25][26]. This setup ensures efficient real-time image processing, which is critical for monitoring and navigation tasks.

B. System Flowchart

The system's flowchart, shown in Figure 3, begins with establishing a wireless network connection between the laptop/PC and the Raspberry Pi, ensuring they are on the same local network. Once connected, the Raspberry Pi communicates with the FCU Autopilot using the Mavros node. The waypoint navigation program is executed on the workstation, and the system completes its operation once the quadcopter reaches designated waypoints, captures and compares images, and lands at the home position.

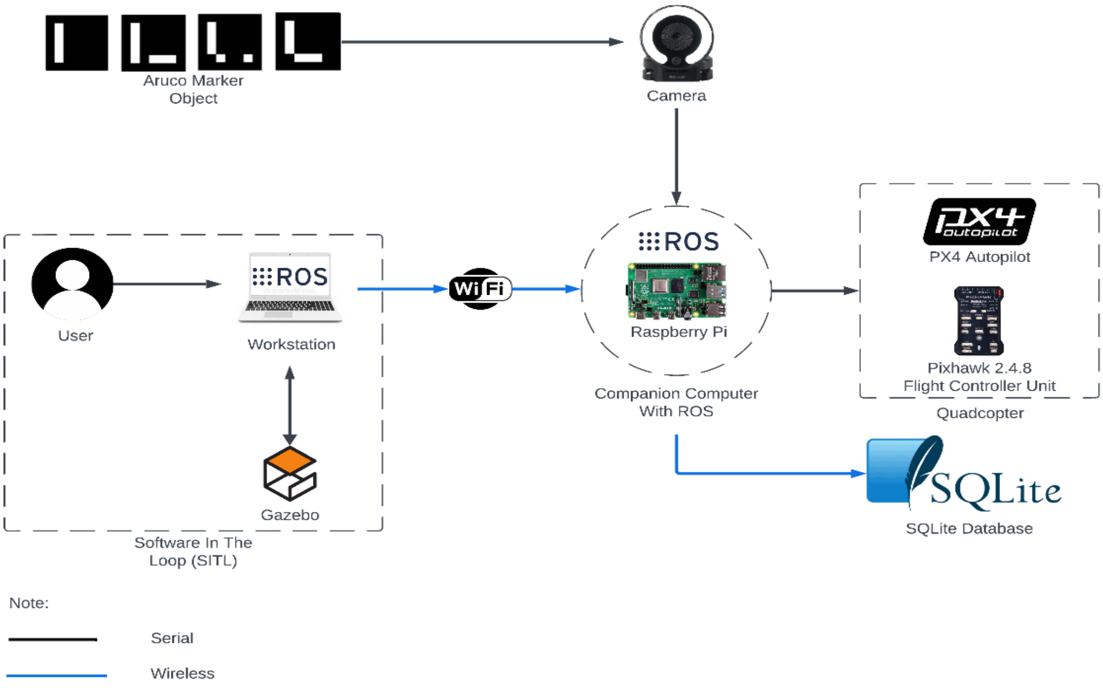


Figure 1. System Overview

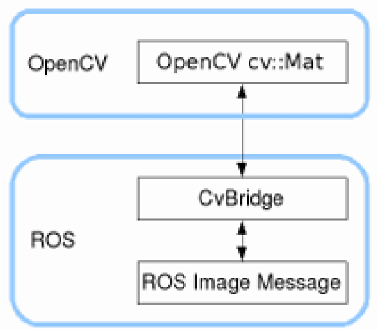


Figure 2. System Design OpenCV

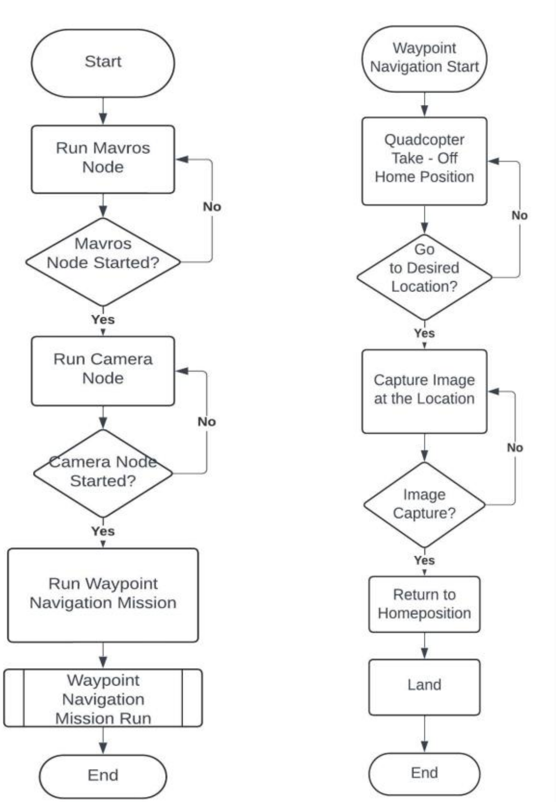


Figure 3. System Flowchart

III. Results and Discussion

The implementation and unit testing phases encompass communication, movement, and image processing evaluations. These evaluations provide valuable insights into the system's performance, revealing its capabilities and limitations in real-world scenarios. By analyzing these key findings, this section highlights the system's broader implications and explores opportunities for future enhancement.

A. Communication System Performance

The communication system range testing evaluated performance and stability under various conditions. Key components included a workstation as the data sender and a companion computer as the receiver, supported by a router/Wi-Fi network. Data transmission metrics, such as transmission success rate and response time, were recorded. Testing revealed an effective communication range of up to 150 meters. Expanding the communication network is essential for broader applications like national border monitoring. Table 1 and Figure 4 summarize these findings, underscoring the need to enhance range capabilities for optimal system performance.

While this range is suitable for controlled testing environments, broader applications like national border monitoring require enhanced range capabilities. The research highlights certain hardware limitations, particularly the computational capacity of the Raspberry Pi 4B. These limitations affected the system's real-time processing capability, especially during high-frequency data transmissions and image processing tasks. Furthermore, the maximum communication range of 150 meters, while sufficient for testing, may prove inadequate for larger-scale national border deployments. Addressing these limitations through exploring alternative communication protocols like LoRa or 5G can significantly enhance system scalability and performance.

B. Quadcopter Movement Accuracy

TABLE 1.
DATA COMMUNICATION RANGE TESTING

Distance (m)	Data Sent	Description	Response Time(ms)
0.5	PING (Network)	Success	3,43
0.5	PING (Network)	Success	3,87
0.5	PING (Network)	Success	2,34
0.5	PING (Network)	Success	2,34
0.5	PING (Network)	Success	2,57
25	PING (Network)	Success	2.38
25	PING (Network)	Success	3.86
25	PING (Network)	Success	2.13
25	PING (Network)	Success	4.57
50	PING (Network)	Success	4.09
50	PING (Network)	Success	3.87
50	PING (Network)	Success	2.34
50	PING (Network)	Success	10.81
100	PING (Network)	Success	5.88
100	PING (Network)	Success	32.51
100	PING (Network)	Success	10.18
100	PING (Network)	Success	3.38
100	PING (Network)	Success	5.44
150	PING (Network)	Success	110
150	PING (Network)	Success	284
150	PING (Network)	Success	117
150	PING (Network)	Success	98.20
150	PING (Network)	Success	60.92
200	PING (Network)	Failed	Request Timed Out
200	PING (Network)	Failed	Request Timed Out
200	PING (Network)	Failed	Request Timed Out
200	PING (Network)	Failed	Request Timed Out
200	PING (Network)	Failed	Request Timed Out

Movement tests verified the quadcopter's ability to navigate accurately and maintain stable altitudes. Using ROS and Gazebo, input coordinates were tested against the quadcopter's actual movements. The results, shown in Table 2, indicate an average positional error of 3.3% on the X-axis and 2.5% on the Y-axis, with reliable stability in altitude.

The objects involved include the quadcopter, ROS, and Gazebo as the simulator. The tools used include input values for coordinates on the X, Y, and Z axes, as

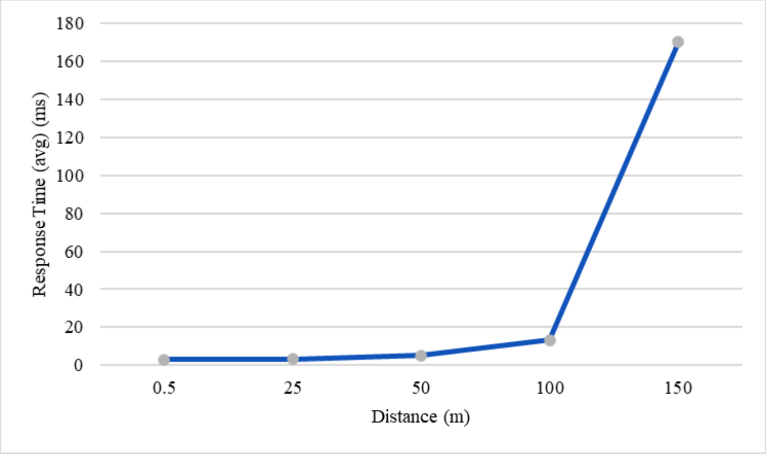


Figure 4. Graph Network Communication

TABLE 2.
ARUCO MARKER POSITION DISPLACEMENT TESTING

Trials	Target X (m)	Target Y (m)	Actual X (m)	Actual Y (m)	X Error (%)	Y Error (%)	Altitude (Z) Stability (m)
1	2	2	2,064	1,937	3,2%	3,2%	±0.1
2	2	2	1,977	1,897	1,2%	5,2%	±0.3
3	2	2	1,887	1,960	5,7%	2,0%	±0.5
4	2	2	2,040	1,999	3,0%	0,0%	±0.1
5	2	2	2,110	2,081	5,5%	4,1%	±0.2
6	2	2	2,072	2,055	3,6%	2,8%	±0.1
7	2	2	1,968	2,018	1,6%	0,9%	±0.5
8	2	2	1,947	1,960	2,7%	2,0%	±0.3
9	2	2	1,952	1,964	3,6%	1,8%	±0.2
10	2	2	2,042	2,068	2,6%	3,4%	±0.6
Percentage Error					3,3%	2,5%	±0.1

TABLE 3.
OBJECT DETECTION TESTING WITH DISTANCE VARIABLE

Trials	Distance Between Quadcopter to Object (m)	Detection of ArUco Object
1	2	Success
2	4	Success
3	8	Success
4	10	Success
5	12	Success
6	14	Success
7	16	Partial Success
8	18	Partial Success
9	20	Failed
10	20	Failed

well as the ROS topic "/mavros/local_position/pose" to monitor the quadcopter's body frame values and the parameter display in Gazebo to observe the movement measurements. The testing method involves inputting the coordinate values, moving the quadcopter according to those inputs, and monitoring the results by subscribing to the ROS topic and observing the parameters in the Gazebo. This testing verifies the quadcopter's ability to reach the desired coordinate points and maintain a stable altitude. While these errors are within acceptable thresholds for monitoring tasks, they highlight opportunities for optimization. Future implementations could incorporate adaptive control algorithms, such as machine learning-based systems, to minimize errors and further improve accuracy for demanding applications like boundary marker monitoring

TABLE 4.
ARUCO MARKER POSITION DISPLACEMENT TESTING

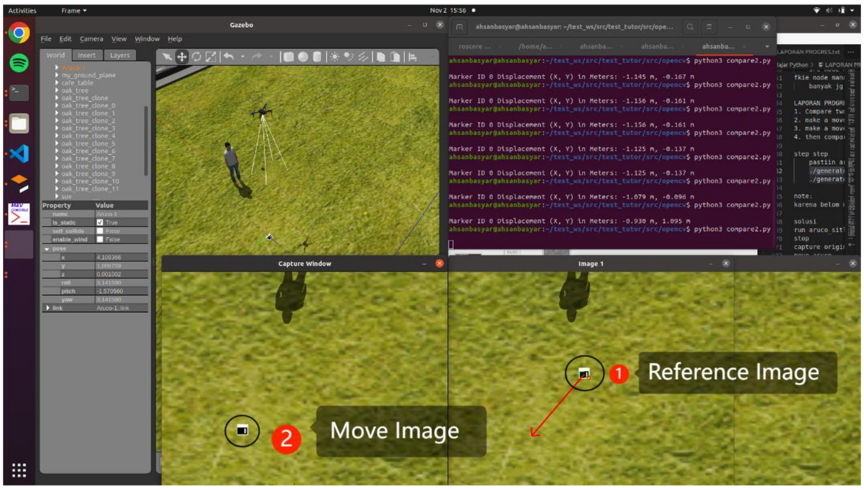
Trials	Distance Between Quadcopter to Object (m)	Referenced Image		Moved Image	
		X(m)	Y(m)	X(m)	Y(m)
1	4	1	0	0.926	0.096
2	4	1	0	1.071	0.047
3	4	1	0	1.119	0.095
4	4	1	0	1.131	0.021
5	4	1	0	0.926	0.049
Average Position (m)				1.035	0.042
Error Average Position (m)				0.035	0.042
Percentage Error				3%	4%

C. Result of Image Processing Results

The testing was conducted for object detection with the variable of different distances to the object. The object involved is the ArUco marker, which will be detected at various distances. The tools used in this testing include the object detection system and a distance measuring tool to ensure the accuracy of the distance between the detection system and the real-world object. As shown in Table 3, the 40x40 cm marker was clearly detected by the camera up to a distance of 14 meters. At a distance of 16–18 meters, detection was partially successful, while at a distance of 20–25 meters, detection failed. The second test involves measuring the accuracy of distance and object displacement using a camera compared to conventional measurements with tape measures. This testing is conducted at various distances with incremental shifts, focusing on measuring the distance and displacement of objects at predetermined positions. Figure 5 illustrates one such test, where the displacement of the ArUco marker is analyzed by comparing the "reference image" with the "move image”. The pixel displacement is then converted into meters using a reference image that contains an object, which is utilized to calculate the focal length. This focal length is a constant variable used for distance calculations. This testing is conducted at various distances with incremental shifts, focusing on measuring the distance and displacement of objects at predetermined positions. Figure 5 illustrates one such test, where the displacement of the ArUco marker is analyzed by comparing the "reference image" with the "move image”. The pixel displacement is then converted into meters using a reference image that contains an object, which is utilized to calculate the focal length. This focal length is a constant variable used for distance calculations.

TABLE 5.
ARUCO MARKER POSITION DISPLACEMENT TESTING

Trials	Distance Between Quadcopter to Object (m)	Referenced Image X(m)	Referenced Image Y(m)	Moved Image X(m)	Moved Image Y(m)
1	8	1	1	0.926	0.940
2	8	1	1	0.977	0.957
3	8	1	1	0.927	0.942
4	8	1	1	0.999	0.925
5	8	1	1	0.986	0.946
Average Position (m)				0.971	0.042
Error Average Position (m)				0.029	0.054
Percentage Error				3%	5%



```
ahsanbasyar@ahsanbasyar:~/test_ws/src/test_tutor/src/opencv$ python3 compare2.py

Marker ID 0 Displacement (X, Y) in Meters: -1.145 m, -0.167 m
ahsanbasyar@ahsanbasyar:~/test_ws/src/test_tutor/src/opencv$ python3 compare2.py

Marker ID 0 Displacement (X, Y) in Meters: -1.156 m, -0.161 m
ahsanbasyar@ahsanbasyar:~/test_ws/src/test_tutor/src/opencv$ python3 compare2.py

Marker ID 0 Displacement (X, Y) in Meters: -1.156 m, -0.161 m
ahsanbasyar@ahsanbasyar:~/test_ws/src/test_tutor/src/opencv$ python3 compare2.py

Marker ID 0 Displacement (X, Y) in Meters: -1.125 m, -0.137 m
ahsanbasyar@ahsanbasyar:~/test_ws/src/test_tutor/src/opencv$ python3 compare2.py

Marker ID 0 Displacement (X, Y) in Meters: -1.125 m, -0.137 m
ahsanbasyar@ahsanbasyar:~/test_ws/src/test_tutor/src/opencv$ python3 compare2.py

Marker ID 0 Displacement (X, Y) in Meters: -1.079 m, -0.096 m
ahsanbasyar@ahsanbasyar:~/test_ws/src/test_tutor/src/opencv$ python3 compare2.py

Marker ID 0 Displacement (X, Y) in Meters: -0.930 m, 1.095 m
ahsanbasyar@ahsanbasyar:~/test_ws/src/test_tutor/src/opencv$ python3 compare2.py
```

Figure 5. Result of System OpenCV

The marker will shift by 1 meter along the X-axis and remain at 0 meters on the Y-axis. The quadcopter will stay at the reference marker coordinate point so that the error calculation will focus solely on the marker's shift along the X and Y axes. The error for each trial is calculated, followed by the computation of the average error across all trials.

As shown in Tables 4 and 5, the system exhibited an average positional error of 3% on the X-axis and 4.5% on the Y-axis, demonstrating robust performance in detecting marker displacement. These findings validate the system's capability for precise monitoring while also identifying areas for further refinement.

IV. CONCLUSION

The research concludes that the waypoint navigation system on the quadcopter, integrated with computer vision technology, effectively detects ArUco markers as border markers and autonomously navigates to target coordinates, with an average percentage error of 3.3% for the X-axis and 2.5% for the Y-axis. The system demonstrated successful detection of a 40x40 cm marker from a height of 5 meters up to a distance of 14 meters. Marker displacement tests showed an average position error of 3.75%, confirming the system's effectiveness in identifying and monitoring border markers. While the system faced hardware limitations, including the computational capacity of the Raspberry Pi and a communication range capped at 150 meters, its use of the Robot Operating System (ROS) and OpenCV ensured seamless integration and real-time image processing. These strengths make the system efficient and suitable for development. Future research should focus on addressing hardware limitations by using a more powerful companion computer, such as one with higher specifications than the Raspberry Pi 4B. Exploring advanced image processing methodologies, adopting high-resolution cameras, and expanding the communication network are also recommended to enhance system performance and enable its deployment in national border monitoring applications. This research contributes to advancing quadcopter-based border surveillance systems with significant potential for real-world implementation.

REFERENCES

[1] J. Agnew, "Borders on the mind: re-framing border thinking," *Ethics Glob. Polit.*, vol. 1, no. 4, pp. 175–191, 2008, doi: 10.3402/egp.v1i4.1892.

[2] Mangku, D. G. S., Yuliartini, N. P. R., Mercury, S. M., & Darayani, N. M. C. (2022). The Role of the National Agency for Border Management in Maintaining the Territorial Sovereignty in the Land Bord between Indonesia and Timor Leste. *Proceeding of 1st Ahmad Dahlan International Conference on Law and Social Justice*, 120–131.

[3] S. Ruhana and T. A. Karim, "Indonesia vs. Malaysia: The Battle for Border Territory Resolved", *ILDiSEA*, vol. 3, no. 1, pp. 1-32, Jan. 2024.

[3] A. F. Moiz, H. M. Khawaja, R. Sohail, and Z. H. Khan, "QuadSWARM: A real-time autonomous surveillance system using multi-quadcopter UAVs," in *Proc. 2023 Int. Conf. Advanced Innovations in SmartCities (ICAISC-2023)*. IEEE, 2023.

[4] R. I. H. Abushahma, M. A. M. Ali, N. A. A. Rahman, and O. I. Al-Sanjary, "Comparative Features of Unmanned Aerial Vehicle (UAV) for Border Protection of Libya: A Review," in *2019 IEEE 15th Int. Colloquium on Signal Processing & Its Applications (CSPA)*, Penang, Malaysia, Mar. 2019, pp. 114–119. IEEE, doi: 10.1109/CSPA.2019.8695991.

[5] L. Natrayan, S. Kaliappan, and S. Pundir, "Control and monitoring of a quadcopter in border areas using embedded system," in Proc. Fourth Int. Conf. Smart Electronics and Communication (ICOSEC-2023), 2023, pp. 91–94. IEEE, doi: 10.1109/ICOSEC58147.2023.10276196.

[6] Y. Mekdad, A. Aris, L. Babun, A. El Fergougui, M. Conti, R. Lazzeretti, and A. S. Uluagac, "A survey on security and privacy issues of UAVs," *Computer Networks*, vol. 224, 109626, 2023. doi: 10.1016/j.comnet.2023.109626.

[7] P. Anggraeni, H. Khoirunnisa, M. N. Rizal, and M. F. Alfadhila, "Implementation of WiFi Communication on Multi UAV for Leader-Follower Trajectory based on ROS," in *2023 Int. Conf. Artificial Intelligence in Information and Communication (ICAIIC)*, Bali, Indonesia: IEEE, Feb. 2023, pp. 697–702. doi: 10.1109/ICAIIC57133.2023.10067024.

[8] R. K. Megalingam, D. V. Prithvi, N. C. S. Kumar, and V. Egumadiri, "Drone stability simulation using ROS and Gazebo," in

Lecture Notes in Networks and Systems, vol. 1, pp. 131–143, 2021. doi: 10.1007/978-981-16-2164-2_11.

[9] H. Khoirunnisa, F. S. Adi, A. Mulyadewi, S. A. Sukarno, Y. Erdani, and P. Anggraeni, "Implementation of IR Lock on Poledrone (Polman Drone Education) for Precision Landing with ROS," in *2023 IEEE 15th Int. Conf. Computational Intelligence and Communication Networks (CICN),* Dec. 2023, pp. 584–590. IEEE.

[10] I. Amiri, A. Shariffuddin, N. Kamel, M. Rahman, M. Bakar, M. B. Mhd Noor, S. Razalli, N. Buniyamin, and M. Khyasudeen, "The development of a GPS-based autonomous quadcopter for precision landing on a moving platform," *Int. J. Vehicle Autonomous Systems,* vol. 1, no. 1, pp. 1–18, 2021. doi: 10.1504/IJVAS.2021.10055418.

[11] M. Nugraha, A. Utomo, A. Taufik, R. Tandioga, and R. Syam, "Development of Quadcopter for Tracking Object Using Image Processing," in *IOP Conf. Series: Materials Science and Engineering,* vol. 619, pp. 012004, 2019. doi: 10.1088/1757-899X/619/1/012004.

[12] A. Priambodo, F. Arifin, A. Nasuha, M. Muslikhin, and A. Winursito, "A Vision and GPS Based System for Autonomous Precision Vertical Landing of UAV Quadcopter," *J. Phys.: Conf. Ser.,* vol. 2406, 012004, 2022. doi: 10.1088/1742-6596/2406/1/012004.

[13] L. Tan, X. Lv, X. Lian, and G. Wang, "YOLOv4_Drone: UAV image target detection based on an improved YOLOv4 algorithm," *Computers & Electrical Engineering*, vol. 93, 107261, 2021. doi: 10.1016/j.compeleceng.2021.107261.

[14] I. Lebedev, A. Erashov, and A. Shabanova, "Accurate Autonomous UAV Landing Using Vision-Based Detection of ArUco-Marker," in *Interactive Collaborative Robotics. ICR 2020. Lecture Notes in Computer Science,* vol. 12336, A. Ronzhin, G. Rigoll, and R. Meshcheryakov, Eds. Cham: Springer, 2020, pp. 279–291. doi: 10.1007/978-3-030-60337-3_18.

[15] M. Aly, "Leveraging Aruco Fiducial Marker System for Bridge Displacement Estimation Using Unmanned Aerial Vehicles," 2023.

[16] R. Perez-Segui, P. Arias-Perez, J. Melero-Deza, M. Fernández-Cortizas, D. Pérez-Saura, and P. Campoy, "Bridging the Gap between Simulation and Real Autonomous UAV Flights in Industrial Applications," *Aerospace*, vol. 10, no. 1, 2023.

[17] H. Qays, B. Jumaa, and A. Salman, "Design and Implementation of Autonomous Quadcopter using SITL Simulator," *Iraqi J. Comput., Commun., Control, and Syst. Eng.*, vol. 10, 2020. doi: 10.33103/uot.ijccce.20.1.1.

[18] M. H. Li and R. Dayansya, "Trajectory analysis of quadcopter UAV using software in the loop simulation," in *IET Int. Conf. Engineering Technologies and Applications (ICETA 2023),* Yunlin, Taiwan, 2023, pp. 200–201. doi: 10.1049/icp.2023.3339.

[19] CMa, Y. Zhou and Z. Li, "A New Simulation Environment Based on Airsim, ROS, and PX4 for Quadcopter Aircrafts," 2020 6th International Conference on Control, Automation and Robotics (ICCAR), Singapore, 2020, pp. 486-490, doi: 10.1109/ICCAR49639.2020.9108103.

[20] Lestari, D., Sujito, Sendari, S., Faiz, M. R., Wang, H. Y., & Prasanta, M. R. (2022). Quadcopter Design with Waypoint Mission Using PID Control System. *Proceedings - 11th Electrical Power, Electronics, Communications, Control, and Informatics Seminar, EECCIS 2022*, 287–291. https://doi.org/10.1109/EECCIS54468.2022.9902907

[21] C. Ma, Y. Zhou, and Z. Li, "A New Simulation Environment Based on Airsim, ROS, and PX4 for Quadcopter Aircrafts," in *2020 6th Int. Conf. Control, Automation and Robotics (ICCAR)*, 2020, pp. 486–490.

[22] N. Nair, K. Sareth, R. Bhavani, and A. Mohan, "Simulation and Stabilization of a Custom-Made Quadcopter in Gazebo Using ArduPilot and QGroundControl," in *Advances in Robotics and Automation,* vol. 1, pp. 205–217, 2022. doi: 10.1007/978-981-19-0836-1_15.

[23] R. Kumar and S. Jayashankar, "Radar and Camera Sensor Fusion with ROS for Autonomous Driving," 2019 Fifth International Conference on Image Information Processing (ICIIP), Shimla, India, 2019, pp. 568-573, doi: 10.1109/ICIIP47207.2019.8985782.

[24] K. Dang Nguyen and T. -T. Nguyen, "Vision-Based Software-in-the-Loop-Simulation for Unmanned Aerial Vehicles Using Gazebo and PX4 Open Source," *2019 International Conference on System Science and Engineering (ICSSE)*, Dong Hoi, Vietnam, 2019, pp. 429-432, doi: 10.1109/ICSSE.2019.8823322.

[25] S. Gatesichapakorn, J. Takamatsu and M. Ruchanurucks, "ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera," 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), Bangkok, Thailand, 2019, pp. 151-154, doi: 10.1109/ICA-SYMP.2019.8645984.

[26] A. Mulyanto, R. I. Borman, P. Prasetyawana, and A. Sumarudin, "2d Lidar and Camera Fusion for Object Detection and Object Distance Measurement of ADAS using Robotic Operating System (ROS)," *Int. J. Informatics Vis.*, vol. 4, no. 4, pp. 231–236, 2020, doi: 10.30630/joiv.4.4.466.