AUTONOMOUS TARGET TRACKING CONTROL METHOD FOR QUADROTORS USING ARTIFICIAL INTELLIGENCE

Thanh Han Trong*, Nghia Cao Xuan*, Hung Dinh Tan#

* School of Electronics and Telecommunications, Hanoi University of Science and Technology, Vietnam

School of Mechanical Engineering, Hanoi University of Science and Technology, Vietnam

Abstract: Tracking moving targets is an attractive application for quadcopters and is a complex area of research due to the aerodynamics of quadcopters and the rate of change of the moving target over time. In this paper, we build a quadcopter for target tracking by integrating a embedded computer Raspberry Pi (RPI) with a Pixhawk flight controller. This article also proposes a lightweight Tracking algorithm that can be deployed on Raspberry Pi, this algorithm harnesses advanced image processing and computing capabilities to significantly enhance target tracking performance, thereby reducing the need for human intervention control in unmanned flights. Controlling the quadcopter using this method helped the tracking system maintain stability in the simulated environment and achieve positive control parameters in real-world settings.

Keywords: Quadrotors, Raspberry Pi, Pixhawk, UAV.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) or drones are aircraft operated without a human pilot onboard [1]. There exists a system for UAVs called Unmanned Aircraft Systems (UAS) that enables communication with physical UAVs [2]. Typically, UAVs are controlled by humans using remote controllers known as Radio Controllers (RC) [3]. Additionally, they can be autonomously controlled by integrated systems onboard the UAV without the need for RC input. In this paper, a quadcopter [4] is utilized to perform autonomous flight combined with tracking algorithm to follow a selected target using Raspberry Pi 4 (RPI4) embedded computer [5] in conjunction with Pixhawk4 flight controller [5] for autonomous flight execution. Then, the paper proposes a target tracking algorithm and a control algorithm to let the Quadcopter automatically follow the target. RPI4 controls the Quadcopter by commanding the drone's controller (Pixhawk) to use the Drone-Kit API to send MAVLink (Micro Air Vehicle Link protocol) messages [6]. After successful connection, the target tracking algorithm is deployed on the RPI to control the Quadcopter to follow the target. Nowadays, tracking algorithms developed by OpenCV such as KCF [7], CSRT[8], MOSSE[9]... KCF is renowned for its high computational performance and good accuracy in tracking objects with small variations. MOSSE provides fast, efficient tracking and requires minimal computational resources. CSRT achieves high accuracy and stability in tracking objects with significant variability. However, these algorithms still have limitations such as ineffective tracking of fast-moving objects, object occlusion, or misidentification of duplicate objects. Subsequently, improved tracking methods using Deep Learning with SORT [10] algorithms have significantly enhanced accuracy and processing capabilities in complex scenarios. Other methods, such as the Siamese neural network [11], by leveraging the Siamese network to learn the relationship between objects in consecutive frames, coupled with the Region Proposal Network (RPN), positive results have been achieved. However, these algorithms require large training datasets, leading to lengthy training times, resource-intensive computations, and limited real-time processing capabilities. Consequently, deploying these algorithms on low-spec embedded devices with sluggish computing and processing speeds like Raspberry Pi 4 is not feasible. The paper also proposes a lightweight Tracking algorithm combining the AKAZE [12] feature extraction algorithm and the Kalman filter [13] to improve the KCF algorithm [14]. The algorithm not only retains its lightweight characteristics but also enables deployment on low-hardware devices such as the Raspberry Pi 4 with ease. Moreover, the algorithm offers enhanced target tracking capabilities compared to existing methods. By leveraging advanced techniques in image processing and machine learning, it achieves high accuracy and stability in tracking moving targets. Real-world experiments with actual UAVs are considered costly. Therefore, UAV systems need to be tested in simulation before real-world deployment. For simulation environment SITL, Gazebo [15] is chosen as it provides a powerful simulation environment for testing and developing robot control software in a safe and virtual environment before real-world deployment.

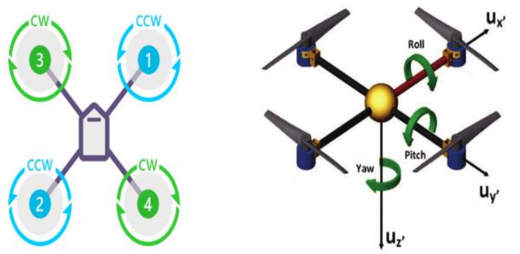
The article is organized as follows. Section 1 offers a general introduction. Section 2 outlines the methodology utilized in the paper. Section 3 presents the results achieved by the paper. Conclusion and future directions are discussed in Section 4.

Contact author: Thanh Han Trong
Email: thanh.hantrong@hust.edu.vn
Manuscript received: 01/2024, revised: 02/2024, accepted: 03/2024.

II. MATERIALS AND METHODS

A. Principles of Quadcopter Flight

A quadcopter [4] is a type of unmanned aerial vehicle (UAV) designed to be capable of flight based on the principle of four motors. The quadcopter consists of four motors connected to a frame at equal distances (typically in an X-shaped configuration). Two motors rotate clockwise, while the remaining 2 motors rotate counterclockwise. This distribution helps create a balanced system and control motion in the air to generate a specific thrust force to lift the quadcopter into the air [9], as illustrated in Figure 1.



(a) Propeller orientation (b) Motion direction

Figure 1. Propeller orientation, motion direction of the Quadcopter

The direction of the aircraft during movement is controlled by two motors, depending on the direction of movement, these motors adjust their speed to create an angle relative to the balance axis. The direction of movement of the aircraft is indicated by the change in speed of the motors.

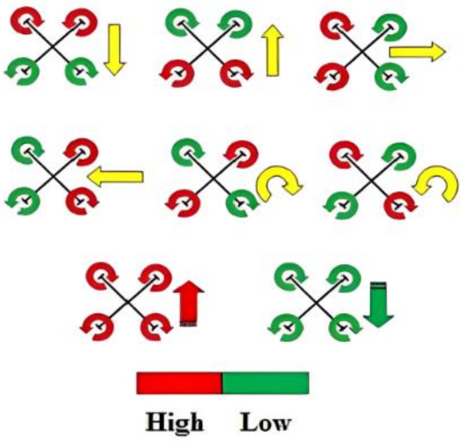


Figure 2. Quadcopter flight principle

To make the aircraft move forward, motors 1 and 2 will maintain or decrease speed while motors 3 and 4 will spin faster. The same applies to other directions. The process of controlling the direction of movement of the aircraft is usually done through an autopilot system or through pilot intervention. This system needs to be constantly considered and adjusted to maintain stability and safety during flight.

B. Quadcopter structure

In this paper, the structure of the unmanned aerial vehicle Quadcopter is displayed in Figure 3.

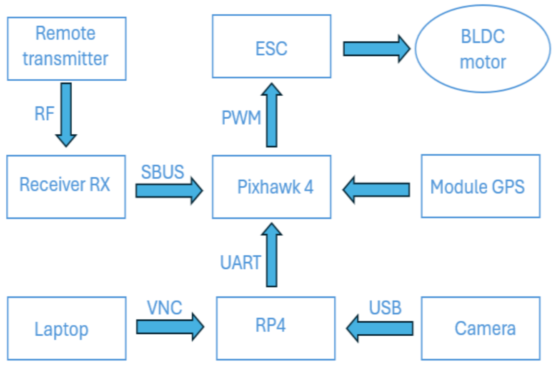


Figure 3. Block diagram of the Quadcopter model

The structure includes: Pixhawk 4 flight controller, Raspberry Pi 4 embedded computer, camera, Electronic Speed Controller (ESC), Signal receiver RX, (Brushless DC Motors (BLDC), transmitter remote, Quad frame, …The combination of Raspberry Pi 4 (RPI4) and Pixhawk 4 in the Quadcopter model is a unique integration that provides high efficiency in the drone field [5]. Raspberry Pi 4, a compact and powerful embedded computer, can run image processing algorithms and collect environmental data. Pixhawk 4, an autopilot flight controller, provides precise and stable control for the drone. Pixhawk 4 integrates sensors and decoders to ensure high accuracy in maintaining stability and navigation. This combination creates a perfect system, where RPI4 handles complex tasks such as image processing, data communication, while Pixhawk 4 is responsible for flight control and data collection from sensors.

Table 1 below presents the devices integrated into the prototype design of the study, along with the corresponding mass specifications of each device.

Table I. Weight of devices in Quadcopter

Device	Weight (gam)
Raspberry Pi4	63
Pixhawk 4	33
ESC 30A x4	168
Signal Receiver RX	18
GPS Module	33
Swivel base + Camera	45
Motor X2216 KV950 x4	280
Quadcopter Frame	480
Pine Lipo 3S 2300 mAh	180
Wire + other devices	60
Total weight	**1360 gam**

With the actual weight of the Quadcopter being 1360 grams, corresponding to a weight of 13.34 N (with the gravitational acceleration g being 9.81m/s2), various flight modes such as ascent (altitude changes), hovering and acceleration will require a greater lift force. The Quadcopter's four propellers are responsible for generating both lift and thrust, so the lift force needs to be doubled to ensure the performance of these tasks, totaling 26.68N (also the value required to maintain stability when losing control of flight). Based on the technical specifications and the provided propeller data, the thrust generated by the four propellers is 764 * 4 = 3056gf ≈ 29.98N, which is greater than 26.68N. Therefore, it can be concluded that the Quadcopter exceeds the required lift threshold to take off into the air. With the given weight, the Quadcopter is fully capable of withstanding wind, consuming less energy compared to larger UAVs, thereby maintaining tracking time and stability in rapidly changing environmental conditions.

C. Overview of target tracking system

The target tracking system based on Computer Vision comprises the UAV platform, the ground station platform, tracking algorithms, target state estimation algorithms and the UAV flight controller running on the Quadcopter platform in real-time. The ground station platform serves as the interface for monitoring the system's status and sending instructions to the Quadcopter.

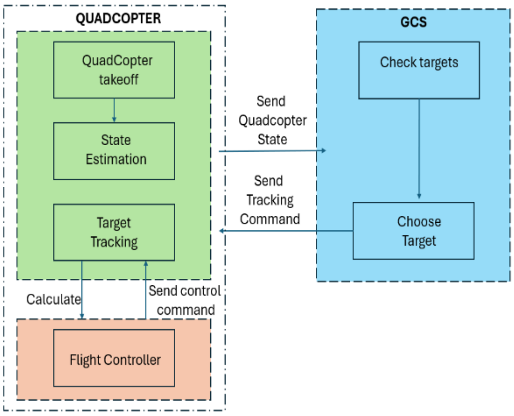


Figure 4. Operation diagram of target tracking function

After completing all pre-flight checks, the Quadcopter will take off and send the current status to the ground station. The ground station then enters target inspection mode and can select targets for tracking. Once the target is selected, the ground station sends a Track command to the Quadcopter. At this time, the position and velocity of the target are calculated by the flight controller to send control commands to the Quadcopter to track the target.

D. Algorithm used in the article

1) *Tracking algorithm*

Tracking is the process of estimating the motion of an object across successive frames, with the prerequisite that the object's position is known from previous frames. In this paper, the target is chosen based on user preference. After the user selects the target to track, the Tracking algorithm determines the position of the target in the frame (in pixel coordinates). Next, the state estimation algorithm is used to evaluate the local position and velocity of the target. Additionally, the flight controller is designed to control the speed and orientation of the Quadcopter accordingly based on the obtained information. This system enables effective target tracking and control of the Quadcopter's performance as illustrated in Figure 5.

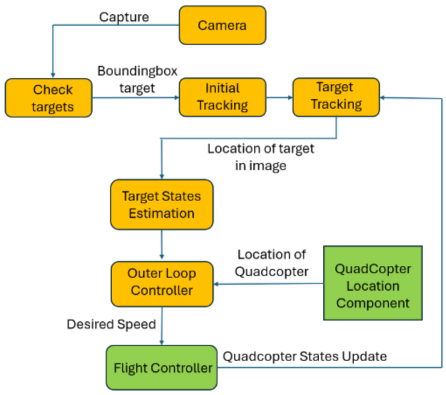


Figure 5. System operating process

Some lightweight tracking algorithms that can be used on RPI4 to track targets include KCF [7], CSRT [8], MOSSE [9]... However, in practical scenarios, objects may be occluded or their colors may change rapidly due to fast motion, causing the recognition algorithms to fail to detect them. Therefore, in this paper, we propose a new model combining the AKAZE [12] algorithm and Kalman filter [13] to enhance the KCF algorithm [14]. The AKAZE algorithm extracts feature points and the Kalman filter predicts the next position of the object.

a) AKAZE Algorithm

AKAZE [12] (Accelerated-KAZE) is an algorithm designed for rapid and efficient operation, particularly in scenarios involving intricate and noisy images. It is an accelerated version of the KAZE (KAZE Features) algorithm, designed to provide high accuracy and scale and position invariance features. With changes in scale and position of the object, the features can still be recognized.

The operating principle of the AKAZE algorithm involves constructing the Hessian matrix [16] and finding the extreme points of the Hessian determinant to identify the feature points in the image. By sliding successive windows and comparing them with the current scale window to find the pixels in each window and all its neighboring pixels, the maximum value is the extreme point. Once the position of the feature point (extreme point) is found, accurate pixel localization will be performed.

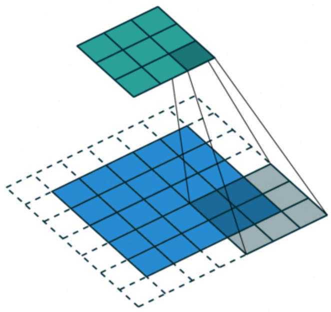


Figure 6. Sliding window process to find the extreme point

In this paper, Haar wavelet functions [17] are used based on the Wavelet transform, which helps analyze and represent image information at different detail levels from high-frequency details to low-frequency features, thereby reducing noise and increasing computational efficiency.

The Haar wavelet functions are defined by the following functions.

$$\phi(x) = \begin{cases} 1, & x \in [0,1) \\ 0, & else \end{cases} \quad (1)$$

$$\psi(x) = \begin{cases} 1, & x \in [0, \frac{1}{2}) \\ -1, & x \in [\frac{1}{2}, 1) \\ 0, & else \end{cases} \quad (2)$$

In which the function ϕ(x) is called the scaling function and the function ψ(x) is called the mother Wavelet function of the Haar wavelet functions. The family of basis functions is defined by:

$$\psi_k^j(x) = \begin{cases} 1, & x \in [\frac{k}{2^j}, \frac{k+0,5}{2^j}] \\ -1, & x \in [\frac{k+0.5}{2^j}, \frac{k+1}{2^j}] \\ 0, & else \end{cases} \quad (3)$$

With j being referred to as the dilation parameter and k being referred to as the translation parameter.

Assuming the scale parameter of the feature point is σ, the search radius is set to 6σ. Create a circular region with a radius of 6σ and then divide it into sections with 60° each, forming a fan-shaped area. Next, rotate the fan-shaped area and calculate the sum of the Haar wavelet features. The direction with the largest sum of wavelet features will be determined as the main direction ation parameter.

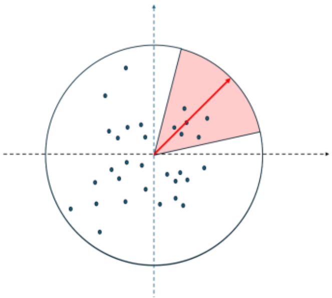


Figure 7. The sector scanning process to find the cardinal direction

b) Kalman Filter

Using the Kalman filter [13], we can predict the future state of the object by leveraging the iteration of two prediction and update states.

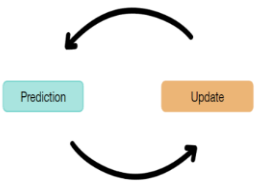


Figure 8. Loop between prediction and update state

- Prediction State:

$$\phi\hat{x}_{k|k-1} = F_k.\hat{x}_{k-1|k-1} \quad (4)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad (5)$$

With $\hat{x}_{k|k-1}, \hat{x}_{k-1|k-1}$ are the estimate of the system state at time k before receiving measurement data and time k-1 based on all previous measurement data, respectively. F_k is the motion matrix, representing the transition of the state from time k-1 to k. $P_{k|k-1}, P_{k-1|k-1}$ are the covariance at time k and k-1, respectively. Qk is the noise matrix.

- Update State:

In this step, the predicted state is corrected based on the difference between the actual measurement results z_k and expected measurement results $\hat{z}_k$ from the measurement matrix H. Equation (6) represents the mathematical expression of the measurement model.

$$\hat{z}_k = H.\hat{x}_k \quad (6)$$

After obtaining the actual measurement results, the difference $\tilde{y}_k$ between real measurement and model measurement is calculated according to equation (7). The complete calculation of this correction step is expressed from equations (8), (9).

$$\tilde{y}_k = z_k - \hat{z}_k \quad (7)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\tilde{y}_k \quad (8)$$

$$P_{k|k} = (I - K_k H_k)P_{k|k-1} \quad (9)$$

With $\hat{x}_{k|k}$ is an estimate of the system state at time k after receiving the measurement data, K_k is the Kalman weight, $P_{k|k}$covariance after receiving measurement data at time. I, H_k are the unit matrix and measurement matrix, respectively.

c) KCF tracker algorithm

KCF [14] (Kernelized Correlation Filter) tracker is a popular method in object tracking in video, based on kernelized correlation filters to predict the position of the object in the next frames of the video . In correlation filter, the correlation between two samples is taken and when these samples match then the correlation value is highest.

In KCF, the algorithm operates in the frequency domain using training samples captured by cyclic displacement of image patches. These samples are used to train a classifier in the frequency domain. The classifier's goal is to find the optimal function that minimizes the squared error between the samples and their regression targets. This function is represented as f(z) = $\omega^T z$, where ω is the optimal coefficient vector.

$$\min_{\omega} \sum_i (f(x_i) - y_i)^2 + \lambda\|\omega\|^2 \quad (10)$$

To solve for the classification coefficients α, the problem is transformed using the Fourier transform and the convolution theorem [18]. The coefficient α is obtained by solving the equation:

$$F(\alpha) = \frac{F(y)}{F(k^x) + \lambda} \quad (11)$$

where F represents the Fourier transform [19], y is the set of regression targets, and kx is the kernel distance calculated using a Gaussian kernel K in the Fourier domain.

For an image patch z of size W × H in a new frame t + 1, where z is cropped in the search window around the object location, the confidence response is calculated as follows:

$$\hat{y}(z) = F^{-1}(F(k^z) \odot F(\alpha)) \qquad (12)$$

Where $\odot$ is the element product, k^z =K(z, $\hat{x}_i$) is the kernel distance between the regression sample z and the learned object shape $\hat{x}_i$. The final target position is determined by the location where the maximum response R occurs, and the response R is calculated by R = max ŷ(z).

In this paper, AKAZE is used to extract features. The feature matching process is carried out to find the similarity between features from images based on their characteristics. Once suitable matching features are found with the previous features, the next step is to determine the corresponding feature pairs between two images. Then, the position of the moving object in the current frame is determined, and the bounding box around the object is computed. For cases where the number of matched points is smaller than the threshold, the Kalman filter is used to estimate the position and size of the object's bounding box. Finally, the bounding box coordinates are used to update the KCF tracker.

2) *Flight attitude estimation algorithm*

As previously described, the Tracking algorithm provides the coordinates of the object's centroid to estimate its state. These coordinates are represented as pixel coordinates. The Quadcopter will adjust its flight state so that these coordinates are shifted from the target coordinates to the coordinates of the center of the camera frame mounted on the Quadcopter.

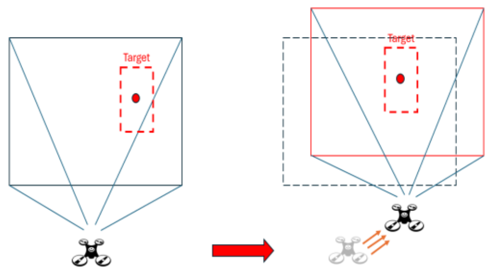


Figure 9. Quadcopter moves according to object coordinates

Assuming the target coordinates in pixels are (x, y) and the coordinates of the center of the frame are (w/2, h/2) where (w, h) is the resolution of the camera. Equations (13) and (14) respectively represent the pixel deviation between the target coordinates and the coordinates of the center of the frame along the x-axis and the y-axis.

$$\Delta x = x - \frac{w}{2} \qquad (13)$$

$$\Delta y = y - \frac{h}{2} \qquad (14)$$

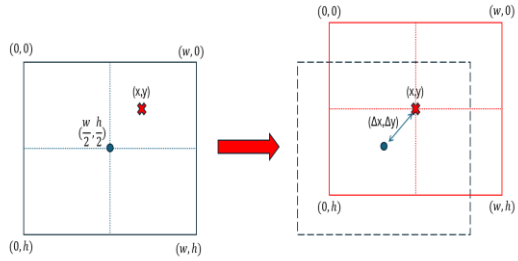


Figure 10. Converting object coordinates to the center of the frame

Once the change in coordinates (in pixel units) is known, we can describe the relationship between the change in target coordinates (pixels) and the change in Quadcopter coordinates (in meters) using the following expressions (15) and (16).

$$\Delta x_{quad} = k.\Delta x \qquad (15)$$

$$\Delta y_{quad} = k.\Delta y \qquad (16)$$

With k representing the scale factor that reflects the relationship between the measurement units on the image and the real-world units in physical space.

By initializing a safe distance d (from d1 to d2 meter) and a safe zone [x1:x2,y1:y2] (pixel), there will be shown in Table 12. Table 2 describes the conditions set as shown in the diagram above and the QuadCopter's movement under those conditions.

Table II. Movement conditions for Quadcoper

No	Condition	Control
1	d > d2	Fly forward
2	d < d1	Fly backward
3	d1 < d < d2	Keep state
4	x > x2	Move right
5	x < x1	Move left
6	x1 < x < x2	Keep state
7	y > y2	Fly down
8	y > y1	Fly up
9	y1 < y < y2	Keep state

III. EXPERIMENT

A. *Tracking Algorithm*

The paper compares the performance of the Tracking algorithm using AKAZE and Kalman to improve the KCF algorithm with lightweight tracking algorithms that do not require high hardware and are popular today such as KCF, MOSSE, and CSRT. We use two evaluation parameters to compare: FPS (Frame Per Second) and IoU (Intersection over Union). FPS is measured by dividing one second by the average processing time of one frame and IoU is evaluated by calculating the agreement between the bounding box predicted by the algorithm and the labeled ground truth bounding box with an agreement ratio in the range of 0.5 – 0.95. Table 3 presents the results of the tracking algorithms with the videos used in the paper. All videos demonstrate the process of tracking a single object, then that object is obscured by another object and its subsequent reappearance.

Table III. Comparison results between tracker algorithms and the algorithm proposed in the article

Tracker / Video	MOSSE		KCF		CSRT		AKAZE+Kalman	
	FPS	IoU 0.5-0.95	FPS	IoU 0.5-0.95	FPS	IoU 0.5-0.95	FPS	IoU 0.5-0.95
Video1	44	0.674	31	0.667	19	0.644	19	0.65
Video2	28	0.204	23	0.281	9	0.228	15	0.418
Video3	45	0.084	27	0.162	10	0.124	25	0.274
Video4	20	0.011	15	0.012	11	0.029	27	0.122
Video5	15	0.194	15	0.187	13	0.189	14	0.435

Analyzing Table 3, it becomes evident that the MOSSE algorithm boasts a high Frames Per Second (FPS) rate but exhibits lower accuracy. On the other hand, the CSRT algorithm shows a lower FPS rate but higher accuracy. The KCF algorithm demonstrates stable FPS and accuracy when compared to MOSSE and CSRT. However, it may lose track of the target when the object moves swiftly or becomes occluded by another object. In contrast, our proposed algorithm strikes a balance between FPS and accuracy, outperforming other tracking algorithms.

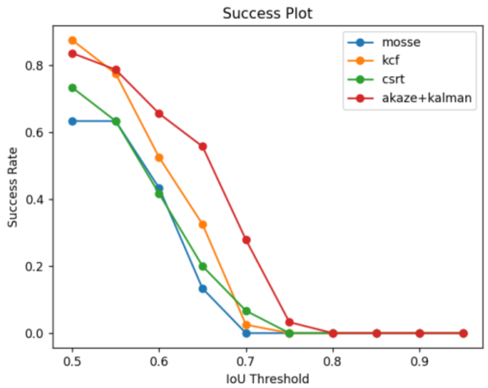


Figure 11. Success Rate with threshold from 0.5 - 0.95

From the Success Plot graph, it can be observed that the success rate of the proposed algorithm is higher than the KCF, MOSSE, and CSRT algorithms as the IoU threshold increases. This indicates strong tracking performance, with the algorithm accurately identifying the object even under various conditions and levels of overlap.

Furthermore, by combining the AKAZE algorithm with the Kalman filter to improve the KCF algorithm, we can solve the issue of occlusion when tracking targets. The AKAZE algorithm's feature extraction and the Kalman filter's predictive ability ensure accurate tracking even in challenging scenarios like fast-moving targets or occlusion, thus enhancing tracking system reliability.

Figure 12. Comparison results with today's popular tracking algorithms

In summary, besides the improvements in FPS and accuracy, our proposed algorithm has addressed the issue of target tracking when the target moves quickly or is occluded by other objects. Moreover, the algorithm is still designed with lightweight factors (similar to KCF), making it an ideal choice for deployment on Raspberry Pi (RPI4). This is particularly important when integrating the tracking system into resource-constrained applications with limited storage space.

B. Simulation results

In this section, we simulated and tested the human-following Quadcopter system based on Gazebo and evaluated the impact of target-following controllers in the simulated environment.

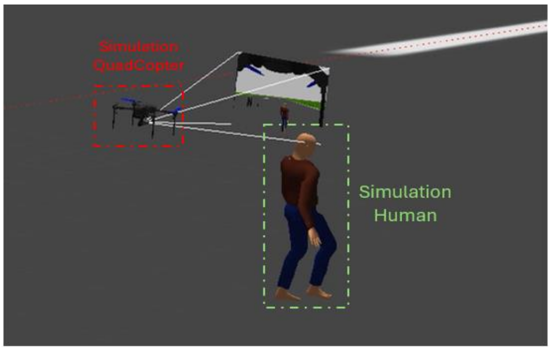


Figure 13. Simulation of Quadcopter following a human

In this paper, in the simulation environment, we will conduct two scenarios. The first scenario tests the Quadcopter's ability to follow a real person.

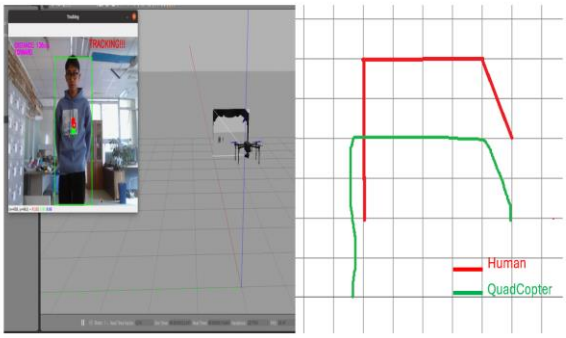


Figure 14. Results of Quadcopter following real people

The second scenario, similar to the first one, involves the Quadcopter tracking a simulated character.

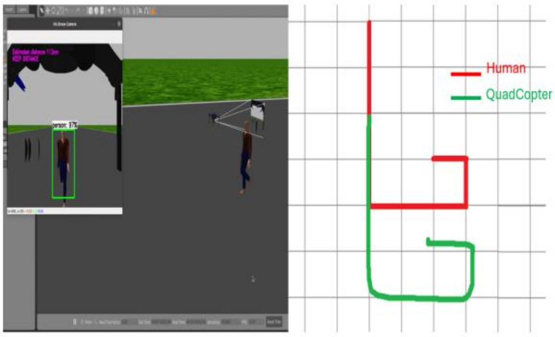


Figure 15. Quadcopter results following the simulated character

C. Reality results

In the real-world, we have successfully built a Quadcopter with all the functionalities outlined in Section II. Figure 17 illustrates the basic structure of the Quadcopter we built.

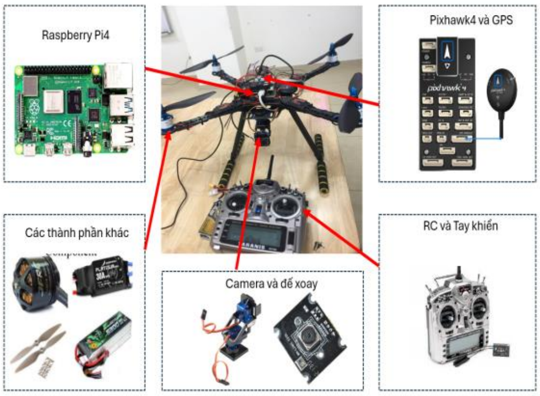


Figure 16. Assembled Quadcopter model

In the real-world test scenario, we enabled the Quadcopter to track and follow a target (the target being a person as shown in Figure 17) and examined the velocity control parameters returned.

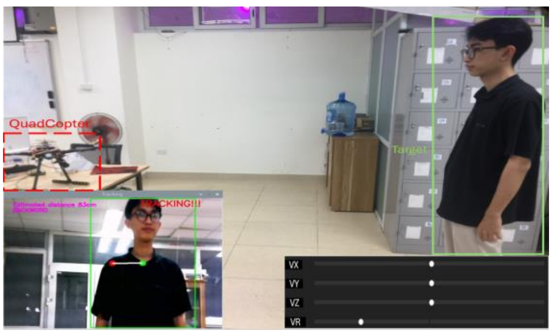
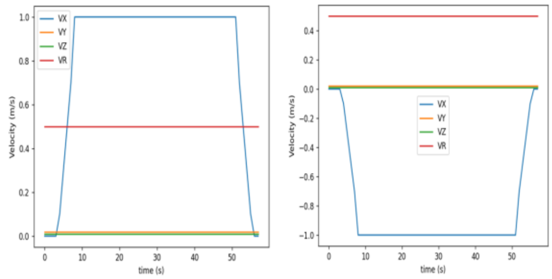


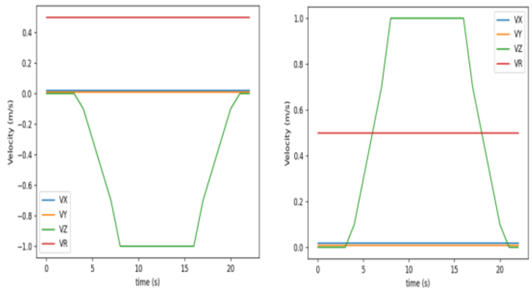
Figure 17. Figure 17. Actual experimental process

When the target moves forward or backward, the Quadcopter will move forward or backward at a initial velocity (in this paper we set velocity is 1 m/s) by controlling pitch_VX until the safe distance is reached, at which point the velocity will return to 0 m/s.



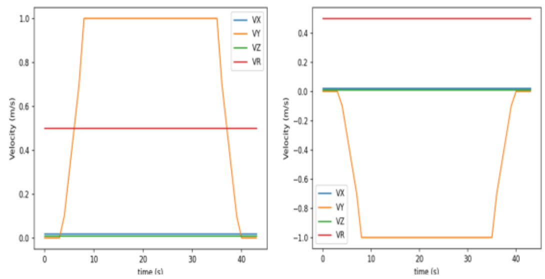
(a) Target moves away (b) Target moves closer

Figure 18. Quadcopter moves forward/backward



(a) Target higher (b) Target lower

Figure 19. Quadcopter flying up/down



(a) Target moves right (b) Target moves left

Figure 20. Quadcopter moves right/left

IV. CONCLUSION

In this paper, an automatic target tracking system based on Tracking algorithm is designed and deployed for the Quadcopter platform. By using the feature extraction algorithm AKAZE combined with the Kalman filter to improve the KCF algorithm, the system has improved FPS, accuracy and improve target tracking capabilities even when the target is occluded. Along with the controller tuning steps, our system can effectively track targets in a short period. Simulation results demonstrate that the developed tracking system achieves stable tracking performance and real-world tests also provide accurate control parameters. This system can be applied in various practical fields such as tracking, obstacle avoidance, 3D object detection, or multi-target tracking. In the future, a new algorithm will be researched, potentially improving automation further by utilizing machine learning algorithms with sensor output as inputs to provide appropriate control commands for specific tasks.

REFERENCES

[1] C. Deng, S. He, Y. Han and B. Zhao, "Learning Dynamic Spatial-Temporal Regularization for UAV Object Tracking," IEEE Signal Processing Letters, vol. 28, pp. 1230-1234, 2021.

[2] Samuel C.Hassler, Fulya Baysal-Gurel, "Unmanned Aircraft System (UAS) Technology and Applications in Agriculture," Department of Agricultural and Environmental Sciences, Otis L. Floyd Nursery Research Center, Tennessee State University, McMinnville, TN 37110, USA, 2019.

[3] T. Susanto, M. Bayu Setiawan, A. Jayadi, F. Rossi, A. Hamdhi and J. Persada Sembiring, "Application of Unmanned Aircraft PID Control System for Roll, Pitch and Yaw Stability on Fixed Wings," 2021 International Conference on Computer Science, Information Technology and Electrical Engineering (ICOMITEE), Banyuwangi,Indonesia, pp. 186-190, 2021.

[4] Nguyen Hung Thai Son, Vo Nguyen Phuc, "Design and construction of Quadcopter," Journal of Science of Lac Hong University, vol. 4, pp. 28-32, 2015.

[5] Pawel Smyczyński, Lukasz Starzec, Grzegorz Granosik, "Autonomous drone control system for object tracking: Flexible system design with implementation example," 2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), Miedzyzdroje, Poland, pp. 734-738, 2017..

[6] Hasan M. Qays, Dr. Bassim A. Jumaa, Dr. Ayman D. Salman, "Design and Implementation of Autonomous," vol. 20, no. 1, January 2020.

[7] Honghong Yang, Sheng Gao, Xiaojun Wu1, Yumei Zhang, “Online multi object tracking using KCF based single object tracker,” Springer-Verlag GmbH Germany, part of Springer Nature 2020, 2020.

[8] Dardagan, Nadja & Brdjanin, Adnan & Džigal, Džemil & Akagic, Amila. (2021). Multiple Object Trackers in OpenCV:ABenchmark. 10.1109/ISIE45552.2021.9576367.

[9] Haggu, Olfa & Tchalim, Agninoube & Baptiste, Magnier. (2021). A Comparison of OpenCV Algorithms for Human Tracking with a Moving PerspectiveCamera. 10.1109/EUVIP50544.2021.9483957.

[10] M. I. H. Azhar, F. H. K. Zaman, N. M. Tahir and H. Hashim, "People Tracking System Using DeepSORT," 2020 10th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia,2020,pp.137-141,doi: 10.1109/ICCSCE50387.2020.9204956

[11] K. Fu, D. -P. Fan, G. -P. Ji, Q. Zhao, J. Shen and C. Zhu, "Siamese Network for RGB-D Salient Object Detection and Beyond," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 9, pp. 5541-5559, 1 Sept. 2022, doi: 10.1109/TPAMI.2021.3073689.

[12] Surendra Sharma, Kamal Jain, "Image Stitching using AKAZE Features," Journal of the Indian Society of Remote Sensing, pp. 1-13, 2020

[13] Liana Ellen Taylor, Midriem Mirdanies, Roni Permana Saputra, "Optimized Object Tracking Technique Using Kalman Filter," J. Mechatron. Electr. Power Veh. Technol 07 (2016), pp. 57-66, 2016.

[14] Honghong Yang, Sheng Gao, Xiaojun Wu1, Yumei Zhang, “Online multi object tracking using KCF based single object tracker,” Springer-Verlag GmbH Germany, part of Springer Nature 2020, 2020

[15] C. Sciortino and A. Fagiolini, "ROS/Gazebo-Based Simulation of Quadcopter Aircrafts," 2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI), Palermo, Italy, 2018, pp. 1-6, doi: 10.1109/RTSI.2018.8548411. keywords: {Mathematical model;Rotors;Drones;Atmospheric modeling;Aircraft;Engines;Sensors;Robotics;nonlinear control;ROS/Gazebo;quad-copters},

[16] Shicheng Wu, Rui Li, Yingjing Shi, Qisheng Liu, "Vision-Based Target Detection and Tracking System for a Quadcopter," IEEE Access, vol. 9, pp. 62043-62054, 2021

[17] Vaibhav Mehandiratta, Mani Mehra, Günter Leugering, "An approach based on Haar wavelet for the approximation," 23 June 2019.

[18] Chu, Q., Ouyang, W., Li, H., et al.: Online multi-object tracking
using cnn-based single object tracker with spatial-temporal attention mechanism. In: IEEE International Conference on Computer
Vision (ICCV), pp. 4836–4845 (2017)

PHƯƠNG PHÁP KIỂM SOÁT THEO DÕI MỤC TIÊU TỰ ĐỘNG CHO QUADROTORS SỬ DỤNG TRÍ TUỆ NHÂN TẠO

Tóm tắt: Theo dõi các mục tiêu di chuyển là một ứng dụng hấp dẫn cho quadcopters và là một lĩnh vực nghiên cứu phức tạp do đặc tính khí động học của quadcopters và tốc độ thay đổi của mục tiêu di chuyển qua thời gian. Trong bài báo này, chúng tôi xây dựng một quadcopter để theo dõi mục tiêu bằng cách tích hợp một máy tính nhúng Raspberry Pi (RPI) với bộ điều khiển bay Pixhawk. Bài báo này cũng đề xuất một thuật toán Theo dõi nhẹ có thể triển khai trên Raspberry Pi, thuật toán này sử dụng xử lý hình ảnh tiên tiến và khả năng tính toán để cải thiện đáng kể hiệu suất theo dõi mục tiêu, từ đó giảm sự cần thiết của việc kiểm soát can thiệp của con người trong các chuyến bay không người lái. Việc điều khiển quadcopter bằng phương pháp này giúp hệ thống theo dõi duy trì ổn định trong môi trường mô phỏng và đạt được các thông số kiểm soát tích cực trong cài đặt thực tế.

Từ khóa: Quadrotors, Raspberry Pi, Pixhawk, UAV

Han Trong Thanh received the B.E., M.E., and Dr. Eng. degrees in Electronics and Telecommunications from Hanoi University of Science and Technology, Vietnam in 2008, 2010 and 2015, respectively. From July to September 2019, He was a visiting researcher in The University of Electro - Communication, Japan. He is currently an Assistant Professor at School of Electrical and Electronic Engineering, HUST. His research interests are Software Defined Radio, Advance Localization System and Signal processing for Medical Radar.

Cao Xuan Nghia is currently a Student at School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Vietnam. His research interests include deep learning, machine learning, digital image processing, computer vision and electronic device for IoT system.

Dinh Tan Hung is Doctor of Aeronautical Engineering, Kiev National Aviation University, Ukraine, 2010, Expert Engineer (Master) in External Economic Administration, Kiev National Aviation University, Ukraine, 2005. He is Master of Aeronautical Engineering, Kiev National Aviation University, Ukraine, 2003. Engineer of Aeronautics and Space Technology, Kiev National Aeronautical University, Ukraine, 2001