

## AUTONOMOUS DETECTION AND TRACKING OF AN OBJECT AUTONOMOUSLY USING AR.DRONE QUADCOPTER

Futuhul Arifin<sup>1</sup>, Ricky Arifandi Daniel<sup>1</sup>, and Didit Widiyanto<sup>2</sup>

<sup>1</sup>Faculty of Computer Science, Universitas Indonesia, Kampus UI Depok, Depok, 16424, Indonesia

<sup>2</sup>Faculty of Computer Science, UPN Veteran, Pondok Labu, Jakarta, 12450, Indonesia

E-mail: [ricky.arifandi@ui.ac.id](mailto:ricky.arifandi@ui.ac.id), [didit.widiyanto89@gmail.com](mailto:didit.widiyanto89@gmail.com)

### Abstract

Nowadays, there are many robotic applications being developed to do tasks autonomously without any interactions or commands from human. Therefore, developing a system which enables a robot to do surveillance such as detection and tracking of a moving object will lead us to more advanced tasks carried out by robots in the future. AR.Drone is a flying robot platform that is able to take role as UAV (Unmanned Aerial Vehicle). Usage of computer vision algorithm such as Hough Transform makes it possible for such system to be implemented on AR.Drone. In this research, the developed algorithm is able to detect and track an object with certain shape and color. Then the algorithm is successfully implemented on AR.Drone quadcopter for detection and tracking.

**Keywords:** *autonomous, detection, tracking, UAV, hough transform*

### Abstrak

Saat ini, ada banyak aplikasi robot yang telah dikembangkan untuk melakukan suatu tugas secara autonomous tanpa interaksi atau menerima perintah dari manusia. Oleh karena itu, mengembangkan sistem yang memungkinkan robot untuk melakukan tugas pengawasan seperti deteksi dan tracking terhadap suatu objek yang bergerak akan memungkinkan kita untuk mengimplementasikan tugas-tugas yang lebih canggih pada robot di masa mendatang. AR.Drone adalah salah satu platform robot terbang yang dapat berperan sebagai UAV (Unmanned Aerial Vehicle). Penggunaan algoritma *computer vision* seperti Hough Transform memungkinkan sistem semacam itu dapat terimplementasi pada AR.Drone. Pada penelitian ini, algoritma yang diterapkan mampu melakukan deteksi dan tracking terhadap suatu objek berdasarkan bentuk dan warna tertentu. Hasil yang diperoleh dari penelitian ini menunjukkan sistem deteksi dan tracking objek secara autonomous dapat diimplementasikan pada quadcopter AR.Drone.

**Kata Kunci:** *autonomous, deteksi, tracking, UAV, hough transform*

### 1. Introduction

Unmanned Aerial Vehicle (UAV) is a kind of flying vehicle without a pilot commanding it [1]. At first, UAVs were developed mainly for military purposes. After World War II, many countries developed UAVs to do staking, surveillance, and penetration to enemy area without any risk of losing men had the vehicle gotten attacked [2]. Nowadays UAVs are used by universities and research institutions as research objects in the computer science and robotics fields. These researches are mostly focused on surveillance, inspection, and search and rescue [3]. UAVs equipped with surveillance system supports such as camera and GPS have done well in national security and defense [4]. One of the most important component to do a surveillance using UAV is camera. Therefore, image processing is required to get informations fr-

om the camera. There are already many algorithms developed for that purpose.

UAV is divided into two categories based on its wing shape and body structure, the fixed wing ones and the rotary wing ones [1]. Quadrotor or quadcopter is one kind of UAVs that is frequently used as research object. This kind of UAV falls into the rotary wing category and has four propellers and motors. Quadcopters have a high maneuverability, it can hover, take off, cruise, and land in narrow area. Beside those, quadcopters have simpler control mechanism compared to the other UAVs.

Those researches above are usually focused on algorithm implementation and hardware usage such as sensors and cameras [4]. Many methods and algorithms are applied in quadcopters to carry out tasks. Addition of other sensors or hardwares are also frequently done to complete the imple-



Figure 1. An AR.Drone quadcopter using indoor hull.



Figure 2. An AR.Drone quadcopter using outdoor hull.

mentation. In this research, the quadcopter used are AR.Drone quadcopter using indoor hull (Figure 1) and AR.Drone quadcopter using outdoor hull (Figure 2).

## 2. Methods

### Detection and Tracking of an Object Autonomously Using AR.Drone Quadcopter

HSV, which stands for hue, saturation, and value defines a type of color space. Alvy Ray Smith created this color model in 1978. HSV is also commonly known as hex-cone color model. There are three components of HSV, the hue, saturation, and value. Hue component defines what color is being represented. It has the value of an angle of 0-360, red is defined in 0-60 range, yellow in 60-120 range, green in 120-180 range, cyan in 180-240 range, blue in 240-300 range, and magenta in 300-360 range. Saturation component indicates how much grey is in the color. It has the value range of 0%-100% or 0-1, with the lowest value meaning grey color and highest value meaning primary color. Grey color (low saturation) results in faded color. Value component defines the brightness of the color. It also has the value range of 0%-100% with lowest value meaning totally black and highest value meaning bright color [5].

RGB, which stands for red, green, and blue, is the computer's native color space, both for capturing images (input) or displaying them (output). It is based on human's eyes that are sensitive to these three primary colors. All other colors can be

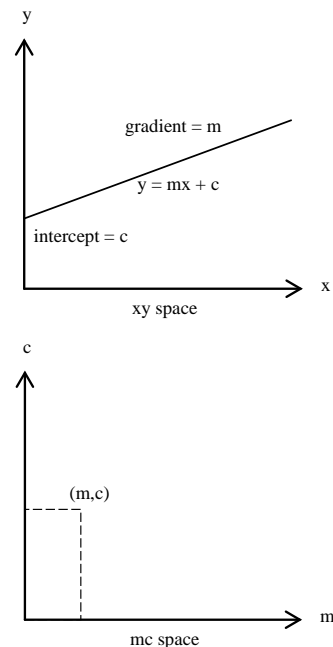


Figure 3. Transformation of a line from xy-space to point in mc-space.

constructed using combinations of these three colors in additive way. It uses Cartesian coordinate system to represent the value of each primary color and combine them to represent a given color [6].

Object detection is one of important fields in computer vision. The core of object detection is recognition of an object in images precisely. Applications such as image search or recognition use object detection method as its main part. Today, object detection problem is still categorized as open problem because of the complexity of the image or the object itself [7]. The common approach to detect object in videos is using the information from each frame of the video. But, this method has high error rate. Therefore, there are some detection methods that use temporary information computed from sequence of frames to reduce the detection error rate [8].

Object tracking, just like object detection, is one of important fields in computer vision. Object tracking can be defined as a process to track an object in a sequence of frames or images. Difficulty level of object tracking depends on the movement of the object, pattern change of the object and the background, changing object structure, occlusion of object by object or object by background, and camera movement. Object tracking is usually used in high-level application context that needs location and shape of an object from each frame[9]. There are three commonly known object tracking algorithms, Point Tracking, Kernel Trac-

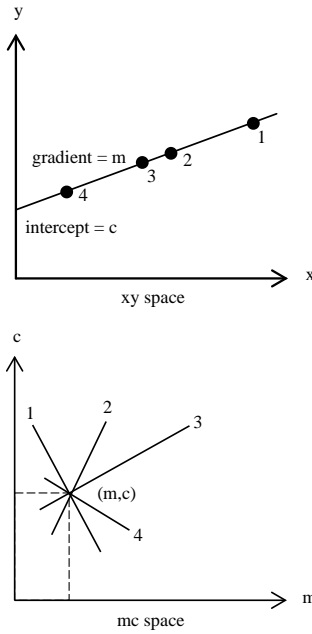


Figure 4. Transformation of points from xy-space to point in mc-space.

king, and Silhouette Tracking. Examples of object tracking application are traffic surveillance, automatic surveillance, interaction system, and vehicle navigation.

Hough Transform (HT) is a technique used to find shape in an image. It is originally used to find lines in an image. [10]. Because of its potential, many modifications and development have been conducted so that HT can not only find lines, but also other shapes like circles and ellipses in an image. Before HT can detect a shape, the image has to be preprocessed first to find the edge of each object using edge detection method such as Canny Edge Detector, Sobel Edge Detector, etc. Those edges will later be processed using HT to find the shape. A line in an image is constructed of points. Therefore, to find a line the first thing to do is represent a line into a point.

The Figure 3 shows a transformation of a line from xy-space to mc-space. Every line is associated with two values, gradient and intercept. The line from the image is represented in xy-space meanwhile the line with gradient and intercept is represented in mc-space. A line passing through point (x, y) has the property as given by equation(1).

$$y = mx + c \tag{1}$$

Moving the variables in equation (1) results in this equation(2).

$$c = -mx \tag{2}$$

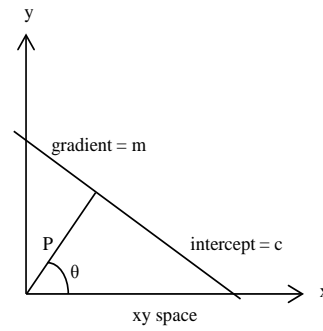


Figure 5. Transformation of points from xy-space to point in mc-space.

Transformation of a point from xy-space to a line in mc-space can also be done. HT converts point in xy-space to line in mc-space. Points from edge of an object in an image are transformed into line. Therefore, some lines constructed of points from edge of the same object will intersect. The intersection of the lines in mc-space is the information from the line in xy-space.

Points 1-4 in Figure 4 are represented into lines in mc-space. The intersection is equivalent to the line in xy-space.

The HT method explained above has a weakness when a vertical line is extracted from an image because the gradient (m) is infinite, therefore infinite space of memory is needed to store the data in mc-space. Therefore, another method is needed to overcome this problem. One of the solutions is using normal function to replace the gradient-intersect function.

In this representation, a line is constructed of two parameter values, distance p and angle  $\theta$ . P is the distance of a line from point (0, 0) and  $\theta$  is the angle of the line to x-axis. P value is in 0 to diagonal length of the image range and  $\theta$  value is in -90 to +90-degree of range. In this method, a line is re-presented as the equation(3).

$$p = x \cos \theta + y \sin \theta \tag{3}$$

where x and y is a point passed by the line. There is a slight change when this method is applied. A line in xy-space will still be transformed into a point in p $\theta$ -space, but, a point in xy-space will be transformed into a sinusoidal curve[11].

A circle can be defined as a center point (a,b) with radius R. Its relationship can be seen in equation(4) and equation(5):

$$\begin{aligned} x &= a + R \cos \theta \\ y &= b + R \sin \theta \end{aligned} \tag{4}$$

HT method finds the three values (x, y, R) to recognize a circle in an image. On every pixel of

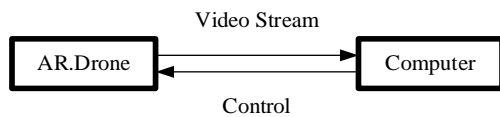


Figure 6. Communication process between AR.Drone and computer



Figure 7. The orange ball used in this research.



Figure 8. Video stream from AR.Drone before conversion in RGB color space.

an object, HT will create a circle with certain radius. The points which are intersections of two or more circles will be stored in voting box. The point with the highest vote count will then be acknowledged as the center point.

There are many variants of HT method that were developed from the standard HT, such as GHTG (Gerig Hough Transform with Gradient), 2-1 Hough Transform, and Fast Hough Transform. In OpenCV, these methods implementations are a bit more complicated than the standard HT. OpenCV implements another HT variant called Hough Gradient in which memory usage can be optimized. In Hough Gradient method the gradient direction is taken into account so that the amount of computation executed decreases.

In this research, development of an autonomous detection and tracking system of an object using AR.Drone is conducted. The term autonomous means the system can detect and track object independently without interactions from user/human. Detection means the robot is able to recognize certain object using its sensors. Track-



Figure 9. Video stream from AR.Drone after conversion in HSV color space.

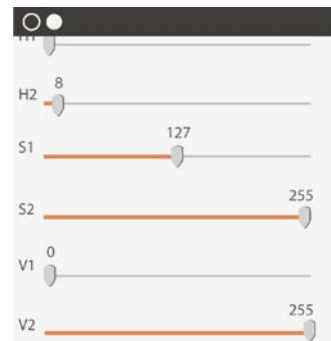


Figure 10. Graphical interface to change value of HSV components.

ing means robot is able to follow the said object movement.

AR.Drone has two cameras, frontal camera and vertical camera. This research is focused on usage of computer vision algorithm as the base of the developed system. Therefore, object detection is carried out using AR.Drone frontal camera as the main sensor.

Being a robot for toy, AR.Drone has limits in computational capacity. Meanwhile, the image processing with computer vision algorithm needs pretty high resources. With that concern, all computations executed in this system are conducted in a computer connected to the AR.Drone wirelessly. In short, the communication flow between AR.Drone and the computer can be seen in Figure 6.

Object detection program receives image or video stream from AR.Drone camera. Every frame of the video stream is processed one by one by the program. The computer vision algorithm will process the image and gives object information as output. The detected object in this research is an orange ball. If the object is not found, no output is given.

By default, images taken from AR.Drone camera is in RGB color space. Image processing will produce better result if the image processed is in HSV color space [12]. Therefore, the image has to be converted first to HSV color space in order to

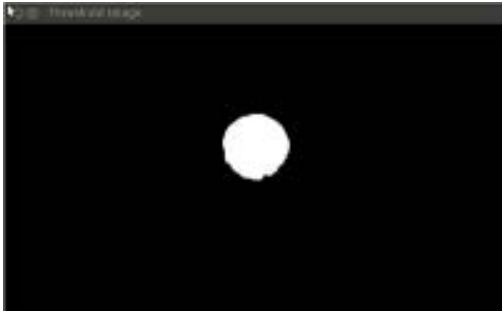


Figure 11. Image after thresholding process.



Figure 12. Object detected with distance of 6 meter.



Figure 13. Object detected with distance of 0.5 meter.

be processed by the program. After conversion, thresholding process is conducted. Thresholding process will divide the image into black and white. The HT method then detects the circle in the black and white image. Therefore, any object with certain shape and color can be detected by this object detection program.

### Parameter Settings

Image processing is conducted using libraries from OpenCV. The first thing to do is convert the image color model from RGB color space to HSV color space. The conversion is executed using `cv.cvtColor` function in OpenCV.

After conversion, the video stream from AR. Drone is converted into binary image format which only shows black and white. Detection process in this research is conducted based on object shape and color. Therefore, thresholding process needs to be done carefully to find the right threshold value. The term threshold means value range for

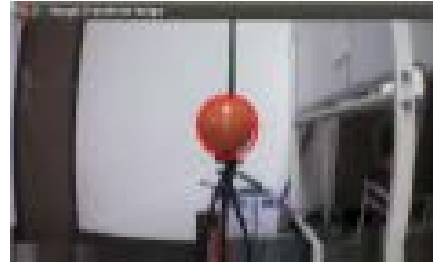


Figure 14. Object detected in simple environment.



Figure 15. Object detected in simple environment.

hue, saturation, and value components. This can be done using program provided by OpenCV.

After finding the right threshold value, the image is then processed using HT method to find the circle shape in it. The `cv.HoughCircles` method from OpenCV is used. This method implements HT method to find circles in an image.

In short, what the object detection program does is convert the image from AR.Drone to Open CV format, then process the converted image to check whether the object is in the image or not. If the object exists in the image, it tells the robot control program that the object is detected along with its coordinate and size (a, b, r).

After thresholding, the image shown in Figure 10 will be shown as in figure 11. The robot control program is implemented as tracker and controller. It receives information from object detection program about the detected object as stated above. The program then sets a rectangle inside the viewfinder of image from AR.Drone and makes sure that the (a, b) point is always inside that rectangle. That way, the AR.Drone tracks the object. The movement of the robot is controlled by setting the value of pitch, roll, and gaze of the AR. Drone.

### 3. Results and Analysis

There are two scenarios designed for experiments in this research. The first scenario tests how well AR.Drone can detect and approach the object. Second scenario is similar to the first scenario, but the AR.Drone does not directly face the object.

TABLE 1  
TEST RESULT FOR FIRST SCENARIO

Distance (m)	Test Results (s)					Average Time (s)
	1	2	3	4	5	
3	2.9	2.9	3.0	2.8	2.7	2.86
4	2.7	3.4	3.3	3.5	2.8	3.14
5	5.1	5.3	3.4	3.9	4.7	4.48
6	6.5	5.9	4.5	7.8	3.7	5.68

Before going on to the test scenarios, the detection system needs to be tested first. This test is conducted to find out how close or how far an object can be detected well.

System performance is also measured by testing in simple environment (no noise) and complex environment (much noise). In this test, the AR.Drone doesn't need to fly yet.

It can be seen in Figure 12 and Figure 13 that the object can be detected well from 0.5 meter up to 6 meter distance. If the object is further away, the object image will be too small to detect, and if the object is closer than 0.5 meter, the computation process will be too much of a burden for the computer.

The Figure 14 and Figure 15 above show how the system can detect the object in different environment settings. Simple environment means there is little to zero noise or any other objects with same color or shape. Complex environment means there is some noises.

A problem in this detection system occurs when the lighting condition is too dark or too bright. Bad lighting condition changes the image color so that the object color can't fall into the configured threshold, therefore can't be detected. Therefore, the tests are all conducted in normal lighting condition.

In first scenario, AR.Drone is placed 3 to 6 meter in front of object and then the program is run. The time needed for the AR.Drone from take-off to object approach is then noted down. Test is conducted five times for each distance (differing 1 meter each). The test result is shown in Table 1.

In second scenario, the setting is similar to the first scenario, but the AR.Drone does not directly face the object. AR.Drone has to rotate its orientation to find the object. AR.Drone is placed 5 meter away from the object with different orientations. There are three orientations tested in this scenario, the object is behind, on the left, and on the right of the AR.Drone. In the design, if the object is not found, AR.Drone will rotate its orientation about 45 degrees clockwise and there will be 3 seconds delay for every rotation. The test result is shown in Table 2.

In the second scenario, there's not many tests conducted, because it is only to test the system when faced with certain condition, not the speed.

TABLE 2  
TEST RESULT FOR SECOND SCENARIO

Object Position	Time (s)
Right	5.9
Left	11.7
Behind	8.3

#### 4. Conclusion

Conclusions of this research are as follows. Based on the tests conducted, implementation of detection and tracking system of an object is successfully carried out by AR.Drone. The system works with constraints of, the distance between the AR.Drone and the computer is not more than 25 meter, normal lighting condition (not too dark, not too bright), object is in horizontal position relative to the AR.Drone, and computation delay is about 1-2 seconds. Usage of frontal camera of the AR.Drone makes it only possible to detect object in horizontal position relative to the AR.Drone, therefore it's not possible yet to detect objects below or above the AR.Drone. The time needed to compute also makes the detection and tracking not real time, as can be seen in the AR.Drone delay in reacting to object change/movement. The factors that affect the delay time are the computation time and transmission time between AR.Drone and the computer. The further away the AR.Drone from the computer, the wireless signal becomes weaker and the transmission process becomes slower and frequently disconnects. In the developed system, AR.Drone can't determine the distance of the object from itself, it can only use the information of the radius of the object. Even so, with the current constraints and limitations of the system, this research has proven that detection and tracking system of an object can be implemented successfully on the AR.Drone quadcopter and a computer.

#### Reference

- [1] H. Chao, Y. Cao, and Y. Chen, "Autopilots for Small Unmanned Aerial Vehicles: A Survey," *International Journal of Control, Automation, and Systems*, 1995.
- [2] Henri Eisenbeiss, "A Mini Unmanned Aerial Vehicle (UAV): System Overview and Image Acquisition," November 2004.
- [3] Tom Krajn et al., "AR-Drone as a Platform for Robotic Research and Education," In *Research and Education in Robotics: EURO-BOT 2011*, Heidelberg, Springer, 2011.
- [4] S.D. Hanford, "A Small Semi-Autonomous Rotary-Wing Unmanned Air Vehicle," 2005.
- [5] "HSV (Hue, Saturation, and Value)," <http://www.tech-faq.com/hsv.html>, 2012, retrieved June 20, 2013.

- [6] "RGB (Red Green Blue)," <http://www.techfaq.com/rgb.html>, 2012, retrieved June 20, 2013.
- [7] Liming Wang et al., "Object Detection Combining Recognition and Segmentation," In Proceedings of the 8th Asian Conference on Computer Vision– Volume Part I (ACCV'07), Berlin, pp. 189-199, 2007.
- [8] Alper Yilmaz et al., "Object Tracking: A Survey," ACM Computing Surveys (CSUR), vol. 38, no. 13, 2006.
- [9] Alper Yilmaz et al., "Object Tracking: A Survey," ACM Comput. Surv., vol. 38, no. 4, December 2006.
- [10] G. Bradski and A. Kaehler, Learning Open CV Computer Vision with the OpenCV Library, O'Reilly, Sebastopol, 2011.
- [11] "The Hough Transform," <http://www.aishack.in/2010/03/the-hough-transform>, 2010, retrieved July 2, 2013.
- [12] S. Surkutlawar and R.K. Kulkarni, "Shadow Suppression using RGB and HSV Color Space in Moving Object Detection," International Journal of Advanced Computer Sciences and Applications, ISSN 2156-5570, vol. 4, 2013.