Vol. 1, No. 1, 2023
ISSN: 2987-3932
https://pubs.ast-ptm.or.id/index.php/eat


# Implementation of Otsu Thresholding Method for UAV Landing Phase Using Raspberry Pi

Husnibes Muchtar [a,1,*], Irtanto Wijaya [a,2]

[a] Muhammadiyah University of Jakarta, Jl. Cempaka putih tengah Jakarta Pusat 10510, Indonesia
[1] husnibes.muchtar@umj.ac.id*; [2] irtanto09@gmail.com
* Corresponding Author

ABSTRACT

Unmanned Aerial Vehicles (UAVs) in Indonesia are usually used for Aerial Photo and Aerial Video. UAVs are often used to take pictures from the air which will later be processed into an Orthophoto for surveying or monitoring purposes but often encounter problems during landing process due to insufficient land for runay and sometime this made UAV being damaged and the data taken some times was also damaged. One of the uses of technology to reduce UAV accident rate during landing process is using digital image processing which allows UAV to carry out the landing process by itself using a mini computer assistance called Raspberry Pi. Method: Otsu thresholding is used for Image processing. There are several stages in this digital image processing process, including image taking, rotating, converting HSV, grayscale, Otsu thresholding, object recognition and decision making. And the testing was carried out in ten experiments with different values of sunlight intensity. Results: For sunlight intensity values between 32,000 - 61,000 lux, UAV successfully landed with minor damage. Conclusions: Otsu thresholding method is proven to function in adapting to several light intensity values. There were only 2 failures due to the very small value of the sunlight intensity.

KEYWORDS
Unmanned Aerial Vehicle
Image Processing
Image Processing
Object Recognition
Raspberry Pi



## 1. Introduction

Indonesia is a country known for its vast tropical forests with biodiversity. Based on data obtained from the Ministry of Environment and Forestry, the total forestry in Indonesia in 2019 reached 94.1 million ha. It is necessary to periodically monitor it in order to record the forest area and prevent forest fires. In the monitoring process, tools that are capable of taking aerial photographs are needed. From this problem, the use of UAVs is very helpful and effective in carrying out these tasks. Unmanned Aerial Vehicle or abbreviated as UAV is an unmanned aircraft that can be controlled remotely by remote control and can also operate automatically [1]. UAVs have various civil and commercial applications. For instance, they have been used in environmental surveillance applications, including fire detection [2], event detection [3], object tracking such as vehicles [4-6], or traffic and crowd monitoring

In Indonesia UAV is often used to take pictures in the form of photos which will later be processed into orthophoto / photo maps for survey purposes. But in conditions where the air craft is on a mission to take pictures in a location in the form of a forest, the plane will find it difficult to land due to the narrow runway. And it is also possible that the aircraft will be damaged from minor to severe. Therefore, an efficient landing method is needed to reduce the possibility of damaging the aircraft and the aircraft can fly back to retrieve new data without going through the repair phase which is quite time-consuming and costly. One landing method that can be used is by crashing the aircraft into a net stretches vertically. But this method requires accurate pilot instincts and there is a high probability that the plane will fail to land in this way, therefore UAV pilots rarely use this method to land the aircraft. Landing by crashing the air craft into the net is much safer than bally landing, because the net will completely stop the plane up to a speed of 0 km / hour.

The UAV is equipped with a camera and a mini computer called Raspberry Pi for controlling it in order to crash itself into the net precisely. The Raspberry Pi is a single-board computer made by the Raspberry Pi Foundation that is used to teach basic computer science in schools and in developing countries [7].

The camera on a mini computer takes live video for image processing purposes. The mini computer frames the original window of the camera to divide it into several frames. The next process is to use the Otsu threshold method, which ignores any color other than the yellow circle in the middle of the net. And then the minicomputer will determine what the action the plane will take, allowing it to crash into the net. Thresholding is one of the simplest and fastest segmentation methods based on assumption that images are formed from regions with different grey levels [8-15].

In image processing, to automatically do threshold and to convert a grayscale image in to a binary image, Otsu's method is used. The Otsu method is a method of automatically selecting a threshold value from the histogram gray level through discrimination analysis. The histogram image is a value that allows it to be used as an illustration of the intensity of an image[16]. Otsu method is expected to be able to separate the object (foreground) and background [17].

In this method the threshold operation is regarded as the partitioning of the pixels of an image into two classes C0 and C1 (foreground and background) at grey-level t, C0 represents pixels within levels {0, 1, 2,…, t} and C1 represents pixels within levels {t + 1, t +2,….S-1}. In image segmentation process, determining an optimal threshold t is usually based on the estimates of dispersion and location of intensities in C1 and C2. As with various other algorithms, Otsu's algorithm uses the average sample value and the deviation to calculate the dispersion and the location. An optimal threshold can be determined by minimizing one of the following (equivalent) criterion functions with respect to t [18].

$$\lambda = \frac{\sigma_B^2}{\sigma_W^2}, \qquad \eta = \frac{\sigma_B^2}{\sigma_T^2}, \qquad \kappa = \frac{\sigma_T^2}{\sigma_W^2}$$

Where $\sigma_W^2$, $\sigma_B^2$, $\sigma_T^2$ are the within-class variance, between class variance, total variance, respectively and $\eta$ is the simplest. The optimal threshold $t$ is defined as

$$t = Arg\ Min\ \eta$$

Where

$$\sigma_T^2 = \sum_{i=0}^{S-1} [1 - \mu]^2 P_i, \quad \mu_T = \sum_{i=0}^{S-1} [iP_i]^2$$

$$\sigma_B^2 = W_0 W_1 (\mu_0 \mu_1), \quad W_0 = \sum_{i=0}^{t} P_i, \quad W_1 = 1 - W_0$$

$$\mu_1 = \frac{\mu_T - \mu_t}{1 - W_0}, \quad \mu_0 = \frac{\mu_T}{W_0}, \quad \mu_t = \sum_{i=0}^{t} (iP_i)$$

$$P_i = \frac{n_i}{n}$$

Where $ni$ is the number of pixels with grey-level $i$ and $n$ is the total number of pixels in a given image defined as

$$n = \sum_{i=0}^{S-1} (n_i)$$

$Pi$ is the probability of occurrence of grey-level $i$. For a selected threshold $t$ of a given image, the class probabilities $W_0$ and $W_1$ indicate the portions of the areas occupied by the classes $C_0$ and $C_1$. The class means $\mu_0$ and $\mu_1$ serve as estimates of the mean levels of the classes in the original grey-level image. The maximum value of $\eta$, can be used as a measure to evaluate the separability of classes $C_0$ and $C_1$ in the original image. It is uniquely determined within the range $0 \leq \eta \leq 1$. The lower bound (zero) is obtained when a given image has a single constant grey level, and the upper bound (unity) is obtained when two-valued images are given. The separable degree $\eta$ of the class, in the discrimination analysis, is

$$\eta = max\ \sigma_B^2$$

Finally, maximizing $\sigma_B^2$ to choose the optimal threshold,

$$T = Arg\ max\ \sigma_B^2$$

## 2. Method

### 2.1 Work System Flowchart

Flow chart shown in Fig. 1 shows how the system work. Starting camera as a device input and then it takes an image and then processing it in the image processing block.

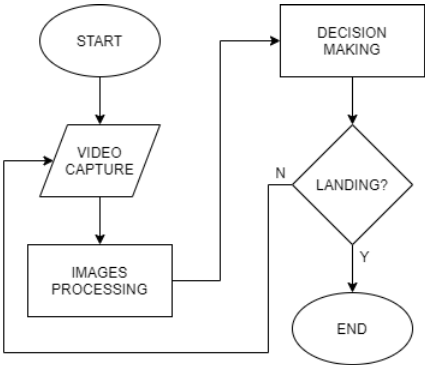


**Fig. 1.** UAV work system flowchart with image processing

After the image is processed, a decision will be made according to the target location. The target location is divided into several frames to facilitate image processing with frame mapping can be seen in Fig. 2.

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

**Fig. 2.** Frame mapping of target location

### 2.2 Hardware Connection Topology

In general, this system works begin with taking picture from a camera embedded in an unmanned air craft which is connected to Raspberry Pi. Image obtained from a camera will be processed in the Raspberry and then a decision is taken where the decision is in the form of controlling servo motor as output devices. Motor servo controls the motion of ailerons for controlling the movement UAV.

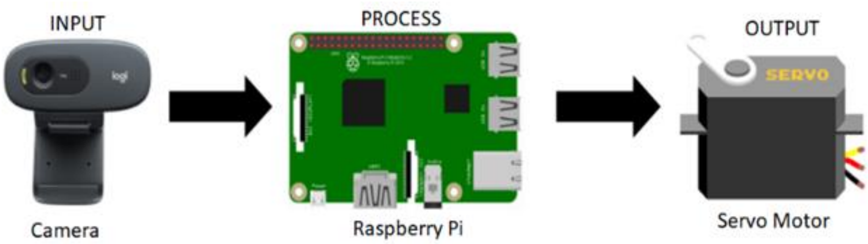


**Fig. 3.** Hardware connection topology

### 2.3 Flowchart of Image Processing

Fig. 4 shows how the system works on the images processing block. When the image has been taken, it will be processed in this block. First, the image will be rotated 90 degrees because of the position of the UAV and the net, the net is vertically upright and this process needs a frame that extends vertically.

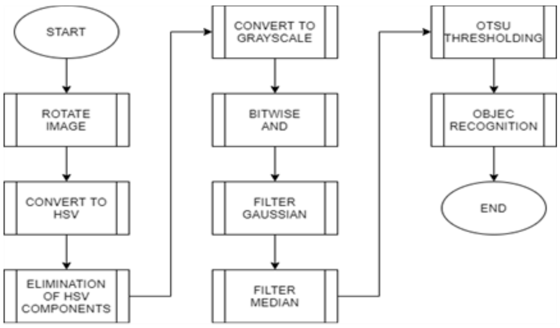


**Fig. 4.** Image processing block

After that, the image is converted to an image with HSV format and then the image obtained can be processed based on Hue value. With this Hue value, image processing is able to process saturation values and values in a wide range. Hue in HSV has a color definition with a range of numbers 0-179, while saturation can be defined as the level of density of a color [19]. HSV image will be eliminated and each component is separated, namely hue, saturation and value. Meanwhile, the HSV image will be limited to a certain hue value so that it will only produce the specified color and ignore other colors. After getting imitated image with HSV format and in grayscale format, the image will be combined with a bitwise operator process and resulting a grayscale image by ignoring the color in the HSV image other than the predetermined hue value. To reduce noises, two filters Gaussian and Median will be applied.

After the image is filtered, the image will enter floating process using the Otsu's thresholding method. With the final result in the form of a binary image, each pixel has a value of 1 bit. If the pixels intensity is greater than the determined threshold then it belongs to foreground class and otherwise to the background [20]. After the image is in binary format, the object in the image will be identified with the object recognition function which will identify the shape of the round object.

## 2.4 Flowchart of Decision Making

Flowchart in Fig. 5 shows how the decision making of work system based on the result of previous image enlargement. When the target is in the frame, it will be entered in to one of the frames (x = 420, y = 620) in accordance to the position of the target in the frame provided.

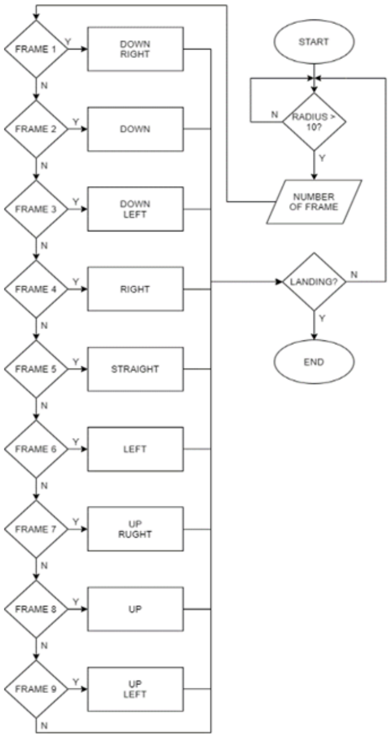


**Fig. 5.** Flowchart of decision making

The decision will be taken by adjusting the position of the target so as to make the aircraft position without being perpendicular to frame number 5 so that the unmanned aircraft can hit the net.

## 3. Results and Discussion

Testing was carried out in three stages, namely software testing, hardware testing and field testing. This test was carried out to determine that the output from the digital image processor for unmanned aircraft is in accordance with what was designed.

### 3.1 Software Testing

In this test, several tests were carried out based on the horizontal distance and vertical distance between the target and the camera on the UAV by considering a radius size more than 10 pixels.

**Table 1.** Test results at a vertical distance above the ground

| | Vertical distance of 1 meter | | | Vertical distance of 2 meter | | |
|---|---|---|---|---|---|---|
| **No** | **Horizontal distance** | **Radius** | **Detection** | **Horizontal distance** | **Radius** | **Detection** |
| 1 | 1 meter | 101 Pixels | Detected | 1 meter | 97 Pixels | Detected |
| 2 | 2 meters | 99 Pixels | Detected | 2 meters | 98 Pixels | Detected |
| 3 | 5 meters | 31 Pixels | Detected | 5 meters | 24 Pixels | Detected |
| 4 | 10 meters | 25 Pixels | Detected | 10 meters | 20 Pixels | Detected |
| 5 | 12 meters | 14 Pixels | Detected | 15 meters | 12 Pixels | Detected |
| 6 | 20 meters | 0 Pixels | Not detected | 20 meters | 0 Pixels | Not detected |

From the tests that have been carried out with varying horizontal distances. The maximum distance that can be detected by the camera is 15 meters with a radius of 14 pixels at a vertical distance of 1 meter and 12 pixels at a vertical distance of 2 meters.

### 3.2 Hardware Testing

Hardware testing aims to test direction of motion in the work plane of the UAV based on commands obtained from the results of digital image processing, so that the direction of motion in the work plane of the UAV can be corrected and will not make extreme movements. Tests are carried out in nine frames in order to get the direction of motion of the work plane from each frame. The test was carried out with the distance between the target and the UAV camera is 1 meter. The test results are shown in Table 2.

**Table 2.** Test results of motion direction of UAV based on frame number

| **Frame number** | **Command** | **Direction of motion** | | |
|---|---|---|---|---|
| | | **Right Aileron** | **Left Aileron** | **Elevator** |
| 1 | Move down right | Up | Down | Down |
| 2 | Move down | Straight | Straight | Down |
| 3 | Move down left | Down | Up | Down |
| 4 | Right | Up | Down | Straight |
| 5 | Straight | Straight | Straight | Straight |
| 6 | Move left | Down | Up | Straight |
| 7 | Move up right | Up | Down | Up |
| 8 | Move up | Straight | Straight | Up |
| 9 | Move up left | Down | Up | Up |

### 3.1 Field Testing

This test is carried out with the aim of obtaining the accuracy of landing an unmanned aircraft with varying light intensity values. Before making a flight, it is necessary to have several tests in order to make the flight safe. This test was carried out at the JGC Aeromodelling Field, East Jakarta and the Cikini College Vocational High School Soccer Field, North Jakarta. From the test results in the field with different light intensity values, it can be seen that digital image processing process was carried out successfully. From a total of ten experiments, when the light intensity is very low, the UAV cannot do the landing phase perfectly.

**Table 3.** Test results of motion direction of UAV based on frame number

| Test order | Intensity value (lux) | Result | Damage to Unmanned Aircraft |
|---|---|---|---|
| 1 | ±32000 | Success | Scratches on the front and left wing |
| 2 | ±45000 | Success | Scratches on the front |
| 3 | ±37000 | Success | Scratches on the front |
| 4 | ±12000 | Failed | - |
| 5 | ±36000 | Success | Scratches on the front and tail |
| 6 | ±61000 | Success | Scratches on the body |
| 7 | ±7000 | Failed | - |
| 8 | ±34000 | Success | Scratched on wings and tail |
| 9 | ±44000 | Success | Scratches on the front |
| 10 | ±60000 | Success | Scratches on the front |

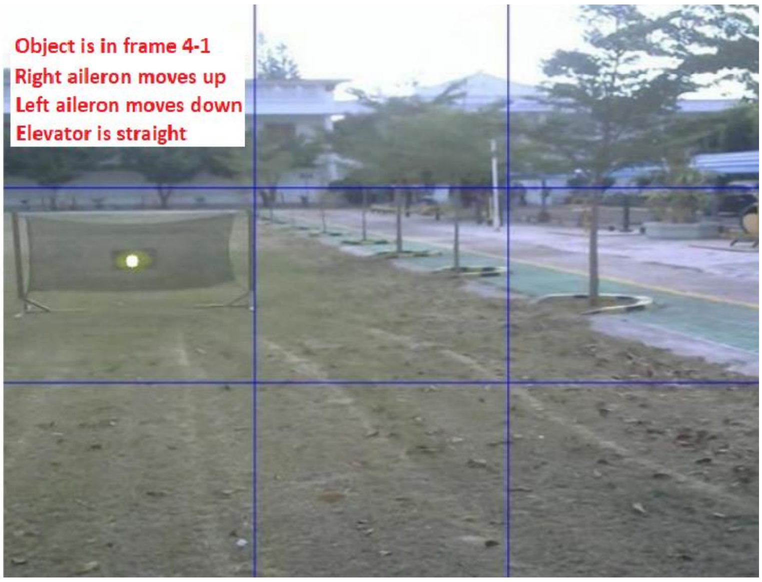


**Fig. 6.** One of field test results

## References

[1] K A Jbara, "A Robust Vision-Based Runaway Detection and Tracking Algorithm for Automatic UAV Landing, " King Abdullah University of Science and Technology, Thuwal, 2015.

[2] M. Kontitsis, K. P. Valavanis, and N. Tsourveloudis, "A uav vision system for airborne surveillance," in Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on, vol. 1. IEEE, 2004, pp. 77–83.

[3] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event detection and analysis from video streams," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 23, no. 8, pp. 873–889, 2001.

[4] S. Todorovic and M. C. Nechyba, "Intelligent missions for mavs: visual contexts for control, tracking and recognition," in Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on, vol. 2. IEEE, 2004, pp. 1640–1645.

[5] A. Ollero, J. Ferruz, F. Caballero, S. Hurtado, and L. Merino, "Motion compensation and object detection for autonomous helicopter visual navigation in the comets system," in Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on, vol. 1. IEEE, 2004, pp. 19–24.

[6] K. Kaaniche, B. Champion, C. Pegard, and P. Vasseur, "A vision algorithm for dynamic detection of moving vehicles with a uav," in Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on. IEEE, 2005, pp. 1878–1883.

[7] Meiyanto Eko Sulistyo, Stephanus Hanurjaya, Muhammad Danang Prastowo, " Monitoring Print Engine Output Using Arduino and Raspberry Pi," JEEICT, Vol.3, No.1, April 2021, Page 6-11.

Vol. 1, No. 1, 2023
ISSN: 2987-3932
https://pubs.ast-ptm.or.id/index.php/eat


[8] Sandhya. G, Giri Babu Kande, T. Satya Savithr, ”Multilevel Thresholding Method Based on Electromagnetism for Accurate Brain MRI Segmentation to Detect White Matter, Gray Matter, and CSF,” BioMed Research International. Vol.2017, ID (783209), 2017.

[9] J.H. Xue, D. Michael, ”Median-based image thresholding, Image and vision computing,” Vol.29, (2011), P.631-637.

[10] A. Yilmaz, O. Javed and M. Shah,” Object tracking: A survey,” ACM Computing Surveys. Vol.38, No.4, (2006), P.I-40.

[11] M. Sezgin, B. Sankur, ”Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging,” Vol.13, No.l, (2004), P.146-168.

[12] H F Ng, “Automatic thresholding for defect detection. Pattern Recognition Letters,” Vol.27, No.14, (2006), P.1644-1649.

[13] N. Otsu, ” A threshold selection method from gray level histograms,” IEEE Transactions on Systems, Man, and Cybernetics, 9:62-66,1979.

[14] J z Liu, W q Li, ”The automatic thresholding of gray-level pictures via two-dimensional Otsu method,” Acta Automatica Sinica, Voll9, No.l, (1993), P.101-105.

[15] J 1 Fan, F Zhao, X f Zhang, ”Recursive algorithm for three-dimensional Otsu's thresholding segmentation method,” Acta Electronica Sinica, Vol.35, No.7, (2007), P. 1398-1402.

[16] S. Dorafshan, M. Maguire and X. Qi, ”Automatic Surface Crack Detection in Concrete Structures Using Otsu Thresholding and Morphological Operations,” 2016, Utah State University, UTC-Report, 01-2016.

[17] N. Otsu, "A Threshold Selection Method," IEEE, Vols. 0018-9472/79/0100-0062$00.75, 1979.

[18] Suha. S, C.R. Venugopal, ”Hybrid Otsu Segmentation and Thresholding of Medical Images with Separability Factor,” International Journal of Advanced Science and Technology, Vol.29, No.11s, (2020), pp.671-675.

[19] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat and a. J. Jatakia, “Human Skin Detection Using RGB, HSV and YCbCr Color Models,” ICCASP/ICMMD-2016. Advances in Intelligent Systems Research, vol. 137, pp. 324-332, 2017.

[20] Jain B. D,” Goal directed evaluation of binarization methods,” IEEE Transactionson Pattern Analysis and Machine Intelligence, 17:1191-1200, 1995.