

**DOKUZ EYLÜL UNIVERSITY**  
**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**AUTONOMOUS FIELD SCANNING, TARGET  
DETECTION AND TRACKING VIA UAV**



by  
**Eren SİVRİTAŞ**

**October, 2019**  
**İZMİR**

# **AUTONOMOUS FIELD SCANNING, TARGET DETECTION AND TRACKING VIA UAV**

**A Thesis Submitted to the  
Graduate School of Natural and Applied Sciences of Dokuz Eylül University  
In Partial Fulfillment of the Requirements for the Degree of Master of Science  
in Electrical and Electronics Engineering**

**by  
Eren SİVRİTAŞ**

**October, 2019  
İZMİR**

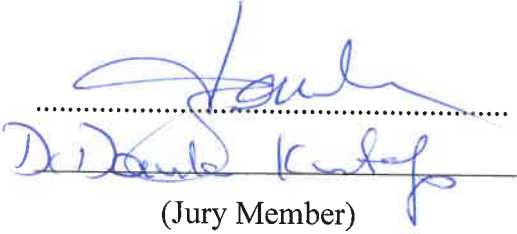
## M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “AUTONOMOUS FIELD SCANNING, TARGET DETECTION AND TRACKING VIA UAV” completed by **EREN SİVRİTAŞ** under supervision of **ASSOC. PROF. DR. ÖZGE CİHANBEĞENDİ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Assoc. Prof. Dr. Özge CİHANBEĞENDİ

Supervisor



(Jury Member)



(Jury Member)



Prof. Dr. Kadriye ERTEKİN

Director

Graduate School of Natural and Applied Sciences

## ACKNOWLEDGEMENTS

Firstly, I would like to express my deepest gratitude to my thesis supervisor Assoc. Prof. Dr. Özge Cihanbeğendi for all her advice, help and support even if it is weekend.

In addition, I would like to thank my family for all their endless patience, understanding, encouragement, trust and support at all time.

Finally, I would like to thank Dokuz Eylül University staff for their support and devotion during this period.

Eren SİVRİTAŞ

# **AUTONOMOUS FIELD SCANNING, TARGET DETECTION AND TRACKING VIA UAV**

## **ABSTRACT**

The use of unmanned air vehicles is becoming more and more important and this technology is developing rapidly. There is a significant trend towards the use of unmanned aerial vehicles, especially in the defense industry. In addition, unmanned aerial vehicles are being developed to be used for search and rescue, tracking, agricultural, mapping, photography, hobby or other purposes.

In parallel with the popularity of unmanned aerial vehicles, artificial intelligence technology is another area that has been developing rapidly in recent years. With the development of artificial intelligence technology, autonomous systems have been widely used in many fields. Unmanned aerial vehicles are one of these areas. As image processing technology and autonomous systems are time-saving, prevent human errors and are more cost-effective, demand for this technology is increasing day by day. Given the growing demand and the benefits of the resulting product, this area has been the subject of more scientific research. Although there are many scientific studies related to target detection and tracking in literature researches, there are not many studies related to detection and tracking systems by searching autonomously for the desired target.

The aim of this thesis is to produce an unmanned aerial vehicle that is effective in terms of weight, flight time and cost and to contribute to this research field which is developed and grown by means of a system which is created by combining this unmanned aerial vehicle with an algorithm that will autonomously search, detect and follow the defined target in the identified area. Within the scope of the thesis, two main topics, image processing and unmanned aerial vehicle control, were studied.

**Keywords:** Unmanned aerial vehicles, drone, autonomous systems, field scanning, target detection, target tracking

# İNSANSIZ HAVA ARACI İLE OTONOM OLARAK ALAN TARAMA, HEDEF TESPİTİ VE TAKİBİ

## ÖZ

İnsansız hava araçlarının kullanımı her geçen gün daha fazla önem kazanmakta ve bu teknoloji hızla gelişmektedir. Özellikle savunma sanayisinde insansız hava araçlarının kullanımına yönelik önemli bir eğilim bulunmaktadır. Buna ek olarak arama-kurtarma, takip, zirai, haritacılık, fotoğrafçılık, hobi veya başka amaçlar için kullanılmak üzere de insansız hava araçları geliştirilmektedir.

İnsansız hava araçlarının kazandığı popülerliğe paralel olarak son yıllarda hızla gelişen bir diğer alansa yapay zeka teknolojisidir. Yapay zeka teknolojisinin gelişmesiyle birlikte otonom sistemler birçok alanda yaygın olarak kullanılmaya başlanmıştır. İnsansız hava araçları da bu alanlardan bir tanesidir. Görüntü işleme teknolojisini ve otonom sistemleri kullanmak zaman kazandıran, olası insan hatalarını önleyen ve daha düşük maliyetli bir yöntem olduğu için bu teknolojiye olan talep her geçen gün artmaktadır. Artan talep ve ortaya çıkan ürünün faydaları göz önüne alınınca bu alan daha fazla bilimsel araştırmaya konu olmuştur. Literatür araştırmalarında hedef tespiti ve takibi ile alakalı çok sayıda bilimsel çalışma olsa da otonom olarak istenilen hedefi arayarak tespit ve takip eden sistemlere yönelik çalışmaların sayısı çok da fazla değildir.

Bu tezin amacı ağırlık, uçuş süresi ve maliyet olarak etkin bir insansız hava aracı üretmek ve bu insansız hava aracını tanımlanan hedefi belirlenen alanda otonom olarak arayacak, tespit edecek ve takip edecek bir algoritma ile birleştirerek oluşturulan sistem aracılığıyla gelişen ve büyüyen bu araştırma alanına katkıda bulunmaktır. Tez kapsamında görüntü işleme ve insansız hava aracı kontrolü olmak üzere iki ana başlık altında çalışma yapılmıştır.

**Anahtar kelimeler:** İnsansız hava aracı, drone, otonom sistemler, alan tarama, hedef tespiti, hedef takibi

## CONTENTS

	Page
M.Sc THESIS EXAMINATION RESULT FORM .....	ii
ACKNOWLEDGEMENTS.....	iii
ABSTRACT .....	iv
ÖZ .....	v
LIST OF FIGURES .....	ix
LIST OF TABLES .....	xi

## CHAPTER ONE - INTRODUCTION.....1

1.1 History of UAVs .....	2
1.2 Classification of UAVs .....	5
1.2.1 Very Small UAVs .....	5
1.2.2 Small UAVs .....	5
1.2.3 Medium UAVs .....	5
1.2.4 Large UAVs .....	6
1.3 Related Works .....	6
1.4 Restriction of Flying Drone .....	7
1.5 Aim of the Study .....	7
1.6 Thesis Outline .....	7

## CHAPTER TWO - METHODOLOGY AND DESIGN .....8

2.1 Image processing .....	8
2.1.1 Python Language .....	9
2.1.2 OpenCV Library .....	9
2.1.3 Gaussian Filtering.....	10
2.1.4 Erosion.....	10
2.1.5 Dilation .....	11

2.1.6 Mini PC .....	15
2.1.7 Camera Module .....	16
2.2 UAV Control .....	17
2.2.1 Components of the UAV .....	18
2.2.1.1 Electrical Components .....	18
2.2.1.1.1 Brushless Motors .....	18
2.2.1.1.2 Battery .....	19
2.2.1.2 Electronics Components .....	20
2.2.1.2.1 Flight Controller .....	20
2.2.1.2.2 GPS and Compass .....	21
2.2.1.2.3 Remote Controller .....	22
2.2.1.2.4 Remote Controller Receiver .....	23
2.2.1.2.5 Telemetry Transceiver .....	23
2.2.1.2.6 Battery Indicator .....	24
2.2.1.2.7 ESC (Electronic Speed Controller) .....	25
2.2.1.2.8 Power Switch .....	26
2.2.1.3 Mechanical Components .....	27
2.2.1.3.1 Frame .....	27
2.2.1.3.2 Propellers .....	28
2.2.2 PID Controller .....	29
2.2.3 Ground Control Station (GCS) .....	29
2.3 Design Steps .....	30
2.3.1 Definition of Target .....	30
2.3.2 Determination of the Field to be Scanned .....	31
2.3.3 Detection of the Target .....	35
2.3.4 Detection of the Moving Direction if the Target Moves .....	36
2.3.5 Producing and Sending the Necessary Commands to UAV System by Mini PC for Tracking the Target .....	41
2.3.6 Moving to Updated Target Location via UAV and Keeping Tracking ...	45
2.4 Cost Analysis .....	48

<b>CHAPTER THREE - CONCLUSION .....</b>	<b>49</b>
<b>REFERENCES.....</b>	<b>51</b>
<b>APPENDICES .....</b>	<b>54</b>
APPENDIX 1: Code Snippets for Filtering .....	55
APPENDIX 2: Code Snippets for Dimensional Detection .....	56
APPENDIX 3: Code Snippets for Finding HSV Values .....	57
APPENDIX 4: Code Snippets to Start Field Scanning .....	58
APPENDIX 5: Code Snippets to Compute Necessary Parameters for Moving to Updated Location .....	59

## LIST OF FIGURES

	<b>Page</b>
Figure 1.1 The first documented UAV in history, the pigeon .....	2
Figure 1.2 Aerial screw design of Leonardo da Vinci.....	3
Figure 1.3 (a) Predator UAV (b) Heron UAV .....	4
Figure 1.4 (a) Aksungur UAV (b) Akıncı UAV .....	5
Figure 2.1 Block diagram of the system working principle.....	8
Figure 2.2 Flow chart of the algorithm ....	13
Figure 2.3 The photo of the mini PC (Raspberry pi 3 Model B+).....	15
Figure 2.4 Camera board of mini PC .....	16
Figure 2.5 2212 920 Kv Brushless DC motor .....	18
Figure 2.6 Battery block of UAV .....	19
Figure 2.7 Kakute F7 flight controller (a) Top view (b) Bottom view .....	20
Figure 2.8 GPS and dual compass module (a) Top view (b) Bottom view .....	21
Figure 2.9 Six-channels remote controller .....	22
Figure 2.10 Remote controller receiver .....	23
Figure 2.11 Telemetry module transceivers.....	23
Figure 2.12 Battery vantage level indicator .....	24
Figure 2.13 Electronic speed controller (ESC) (a) Top view (b) Bottom view .....	25
Figure 2.14 Power switch .....	26
Figure 2.15 3D printer drawings of some plastic parts of the UAV .....	27
Figure 2.16 Body photo of the UAV .....	28
Figure 2.17 Aluminium arm of the UAV .....	28
Figure 2.18 9450 Plastic propellers .....	29
Figure 2.19 Mission planner GCS (Ground control station).....	30
Figure 2.20 The illustration of the field of view .....	32
Figure 2.21 The illustration of the field of view of the camera in terms of altitude (h).....	32
Figure 2.22 Angle of view of the camera in vertical axis .....	33
Figure 2.23 Angle of view of the camera in horizontal axis.....	33
Figure 2.24 Scanning field routing output of mission planner .....	34

Figure 2.25 Object (car) detection in center position (a) Real frame of target (b) Masked image of target .....	35
Figure 2.26 Object (car) detection in left-forward diagonal position (a) Real frame of target (b) Masked image of target .....	35
Figure 2.27 Object (car) detection in right-forward diagonal position (a) Real frame of target (b) Masked image of target.....	36
Figure 2.28 Center frame inside field of view .....	37
Figure 2.29 The car stays at center position .....	37
Figure 2.30 Pixel boundaries of directions .....	38
Figure 2.31 Leaving the center frame in the direction of the horizontal axis (a) Through left direction (b) Through right direction .....	39
Figure 2.32 Leaving the center frame in the direction of the vertical axis (a) Through forward direction (b) Through backward direction.....	39
Figure 2.33 Leaving the center frame in the direction of the diagonal axis (a) Through left-forward diagonal direction (b) Through left-backward diagonal direction.....	40
Figure 2.34 Leaving the center frame in the direction of the diagonal axis (a) Through right-forward diagonal direction (b) Through right-backward diagonal direction.....	41
Figure 2.35 View of the world in terms of latitude and longitude.....	42
Figure 2.36 Illustration of radii according to latitude angle .....	43
Figure 2.37 Illustration of arcs in terms of radii .....	44
Figure 2.38 A photo of the UAV during the tracking operation.....	45
Figure 2.39 Mini PC command output during searching and tracking mission .....	46
Figure 2.40 Routing output of mission planner during searching and tracking mission.....	47
Figure 3.1 Illustration of searching mission by UAV group .....	50

## LIST OF TABLES

	<b>Page</b>
Table 2.1 Simulation target (car) dimensions . . . . .	30
Table 2.2 Actual and simulation values of scanning field . . . . .	33
Table 2.3 Unit pixel size . . . . .	36
Table 2.4 Detailed cost analysis of the autonomous UAV system . . . . .	48



## **CHAPTER ONE**

### **INTRODUCTION**

Unmanned air vehicles (UAVs) are being used in more and more areas and the use of UAVs in human life is increasing (Kumar, Vaddi & Sarkar, 2018).

An unmanned aerial vehicle (or drone) means a pilotless aircraft, a flying machine without a human pilot or passengers inside the vehicle. Control functions for unmanned air vehicles may be either onboard or remote control (Valavanis, 2007).

In parallel with the expansion of the UAV sector, with the rapid development of artificial intelligence technology in recent years, autonomous systems have started to spread in every field and this trend is increasing rapidly. Recently, autonomous systems have also been integrated into unmanned aerial vehicles. In line with the demands of the users, rapidly increasing researches and investments are made in this field. The defense industry, closely monitored by governments, search & rescue and tracking are the main areas of the studies. In addition to these areas, a great interest and demand were occurred by people who want to use UAVs for the purpose of scanning related fields in agriculture or industry and to use for mapping, photography or hobby purposes. Developments in the field of autonomous systems are important in terms of reducing the labor force, reducing the probability of errors and minimizing the possible accident risks (Keipour, Mousaei & Scherer, 2019).

This study consists of two main parts:

- Image processing
- UAV control

More specifically, the study can be divided into 6 sub-groups:

- Definition of target
- Determination of the field to be scanned
- Determination of the target
- Detection of the moving direction if the target moves

- Producing and sending the necessary commands to UAV system by mini PC for tracking the target
- Moving to updated target location via UAV and keeping tracking

While all of these stages are being done a Ground Control Station (GCS) is used for observation of the system.

## 1.1 History of UAVs

First “flying machine” design was planned almost 2,500 years ago, in ancient Greece and China (Valavanis, 2007).

In 425 B.C., a mechanical bird, which is called as “the pigeon” was built by the ancient Greek mathematician Archytas. Depiction of flying pigeon is shown in Figure 1.1 and this is the first documented UAV in history (Valavanis & Vachtsevanos, 2015).



Figure 1.1 The first documented UAV in history, the pigeon (Valavanis & Vachtsevanos, 2015)

In 1483, Leonardo Da Vinci designed an aircraft called "aerial screw", shown in Figure 1.2. This machine is considered as the ancestor of helicopter (Valavanis & Vachtsevanos, 2015).

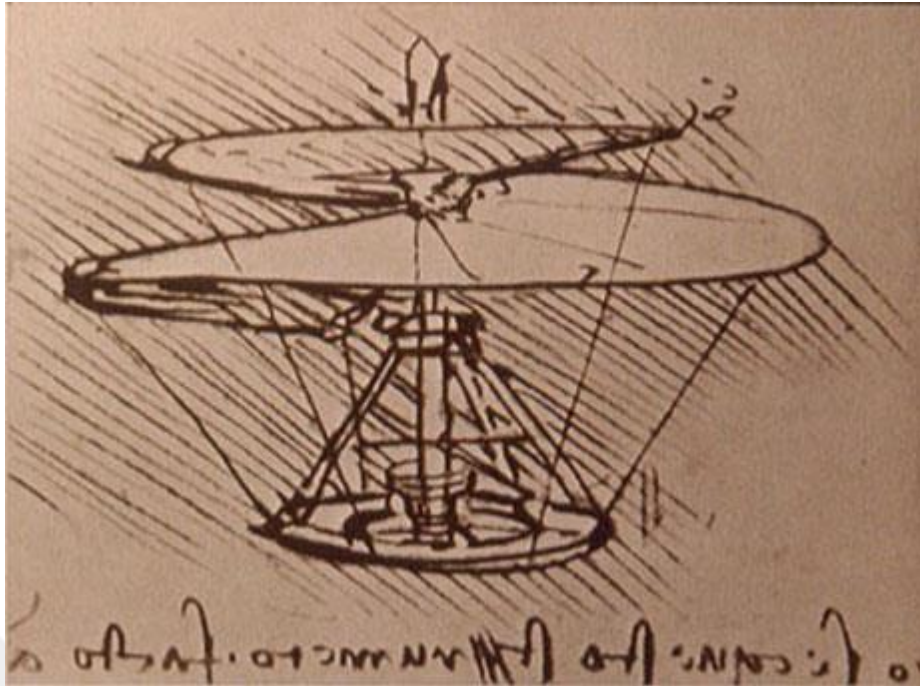


Figure 1.2 Aerial screw design of Leonardo da Vinci (Valavanis & Vachtsevanos, 2015)

On 22 August 1849, the UAV was used for the first time for military purpose by the Austrians and they attacked Venice using unmanned balloons loaded with explosives (Medium, 2019).

In 1883, Douglas Archibald integrated anemometer into the kite and measured wind velocity. Cameras were attached to kites in 1887 by Mr. Archibald, thus one of the world's first reconnaissance UAVs was created (Fahlstrom & Gleason, 2012).

Elmer Sperry, was developing the first gyroscope control system. Sperry started to work with airframe designer Glenn Curtiss and he began adopting his system of control on aircraft. On March 6, 1918, the Curtiss-Sperry Aerial Torpedo was successfully launched unmanned, completed its mission and landed. The Aerial Torpedo is the first modern unmanned aircraft (Marshall, Barnhart, Shappee & Most, 2016). Using gyroscope is known as the beginning of "attitude control", which is used for the automatic steering of an aircraft (Nonami, Kendoul, Satoshi, Wang & Nakazawa, 2010). Radio control techniques were used to control this unmanned aircraft (Custers, 2016).

After 1920, airplanes were transformed into drones. The Larynx (1927), the Fairy queen (1931) and the DH.82B Queen Bee can be given as examples for these drones. The name Queen Bee is said to have led to the use of the term ‘drone’ (a male bee) for unmanned aircraft (Custers, 2016). These first drones are also referred to as "Target drone".

In the mid-1950s, the first reconnaissance drone named as the SD-1 or known as the MQM-57 Falconer was developed. The target drones became the basis of this drone during developing stage. The SD-1 carried a camera and it was remotely operated (Valavanis & Vachtsevanos, 2015).

Over the years, more modern systems have been developed. CL-289, RQ-2 Pioneer, Sperwer, RQ-4 Global Hawk and MQ-8 are some of them (Valavanis & Vachtsevanos, 2015). Predator and IAI Heron UAVs are some of the most well-known UAVs in the world. These UAVs can be seen in Figure 1.3. Aksungur and Akıncı UAVs are the newest models of the UAV family and the photos of these UAVs are given in Figure 1.4.



Figure 1.3 (a) Predator UAV (Howstuffworks, 2019) (b) Heron UAV (Ainonline, 2019)



Figure 1.4 (a) Aksungur UAV (Militaryfactory, 2019) (b) Akıncı UAV (Defenceturk, 2019)

## 1.2 Classification of UAVs

There are many classification methods in literature. These classifications may vary according to countries and institutions/departments. For this study, classification according to dimensions is more useful. Size classes are not standardized, but they are appropriate from the general point of view (Fahlstrom & Gleason, 2012).

### 1.2.1 Very Small UAVs

The very small UAVs range starts from a few cm, mentioned as micro sized, and goes up to a length of 50 cm. Most of the drones can be counted in this class.

### 1.2.2 Small UAVs

One of the dimensions of Small UAVs is greater than 50 cm and the maximum dimension can be up to two meters long. Bayraktar and Raven B can be examples for this category.

### 1.2.3 Medium UAVs

If the UAV is large to be carried by one person but small against to light aircraft, this UAV may be posited in Medium UAVs category. These UAVs' wingspans are about 5–10 meters and the UAVs can carry payloads around 200 kg. Pioneer, Eagle eye and Fire scout are some examples of Medium UAVs class.

#### ***1.2.4 Large UAVs***

All other UAVs that are larger than below categories are called Large. Aksungur, Predator and Global Hawk are in this category (Fahlstrom & Gleason, 2012).

### **1.3 Related Works**

In the literature search, many projects have been implemented related to autonomy and UAV systems.

A study was implemented in 2014 to design automated surveillance quadcopter with image recognition for indoor environments. This project is customized for indoor environments (Moran, Borja & Casupanan, 2014).

There are some studies in 2018 (Codevilla, Müller, López, Koltun & Dosovitskiy) and 2019 (Kumar, et al., 2019) working on autonomy technology by using deep learning modeling.

Another study was carried out in 2019 with the aim of designing a low-cost search & rescue drone. Deep learning & neural network techniques were used for this study (McClure, 2019).

A useful project which works autonomously was implemented in 2019 (Lin, Wang, Peng & Yang, 2019), but this product is heavy, expensive and has short flight duration. Therefore, these features are not suitable with the aim of this thesis study.

In this study, different from the others, an effective UAV has been produced in terms of cost, flight time and weight. The system works autonomously from the beginning of the mission until the end of mission. The UAV can work in outdoor environment. The target requirements can be easily modified to adapt to new target. Scanning field, scanning route, scanning distance steps and scanning speed can be easily changed. The system does not need to study and time for deep learning

modeling. In case of the mechanical parts are damaged after the accident, plastic materials can be easily recovered by obtaining them from the 3D printer. The combination of all these features makes this study different and makes the output product of the project useful.

#### **1.4 Restriction of Flying Drone**

There are some procedures that need to be performed in order to start UAV flight. These procedures may vary by countries or regions. The parameters such as weight of the UAV, the location of the flight area, the distance from houses affect these procedures. Permits are required from institutions such as governorship, gendarmerie and directorate general of civil aviation. Permissions have been obtained from the relevant places for flights made within the scope of this project.

#### **1.5 Aim of the Study**

The aims of this thesis are to produce an unmanned aerial vehicle that is effective in terms of weight, flight time and cost, to create a system by combining this unmanned aerial vehicle with an algorithm that will autonomously search, detect and follow the defined target in the identified area, and to contribute to developing and growing scientific research area by means of this combined system. Within the scope of the thesis, two main topics, image processing and unmanned aerial vehicle control, were studied.

#### **1.6 Thesis Outline**

This thesis consists of three chapters. In the first chapter, Introduction, brief information was given about UAV. Also, History of UAV and related works were told in this chapter. In the second chapter, Methodology and Design, the details and components of the system, design steps, results of the study and cost analysis were written. In the third chapter, conclusion of the thesis was given. Finally, all references and appendices were listed at the end of the thesis.

## CHAPTER TWO

### METHODOLOGY AND DESIGN

The result of the study will be demonstrated with the combination of both image processing and UAV control phases. A block diagram of the system is given in Figure 2.1.

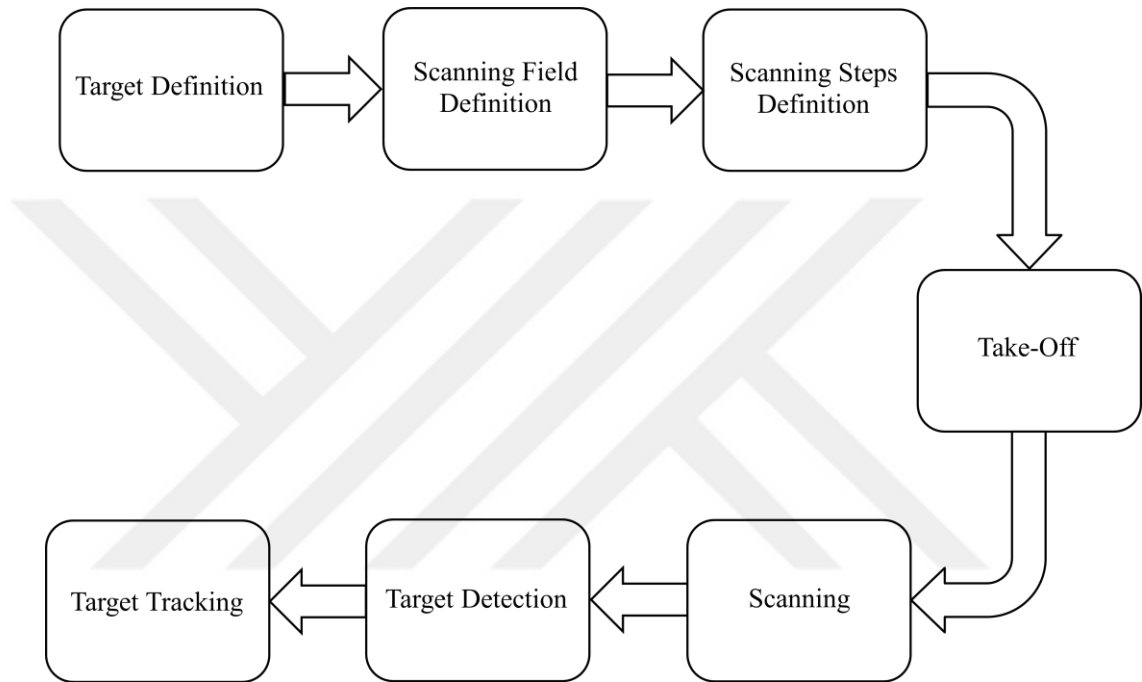


Figure 2.1 Block diagram of the system working principle

### 2.1 Image processing

Image processing is a kind of operation on an image. Digital image processing means that digital images are processed by a digital computer. Inputs and outputs of the process are images and the digital images composed of a finite number of pixels. Image processing is very important for many purposes such as classification, extraction, pattern recognition, image enhancement, etc. Basically, color spaces, filters and transforms are used to achieve the goal. HSV color space, Gaussian filtering, Erosion and Dilation morphological transforms were used for this project (Gonzalez & Woods, 2002).

A mini PC was used for image processing process. Also, a camera module which is compatible with mini computer was used. The Python language is very popular as a software language for image processing technology, so the Python language has been chosen for the study. In addition, the OpenCV (Open Source Computer Vision Library) library, which is widely used in the image processing area, was also utilized (Kumar, et al., 2019; Moran, et al., 2014; McClure, 2019; Vaddi, Jannesari & Kumar, 2019).

### ***2.1.1 Python Language***

Python is an interpreted high-level programming language, so Python programs can run immediately without long compiling process. Since Python is a very important programming language for many technological areas, it has gained in popularity in the last decade. It can be used for general purposes, but especially it is widely used in the fields of Artificial intelligence (AI), machine learning, deep learning, neural networks and robotics (Villán, 2019; Lutz, 2009).

Google and YouTube use Python because Python has many advantages in terms of adaptation, maintenance and development. Python has a simple syntax and it makes the code easier to understand. In some cases, Python reduces the length of the code by around 80% compared to other programming languages. Most Python programs can be run on many computer platforms without any change. Porting Python code between Linux and Windows is very easy. Python also has a standard library which is large collection of prebuilt and portable functions. Python's ease of use makes programming more pleasure than hassle (Villán, 2019; Lutz, 2009).

### ***2.1.2 OpenCV Library***

OpenCV is a library type for programming languages. It is open source and widely used in an image processing area. OpenCV provides ease of use for the computer vision infrastructure. It is optimized to take advantage of multi-core processors and provides high performance even when real time process is required.

OpenCV is generally used with Python and it is compatible with C++, Java, Matlab and Ruby as well. It supports many operating systems such as Windows, Linux, Macintosh and Android (Villán, 2019; Bradski & Kaehler, 2008).

Since, machine learning is used in computer vision often, OpenCV is also compatible with machine learning and deep learning technologies. It includes a full, general-purpose Machine Learning Library (MLL) that focuses on statistical pattern recognition and clustering (Villán, 2019; Bradski & Kaehler, 2008).

### ***2.1.3 Gaussian Filtering***

Images may contain different types of noise, like other types of signals. Sources (camera sensors) are one of the main reasons of noises (TutorialKart, 2019). We can use a blurring (smoothing) filter to eliminate noises and camera artifacts occurred in an image (BogoToBogo, 2019; Bradski & Kaehler, 2008).

OpenCV has a special function to blur an image by using a Gaussian kernel (Villán, 2019). This is the most commonly used method for image processing (BogoToBogo, 2019). The Gaussian filter is a low pass filter, so it removes the high frequency components (Tutorialspoint, 2019). Gaussian filtering is done by convolution of each point in the input array with a defined Gaussian kernel and output array is produced (OpenCV, 2019). Width and height of kernel should be specified as positive and odd number (OpenCV, 2019). Smoothing can also be used to reduce the resolution of an image (Bradski & Kaehler, 2008).

### ***2.1.4 Erosion***

The Erode function is generally used for removing small noises (GeeksforGeeks, 2019). Main working principle of the function is eroding away the boundaries of the object (OpenCV, 2019). This erosion causes the area of the object to be smaller than original dimensions (Villán, 2019).

The Erode function generates a new image from the original image by scanning over the image with the specified kernel (Bradski & Kaehler, 2008). A pixel in the original image will be set as 1 only if all the pixels under the kernel is 1, otherwise related pixel will be set to 0 and it will be eroded. Thus all the pixels near boundary will be removed. This function can also be used for detaching two connected objects (GeeksforGeeks, 2019).

### ***2.1.5 Dilation***

The dilation operation is generally used to expand the boundary of the object. The dilation causes the area of the object to be larger than the original dimensions (Villán, 2019). The Dilate function generates a new image from the original image by scanning over the image with the specified kernel (Bradski & Kaehler, 2008). This operation is just opposite of erosion. In Dilation operation, the pixel is set to '0' if there is not any pixel of "1" under the kernel. Otherwise, related pixel is set to "1". So, it increases the size of object (OpenCV, 2019). Dilation is a good method to accentuate the features (GeeksforGeeks, 2019).

In general, dilation expands the region whereas erosion reduces related region (Bradski & Kaehler, 2008). Normally, erosion is followed by dilation for noise filtering purpose. Because, while noises are removed by erode function, the object is also shrunk. So, dilate function is used to expand the shrunk object. Because, the noise is gone and it will never be occurred again. So, object area can be increased again (OpenCV, 2019). Briefly, dilation smooths the concavities and erosion smooths away the protrusions. The exact result depends on the kernel, but these results are generally obtained (Bradski & Kaehler, 2008).

An example code snippet for color space conversion, filtering and morphological transformations can be seen in appendix 1.

A simple and user-friendly algorithm was designed for this mission. All parameters can be modified easily and the system can be adapted for another mission which has different target. Flowchart of the algorithm is given in Figure 2.2.



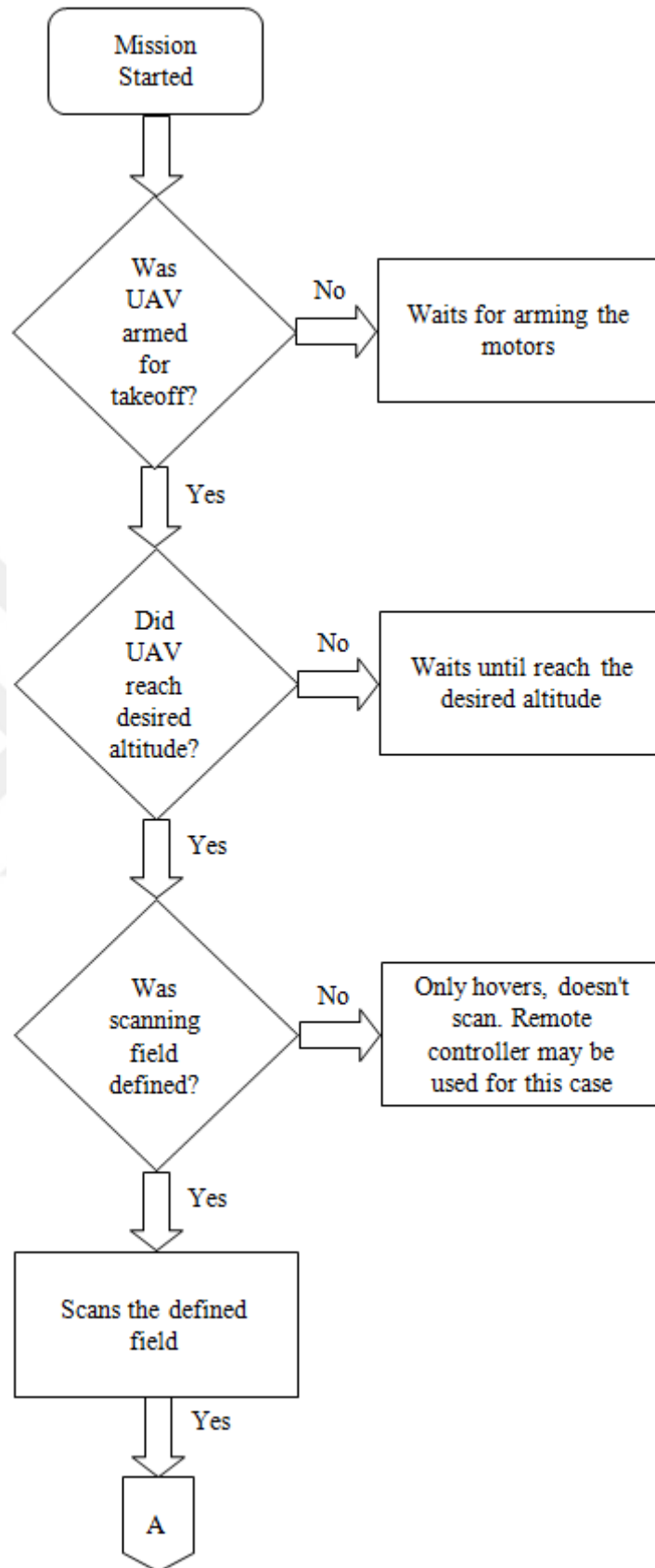


Figure 2.2 (a) Flow chart of the algorithm

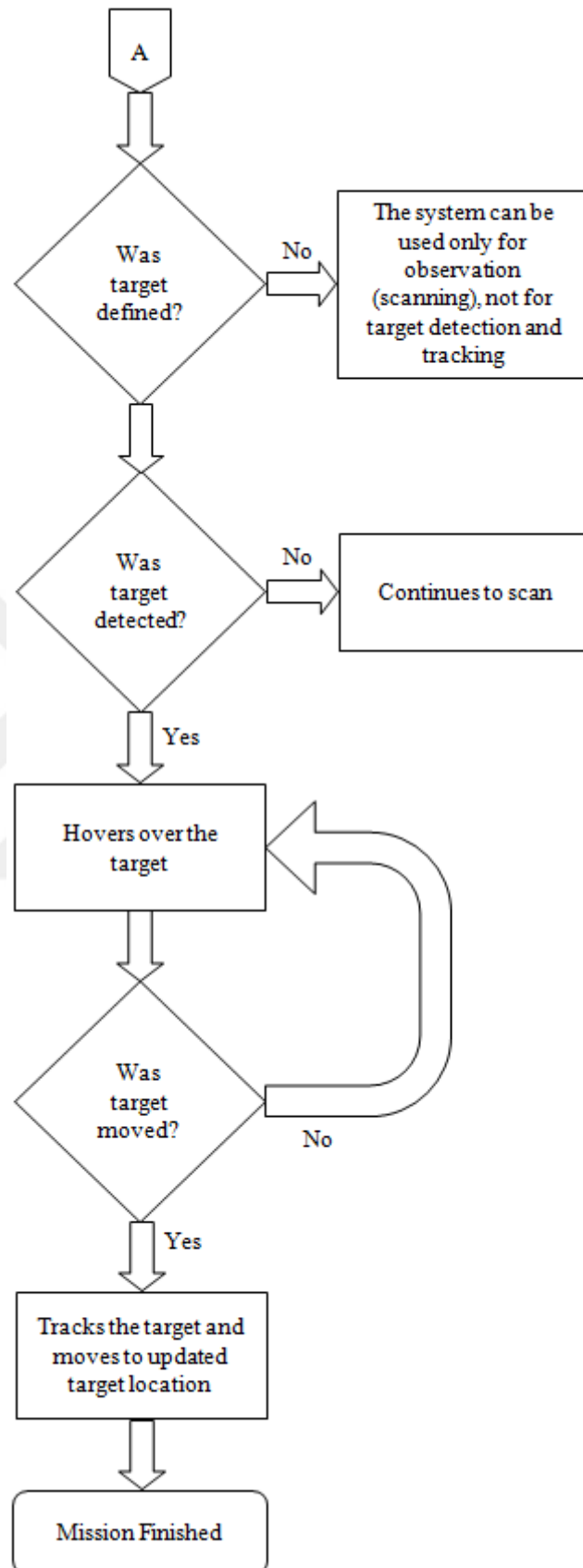


Figure 2.2 continues

### 2.1.6 Mini PC

A Raspberry pi 3 Model B+ is used for image processing and command producing stages. The computer has very small size, lightweight and low-cost even it has good performance. The photo of the mini PC is given in Figure 2.3.



Figure 2.3 The photo of the mini PC (Raspberry pi 3 Model B+) (Raspberry pi, 2019)

Specifications of mini PC:

- Processor: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- Memory: 1GB LPDDR2 SDRAM
- Connectivity:
  - 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, bluetooth 4.2, BLE
  - Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
  - 4 USB 2.0 ports
- Access: Extended 40-pin GPIO header
- Video&sound:
  - Full-size HDMI
  - CSI camera port for connecting a Raspberry Pi camera

- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port

### 2.1.7 Camera Module

A camera module is used to get images for image processing. This module is manufactured by the same manufacturer of mini PC and it is totally compatible with Raspberry pi. The photo of the camera is shown in Figure 2.4.

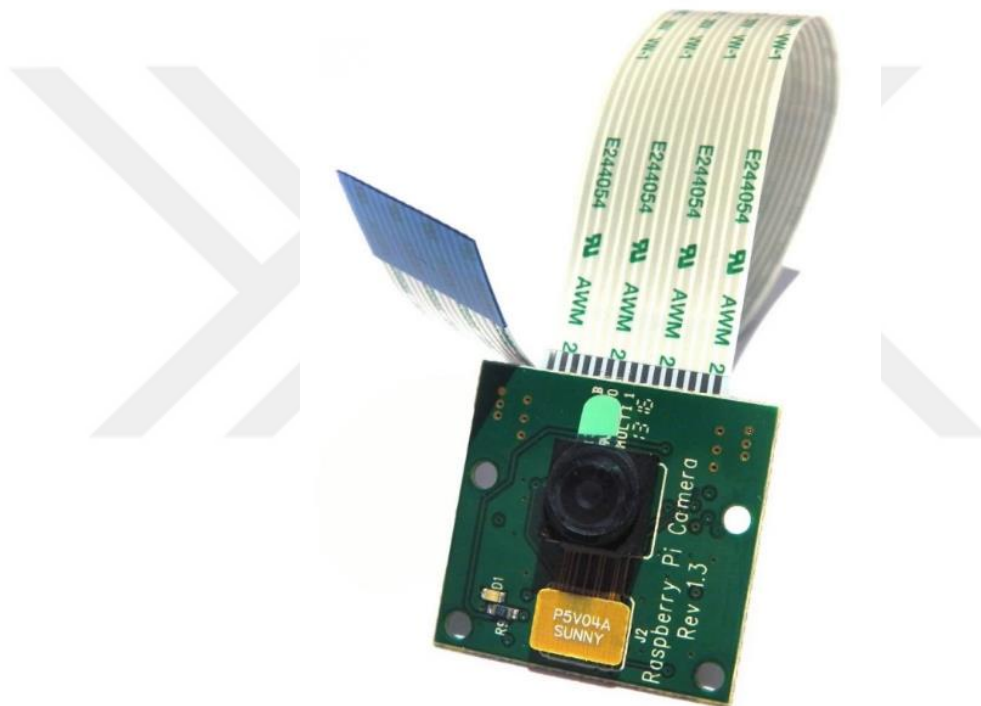


Figure 2.4 Camera board of mini PC (Pisupply, 2019)

Features of the camera module:

- 5MP Omnivision 5647 Camera Module
- Still Picture Resolution: 2592 x 1944
- Video: Supports 1080p @ 30fps, 720p @ 60fps and 640x480p 60/90 Recording
- 15-pin MIPI Camera Serial Interface - Plugs Directly into the Raspberry Pi Board

- Size: 20 x 25 x 9mm
- Weight 3g

## 2.2 UAV Control

An effective unmanned aerial vehicle was produced in terms of weight, flight time and cost. Dimensions of the UAV are 41 x 41 x 21 cm. and weight of the UAV is 1420 grams. This system can fly almost 40 minutes with 14.8 Volts (4S LiPo), 6000 mAh battery. The UAV system consists of many components such as:

- Flight controller (includes PID controller)
  - Inertial measurement unit (IMU)
  - Barometer
  - Power Distribution Unit (PDU)
- GPS
- Compass
- Brushless motor
- Electronic Speed Controller (ESC)
- Propellers
- Battery
- Power switch
- Telemetry transceiver
- Remote controller (R/C)
- Remote controller receiver

### 2.2.1 Components of the UAV

The system composed of electrical, electronics and mechanical components.

#### 2.2.1.1 Electrical Components

*2.2.1.1.1 Brushless Motors.* 2 pieces of CW (clockwise) and 2 pieces of CCW (counter clockwise) high speed Brushless DC (BLDC) motors were used to rotate propellers. A piece of motor is shown in Figure 2.5. The motor is out-runner type, so outside case rotates while inside part stays fixed. Out-runner type motor produces more torque than in-runner type motor so they are more suitable to be used for UAVs to rotate propellers (Dronetrest, 2019).

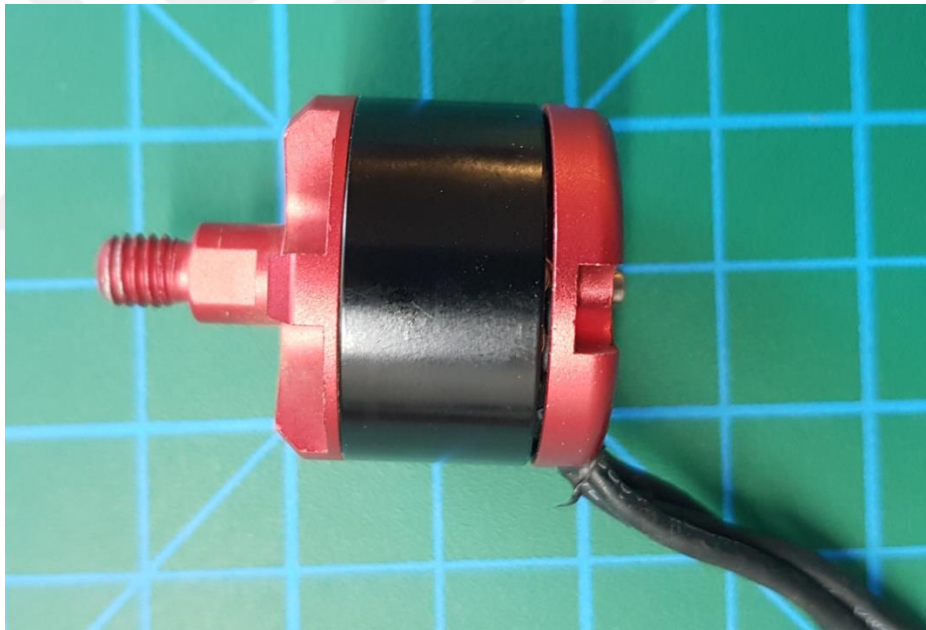


Figure 2.5 2212 920 Kv Brushless DC motor (Personal archive, 2019)

Motor type is 2212 920 Kv where 22 means diameter of stator in millimeters, 12 means height of stator in millimeters. Kv tells how the generated back-emf in the motor relates to motor speed. So, the motor gives one volt of back-emf at 920 rpm. This parameter can be used to estimate the motor speed (rpm) roughly at a certain voltage by multiplying Kv value with voltage level.

Features of 2212 920KV brushless motor:

- Dimensions: 28mm x 24mm
- Kv: 920
- Max Thrust: 1200 grams
- Weight: 56 grams
- Shaft Diameter: 4mm

*2.2.1.1.2 Battery.* Battery capacity is one of the most important parameters to determine flight duration. In this study 4 pieces of 3.7V 3000 mAh batteries were connected in series and this block was connected parallel with another identical block. Thus, the battery block supplies 6000 mAh at 14.8 volts. These values provide for the system 40 minutes of flight duration. Photo of the battery block can be seen in Figure 2.6.



Figure 2.6 Battery block of UAV (Personal archive, 2019)

### 2.2.1.2 Electronics Components

**2.2.1.2.1 Flight Controller.** Kakute F7 flight controller was used for UAV control. The flight controller module includes Inertial measurement unit (IMU), Barometer and Power distribution unit (PDU). IMU consists of gyroscope and accelerometer. It was used to determine motion and acceleration in three axes (x, y, z). The IMU provides that the UAV to respond to vehicle instabilities and maintain a safe flight (Bennett, Rickert, Crawford, Lamppin & Warman, 2017). Barometer is a pressure sensor and it was used in the system to calculate the altitude with the help of (by measuring) pressure. PDU provides the voltage level conversion by reducing 14.8V battery voltage level to 5V for input voltage of the flight controller. This voltage level was also used for peripheral devices. Ardupilot 3.7.0 developer version firmware was used in the system. The photos of the Kakute flight controller are given in Figure 2.7.

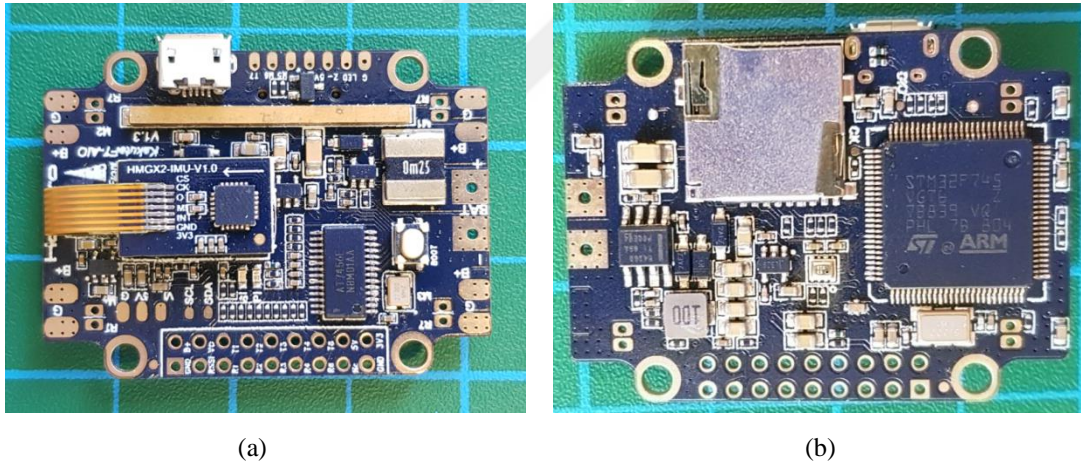


Figure 2.7 Kakute F7 flight controller (a) Top view (b) Bottom view (Personal archive, 2019)

2.2.1.2.2 *GPS and Compass.* GPS was used to determine the location and as a second source of altitude identifier. A Compass was used to find the direction of the UAV. GPS and compass were used as a dual module. Top and bottom views of the dual GPS and compass module are shown in Figure 2.8.

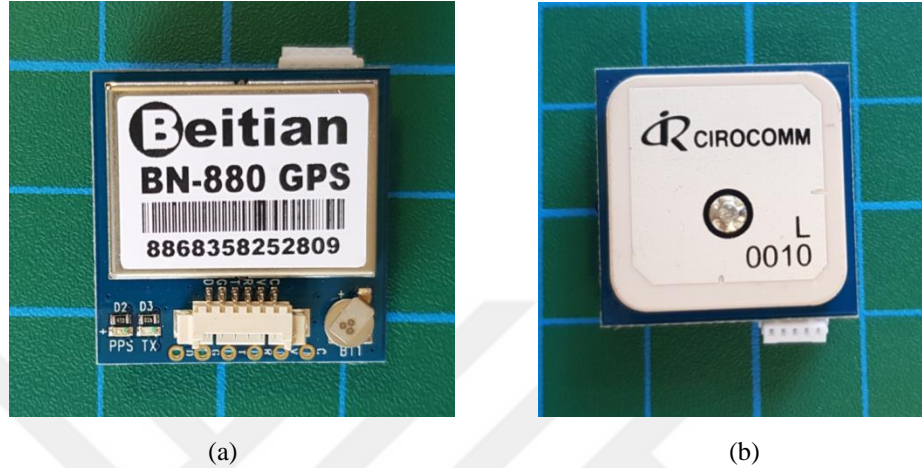


Figure 2.8 GPS and dual compass module (a) Top view (b) Bottom view (Personal archive, 2019)

Description:

- Model: BN-880
- Power Supply: DC Voltage 2.8V~6.0V, Typical: 3.3V or 5.0V
- Consumption: Capture 50mA@5V
- Receiving Format: GPS, GLONASS, Galileo, BeiDou, QZSS and SBAS
- Receiving Channel: 72 Searching Channel
- Level Positioning Precision: 2m At Open Wind
- Output Frequency: 1Hz-10Hz, Default 1Hz
- Speed Precision: 0.1 m/s (Without Aid)
- Acceleration Precision: 0.1 m/s (Without Aid)
- Dynamic Characteristics: Max Height: 18000m  
Max Speed: 5153m/s  
Max Acceleration: 4G
- UART Interface: UART Port:TXDA and RXDA
- Size: 28mm x 28mm x 10mm
- Weight: 10g

*2.2.1.2.3 Remote Controller.* A remote controller was used in development stage and it was always ready to control UAV manually if any necessary case occurred. Used remote controller during study can be seen in in Figure 2.9.



Figure 2.9 Six-channels remote controller (Engineerstoy, 2019)

Specifications:

- Channels: 6
- RF range: 2.408 - 2.475GHz
- Bandwidth: 500 KHz
- Bands: 135
- RF power: Less than 20 dBm
- Protocol: AFHDS 2A
- Modulation type: GFSK
- Low voltage alarm: Yes (lower than 4.2V)
- Power input: 6V DC 1.4AA\*4
- Weight: 392g
- Size: 174 x 89 x 190 mm

*2.2.1.2.4 Remote Controller Receiver.* The remote controller receiver gets the incoming data stream from remote controller and it transmits the commands to the flight controller via sBUS protocol. The receiver which is shown in Figure 2.10 is compatible with remote controller and it is sold in same package.



Figure 2.10 Remote controller receiver (Engineerstoy, 2019)

*2.2.1.2.5 Telemetry Transceiver.* Used set of telemetry module consists of a piece of micro radio telemetry module and a piece of ground radio telemetry module. Radio telemetry modules provide ground station computer for wireless communication with the UAV. The telemetry modules are shown in Figure 2.11.



Figure 2.11 Telemetry module transceivers (Holybro, 2019)

Features of telemetry radio module:

- 915 or 433 MHz
- 100 mW maximum output power (adjustable)
- -117 dBm receive sensitivity
- 2-way full-duplex communication through adaptive TDM
- UART interface
- Transparent serial link
- MAVLink protocol framing
- Frequency Hopping Spread Spectrum (FHSS)
- Error correction corrects up to 25% of bit errors
- Configurable through Mission Planner and APM Planner
- Supply voltage: 3.7-6 VDC (from USB or DF13 connector)
- Transmit current: 100 mA at 20 dBm
- Receive current: 25 mA
- Serial interface: 3.3 V UART
- Dimensions: 26mm x 19mm

*2.2.1.2.6 Battery Indicator.* A battery indicator was used to provide safer flight and ease of study. Because, the battery level is very important for the UAV to fly safely. The battery voltage level can also be observed from ground control station but to have a more accurate result, the battery voltage level was monitored with the aid of voltage indicator without removing the battery to measure. This feature is really important with respect to ease of use. The battery indicator is shown in Figure 2.12.

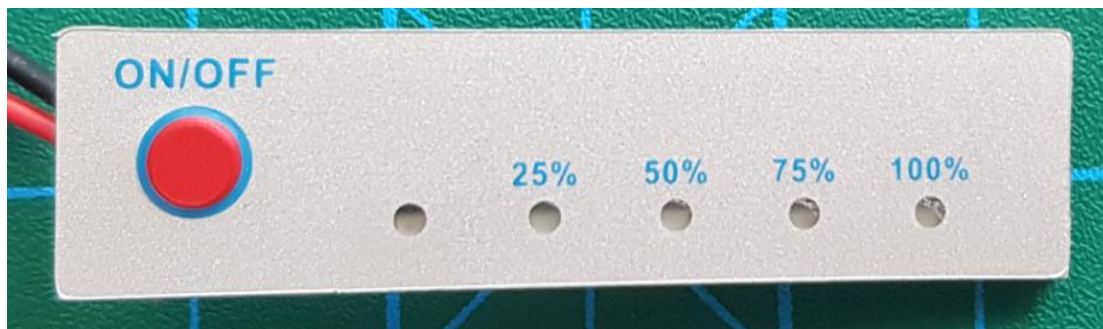


Figure 2.12 Battery voltage level indicator (Personal archive, 2019)

2.2.1.2.7 ESC (*Electronic Speed Controller*). Electronic speed controller was used to drive motors and control motor behaviours. The used ESC in the system is shown in Figure 2.13.

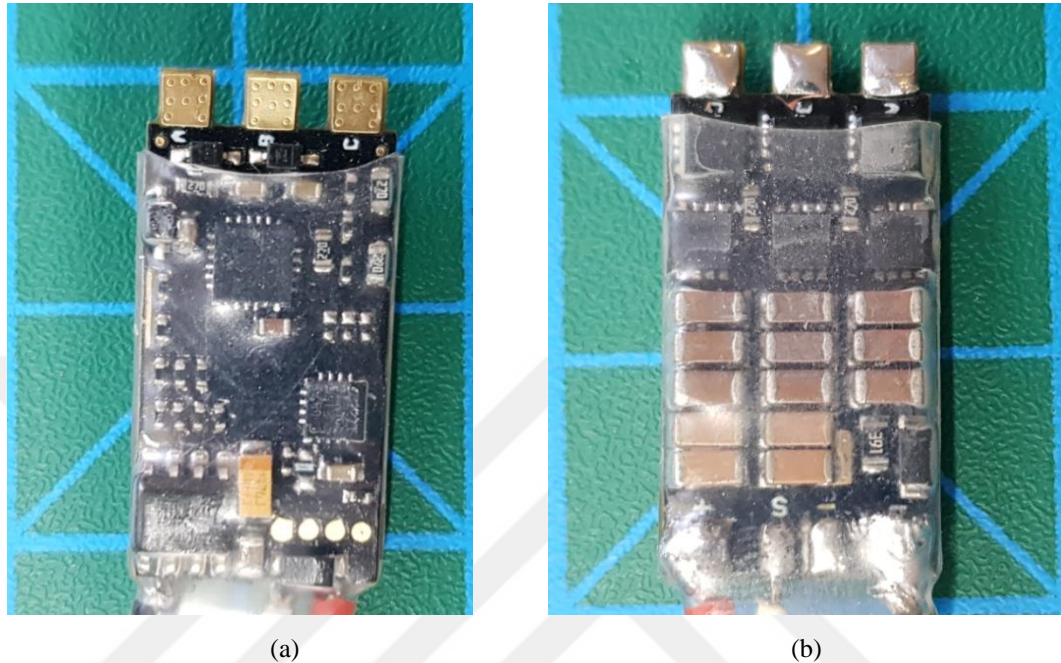


Figure 2.13 Electronic speed controller (ESC) (a) Top view (b) Bottom view (Personal archive, 2019)

Specifications:

- Continuous current: 25A
- Burst current: 35A (10S)
- Voltage range: 2-4S Lipo
- Weight: 5.9 g.
- Dimensions: 12.8 x 25.8 x 5 mm

*2.2.1.2.8 Power Switch.* A power switch was used to control supply voltage to conduct to electrical and electronics equipment or isolate from them. This module provides the On/Off switch to the system and the system is not energized as soon as the battery is inserted. The module also prevents possible damages. The power switch module circuit contains a pair of P-channel MOSFETs and it has a reverse voltage protection. The module is shown in Figure 2.14.

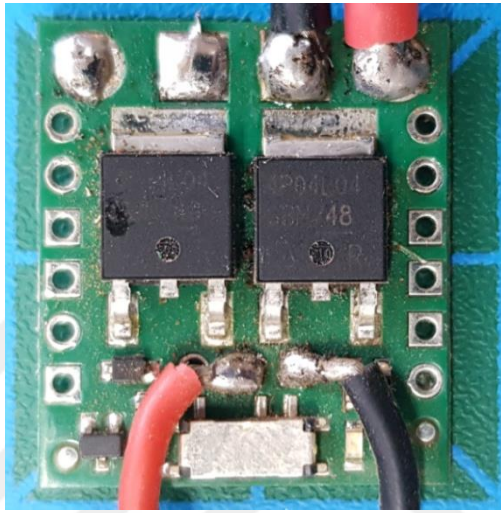


Figure 2.14 Power switch (Personal archive, 2019)

Features of Power switch module:

- Absolute max voltage: 40V
- Recommended operating voltage: 4.5-32V
- MOSFET combined on resistance (max): 13 m $\Omega$  @ 4.5 V, 8.6 m $\Omega$  @ 10 V
- Continuous current at 55°C: 6A
- Continuous current at 150°C: 16A
- Maximum current: 90A
- Current consumption in on state: ~65  $\mu$ A/V
- Size: 20 x 25.4 x 0.4mm
- Weight: 2.7gr

### 2.2.1.3 Mechanical Components

**2.2.1.3.1 Frame.** The UAV's body, joints and foot pads were made of plastic material, and the UAV's legs and arms were made of aluminum material. Plastic parts were produced from 3D printer. Some of the 3D printer drawings and photos of the frame are shown in Figure 2.15, Figure 2.16 and Figure 2.17.

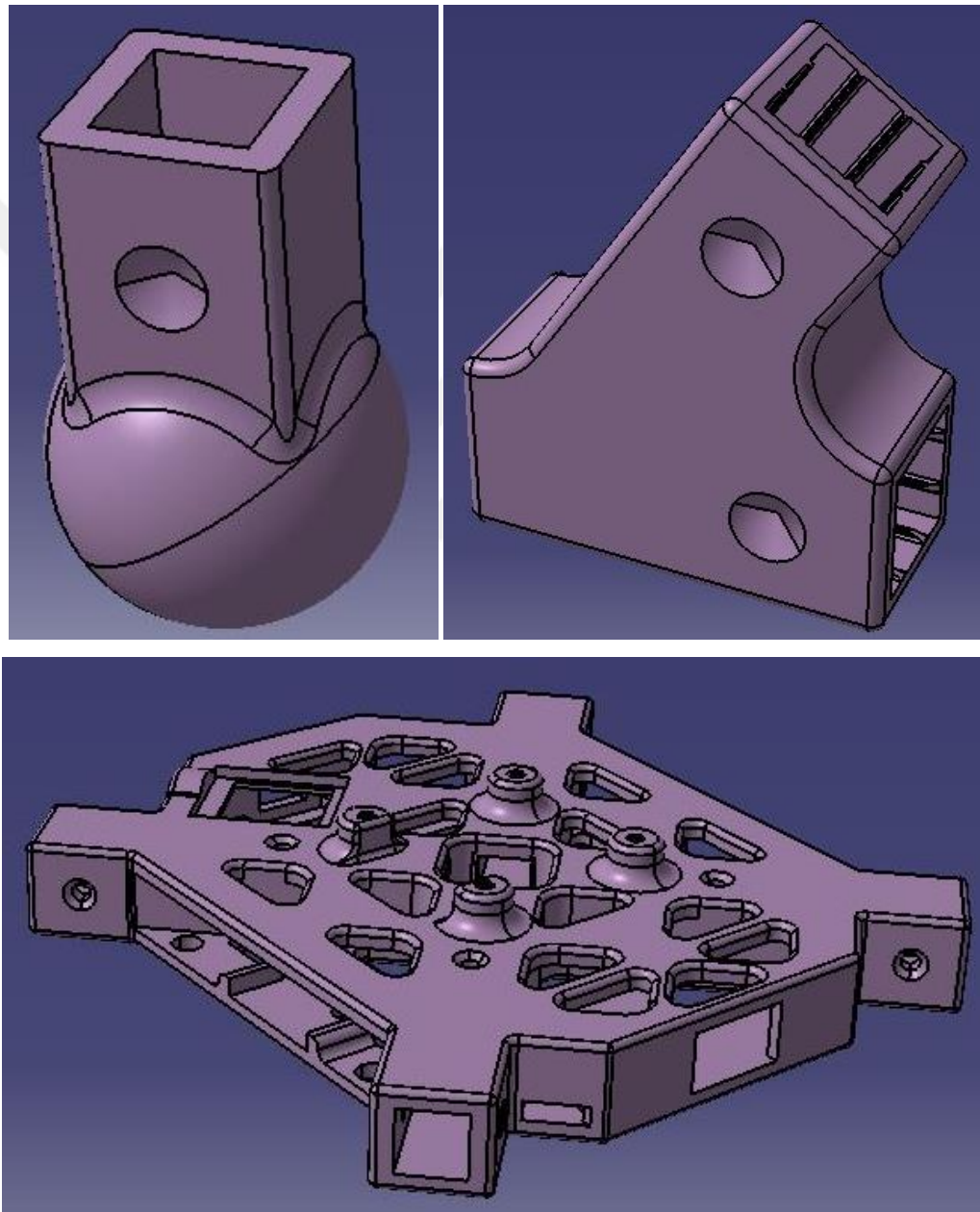


Figure 2.15 3D printer drawings of some plastic parts of the UAV

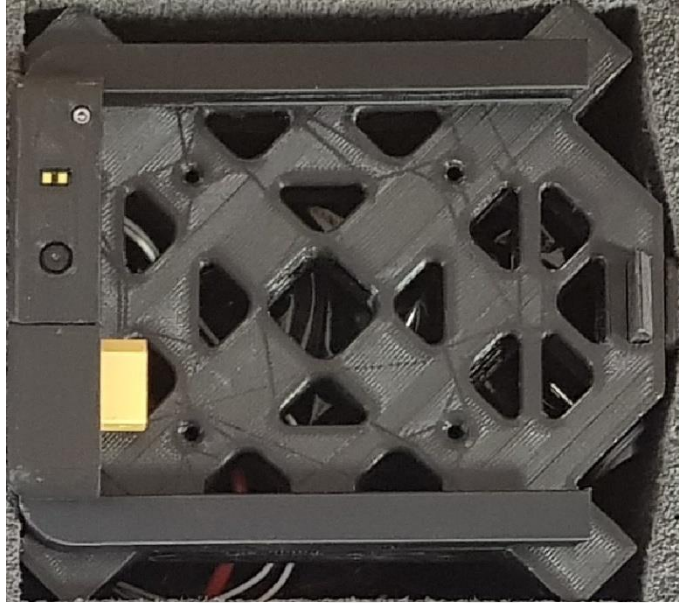


Figure 2.16 Body photo of the UAV (Personal archive, 2019)



Figure 2.17 Aluminium arm of the UAV (Personal archive, 2019)

*2.2.1.3.2 Propellers.* 2 pieces of CW (clockwise) and 2 pieces of CCW (counterclockwise) propellers were used to generate air flow that provides lift force. Black nuts on the propellers refer to CW spinning direction and silver nuts refer to CCW spinning direction. The propellers are shown in Figure 2.18.

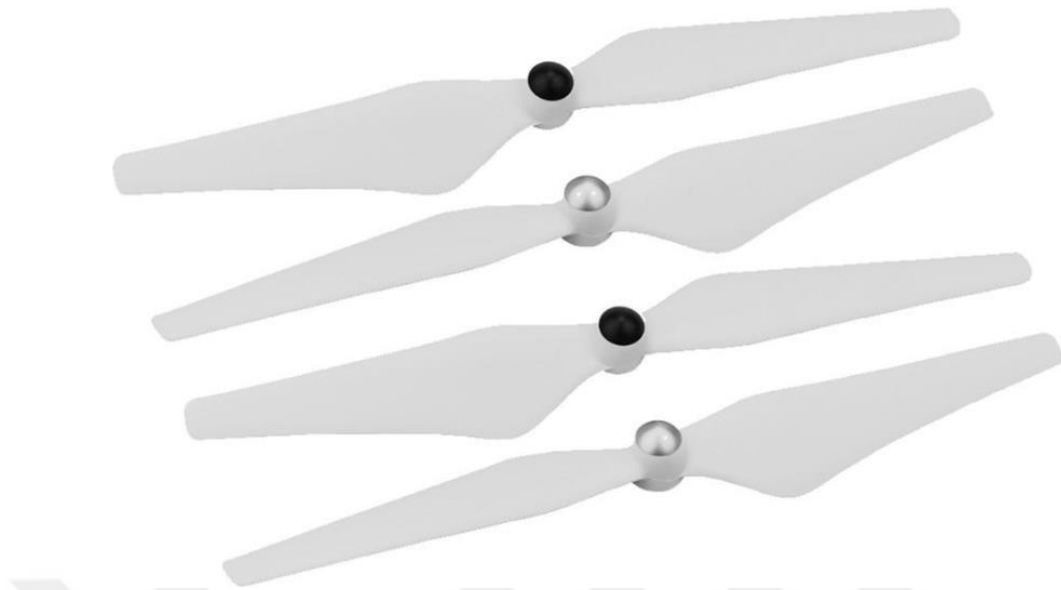


Figure 2.18 9450 Plastic propellers (Dronesxpress, 2019)

Propeller type is mentioned as 9450 and it means that the propeller has 9.4 inches (23.9 cm) length (diameter), 5 inches (12.7 cm) screw pitch. Propellers' material is durable plastic, thus propellers are lightweight and durable. They are self-tightened and this design feature provides secure connection between motor and propeller.

### ***2.2.2 PID Controller***

The proportional-integral-derivative (PID) controller is the most commonly used control loop feedback mechanism. The system provides to minimize the error of the controlled system dynamically. The PID controller consists of three control parameters. These are proportional, integral and derivative, respectively (Barták & Vykovský, 2015).

### ***2.2.3 Ground Control Station (GCS)***

Mission Planner was used as a ground control station. All telemetry data can be observed with the help of this program (McClure, 2019). A screenshot of Mission Planner can be seen in Figure 2.19.



Figure 2.19 Mission planner GCS (Ground control station)

## 2.3 Design Steps

### 2.3.1 Definition of Target

A black car was defined as a target in this study. Here, two parameters are considered such as dimensions and color.

The car is defined as a rectangular from the top view. The dimensions of the car were given in Table 2.1. The aspect ratio of the car identified as a target was given as an input parameter to the system and its value is 2.56. It is useful to define the tolerance by  $\pm 10\%$  to avoid missing the target due to measurement error.

Table 2.1 Simulation target (car) dimensions

	Actual Size	Model Size (1:24 scale)	Aspect Ratio
<b>Length</b>	5052 mm.	210 mm.	2.56
<b>Width</b>	1968 mm.	82 mm.	

Necessary code snippets for dimensional detection can be seen in appendix 2.

H, S, V parameters were used to describe the color. Three ways can be used to determine these parameters. These are using RGB-to-HSV converter, using OpenCV command (a kind of converter) and empirical analysis. The best method is combining two ways to improve accuracy. Using both of converter and empirical analysis methods together was preferred. To use OpenCV commands for obtaining HSV values, RGB values of the colors must be known. The code snippets to obtain the HSV values of desired color can be seen in appendix 3.

Theoretically, the RGB and HSV values for black color are both 0. The system requires pure deep black color with these values but if the saturation of black color, sunlight, etc. are taken into account, doing empirical analysis is necessary to obtain more accurate result. So, empirical analysis was performed and an ideal range was found as upper and lower limit according to analysis result. The obtained upper and lower limits of HSV can be seen below:

```
hsv_lover = (0, 0, 0)
hsv_upper = (255, 255, 40)
```

### ***2.3.2 Determination of the Field to be Scanned***

The area where the target wanted to be searched in, can be entered in the system as a coordinate, or it can be given as a distance from the current location. In this study, it was requested to detect and follow the target by scanning the area at a certain distance from the located area (current location). The angle of view of the camera is  $53.50 \pm 0.13$  horizontally and  $41.41 \pm 0.11$  degrees vertically. The field of view of the camera is related to altitude and it can be calculated from the angle of view of the camera by using cosinus theorem seen in Eq. 2.1 and Eq. 2.2 in the light of Figure 2.20 and Figure 2.21.

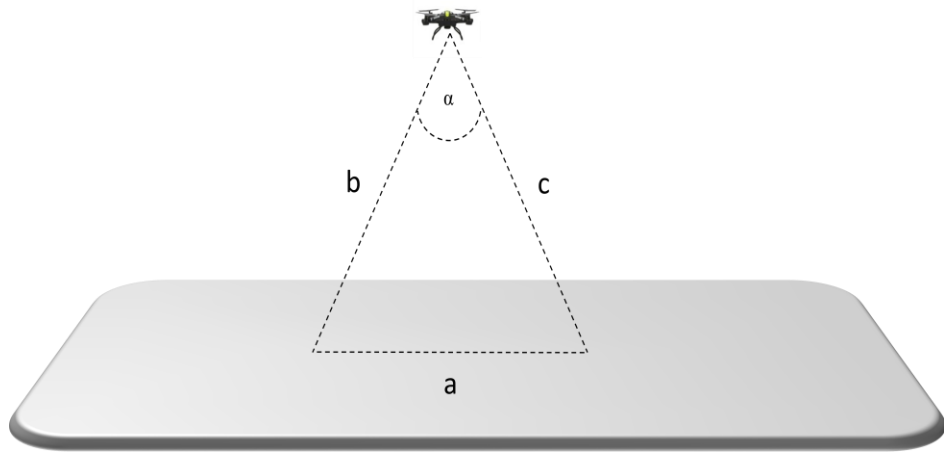


Figure 2.20 The illustration of the field of view

$$a^2 = b^2 + c^2 - 2bc\cos\alpha \quad (2.1)$$

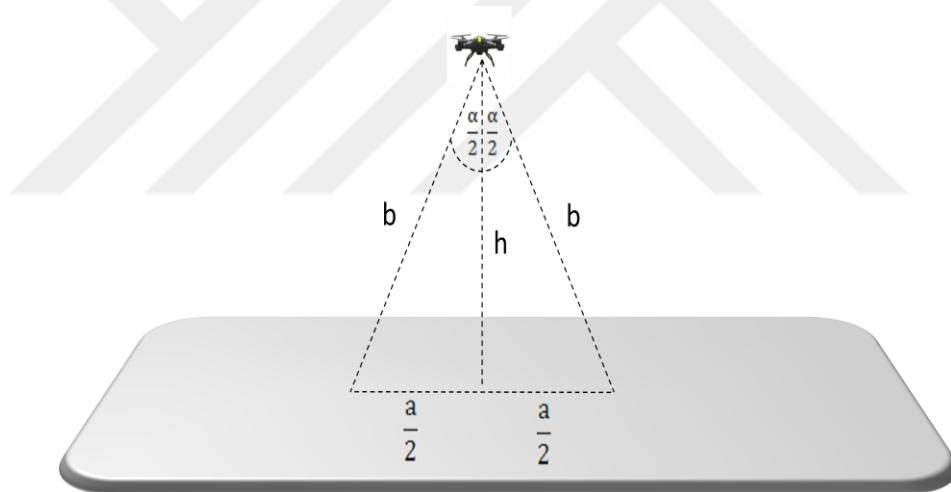


Figure 2.21 The illustration of the field of view of the camera in terms of altitude (h)

$$b = \frac{h}{\cos(\frac{\alpha}{2})} \quad (2.2)$$

Illustrations of the viewing angle and the field of view of the camera can be seen in Figure 2.22 and in Figure 2.23, respectively.

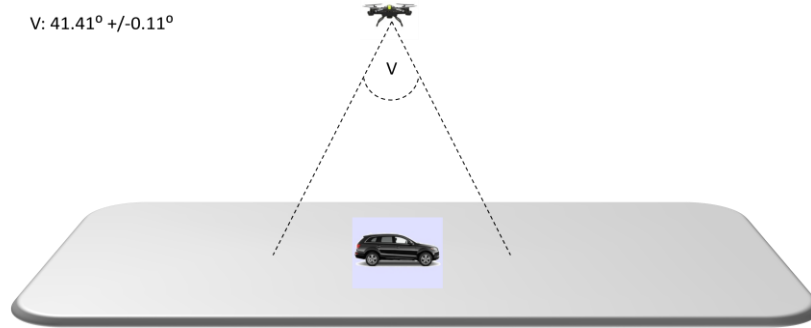


Figure 2.22 Angle of view of the camera in vertical axis

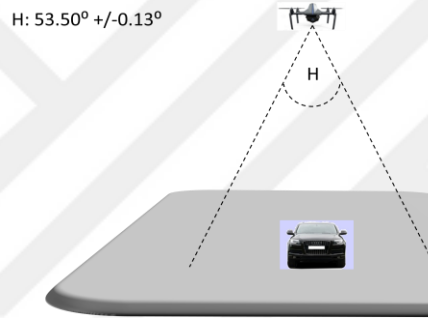


Figure 2.23 Angle of view of the camera in horizontal axis

For this study, 480 meters horizontal and 720 meters vertical area were identified as searching field. These dimensions are equal to 345,600 m<sup>2</sup> area. Since the mission was simulated with the model car, horizontal and vertical dimensions were divided by scale value which is 24. Actual and simulation values can be seen in Table 2.2.

Table 2.2 Actual and simulation values of scanning field

	<b>Simulation (1:24 scale)</b>	<b>Actual</b>
<b>Horizontal</b>	20 m.	480 m.
<b>Vertical</b>	30 m.	720 m.
<b>Altitude</b>	5 m.	120 m.
<b>Area</b>	600 m <sup>2</sup>	345,600 m <sup>2</sup>



### 2.3.3 Detection of the Target

Once the UAV has reached the identified area, it begins scanning the related field to find the target. When the specified target has been detected, the UAV attempts to remain stationary over the target by sending command to UAV to stay at current location. Three performed detection examples are given below in Figure 2.25, Figure 2.26 and Figure 2.27.



Figure 2.25 Object (car) detection in center position (a) Real frame of target (b) Masked image of target



Figure 2.26 Object (car) detection in left-forward diagonal position (a) Real frame of target (b) Masked image of target

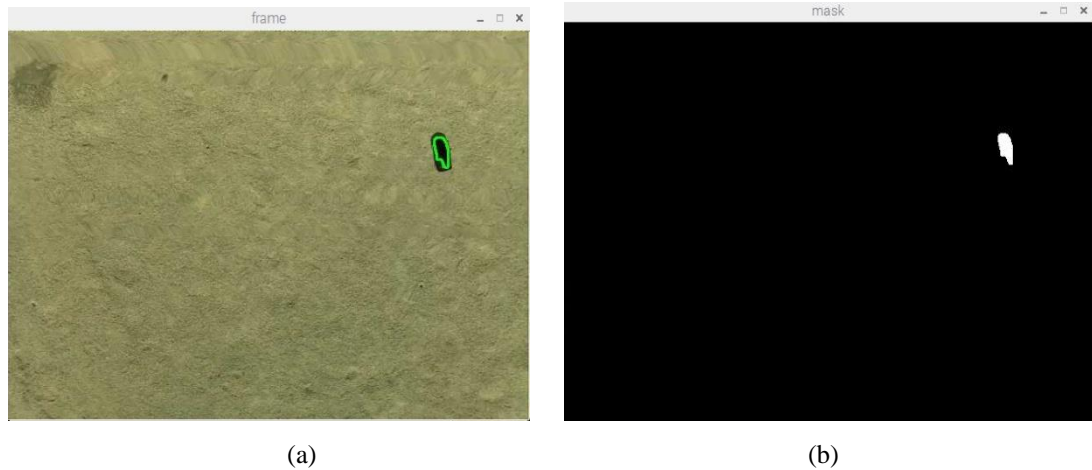


Figure 2.27 Object (car) detection in right-forward diagonal position (a) Real frame of target (b) Masked image of target

#### 2.3.4 Detection of the Moving Direction if the Target Moves

Field of view of the camera was calculated in pixel unit and boundary scan was implemented with respect to these pixel data. The resolution of the camera was chosen as 640 x 480 pixels. It means that horizontal line of field of view of the camera is divided by the number of horizontal pixels (640) to calculate the size of 1 pixel on horizontal axis. Likewise, vertical line of field of view of the camera is divided by the number of vertical pixels (480) to calculate the size of 1 pixel on vertical axis. Table 2.3 illustrates this conversion for the altitude of 5 meters. Horizontal and vertical field of views are calculated from Eq. 2.1 and Eq. 2.2. The illustration of the conversion of the field of view to pixel unit is given in Figure 2.28.

Table 2.3 Unit pixel size

	Horizontal	Vertical
<b>Simulation (1:24 scale)</b>	5 m.	3.9 m.
<b>Actual</b>	120 m.	93.6 m.
<b>Pixel number</b>	640	480
<b>Unit pixel size (For Simulation)</b>	0.78125 cm	0.8125 cm
<b>Unit pixel size (For Actual)</b>	18.75 cm	19.5 cm

The target is wanted to be kept at the center of the field of view. Therefore, a center frame was created with the boundaries of between 200 and 440 pixels horizontally and between 150 and 330 pixels vertically. Figure 2.29 illustrates the desired target position.

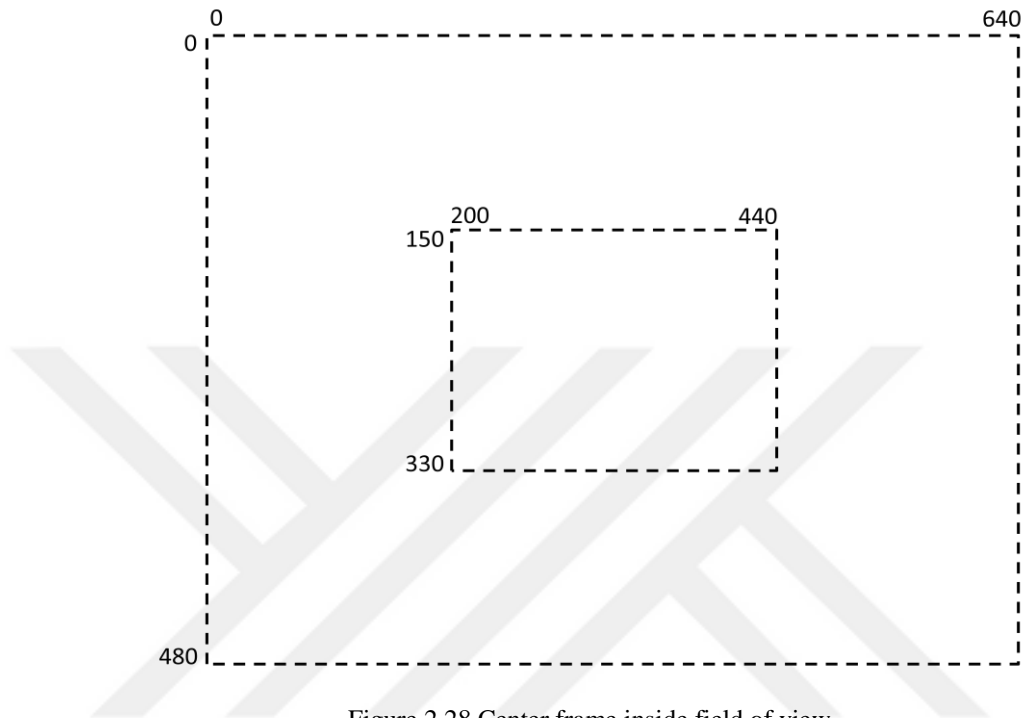


Figure 2.28 Center frame inside field of view

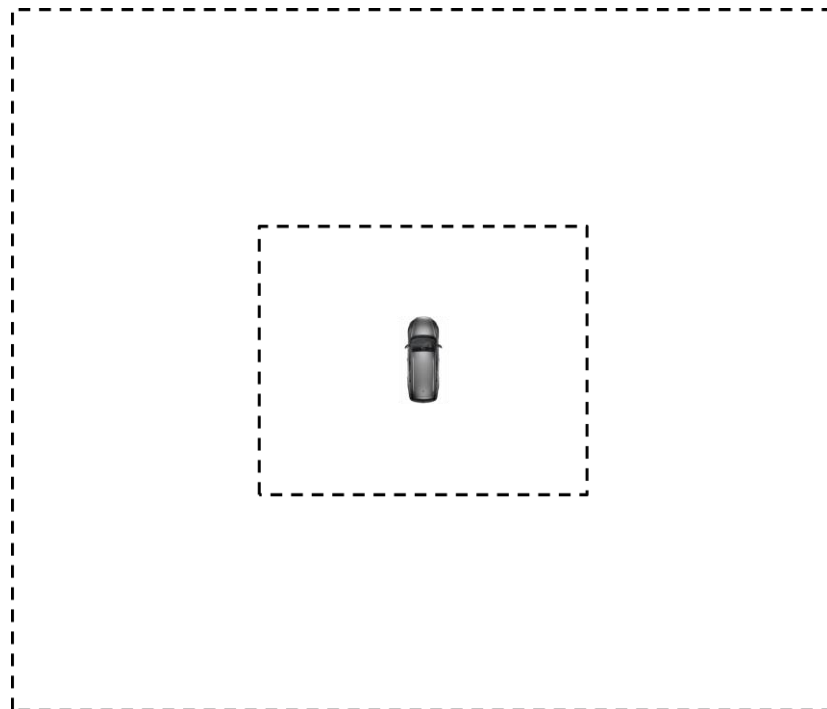


Figure 2.29 The car stays at center position

If the target moves after it has been detected, the system determines the direction of exit from the boundaries of the specified frame. These boundaries and directions can be seen in Figure 2.30.

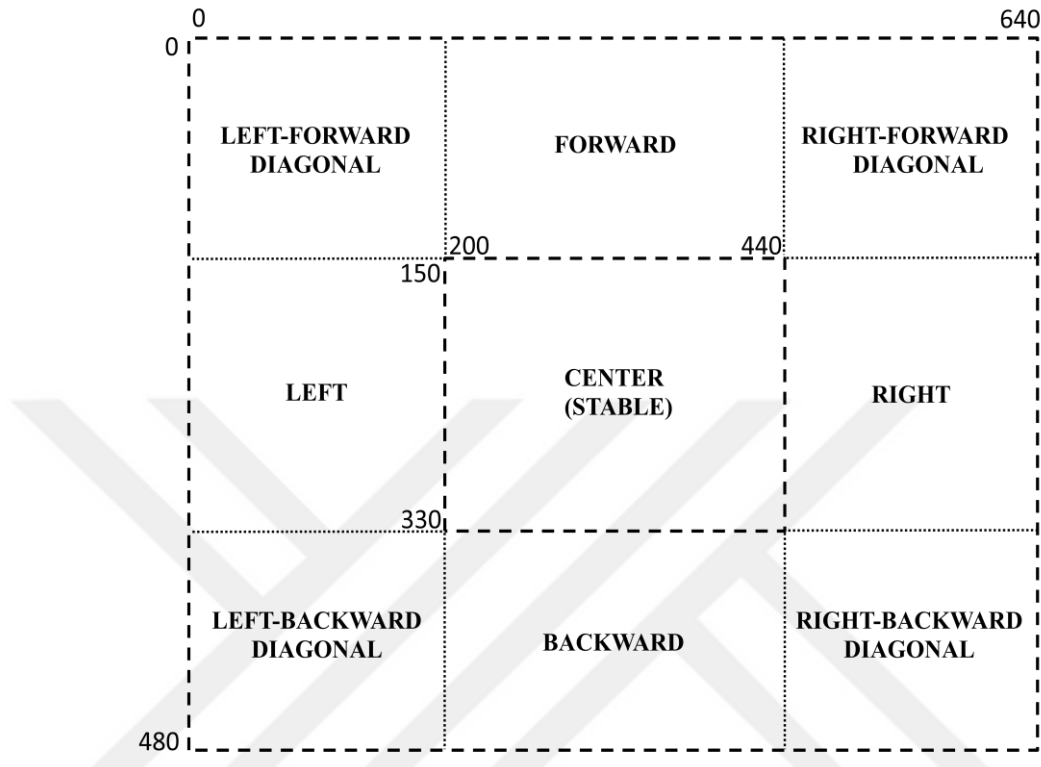


Figure 2.30 Pixel boundaries of directions

The comparison of the current location and the boundary frame is performed with the below comparison code lines. The system determines the direction to go according to this comparison. Here, "x" represents horizontal axis and "y" represents vertical axis in pixel unit.

if  $x < 200$ :

If this case is true, the system decides to go to left direction.

if  $x > 440$ :

If this case is true, the system decides to go to right direction.

Illustration of the above two behaviours can be seen in Figure 2.31.

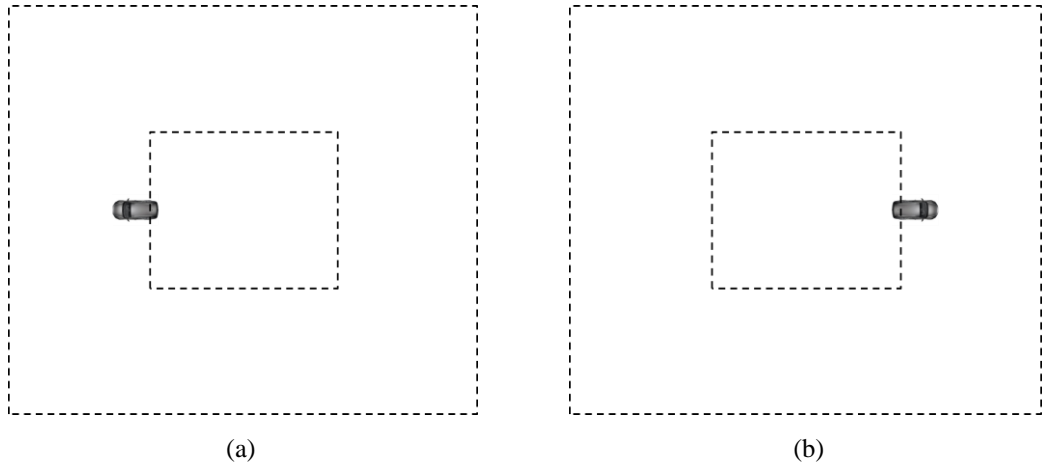


Figure 2.31 Leaving the center frame in the direction of the horizontal axis (a) Through left direction (b) Through right direction

if  $y < 150$ :

If this case is true, the system decides to go to forward direction.

if  $y < 330$ :

If this case is true, the system decides to go to backward direction.

Illustration of the above two behaviours can be seen in Figure 2.32.

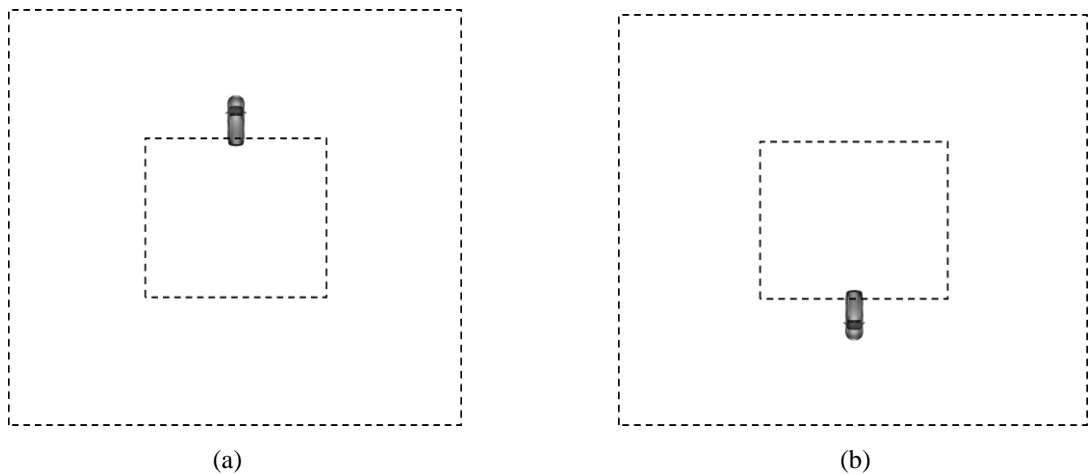


Figure 2.32 Leaving the center frame in the direction of the vertical axis (a) Through forward direction (b) Through backward direction

if ( $x < 200$  and  $y < 150$ ):

If this case is true, the system decides to go to left-forward diagonal direction.

if ( $x < 200$  and  $y > 330$ ):

If this case is true, the system decides to go to left-backward diagonal direction.

Illustration of the above two behaviours can be seen in Figure 2.33.

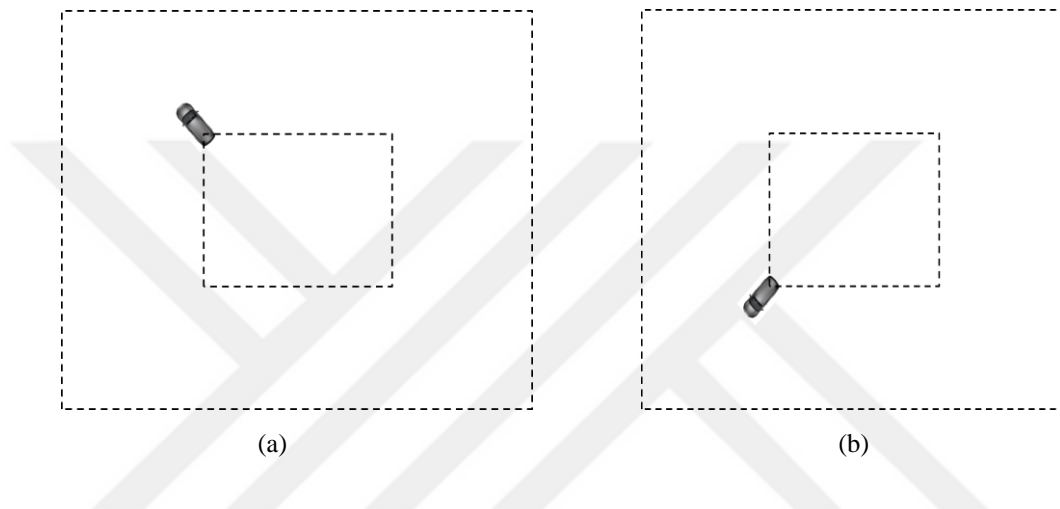


Figure 2.33 Leaving the center frame in the direction of the diagonal axis (a) Through left-forward diagonal direction (b) Through left-backward diagonal direction

if ( $x > 440$  and  $y < 150$ ):

If this case is true, the system decides to go to right-forward diagonal direction.

if ( $x > 440$  and  $y > 330$ ):

If this case is true, the system decides to go to right-backward diagonal direction.

Illustration of the above two behaviours can be seen in Figure 2.34.

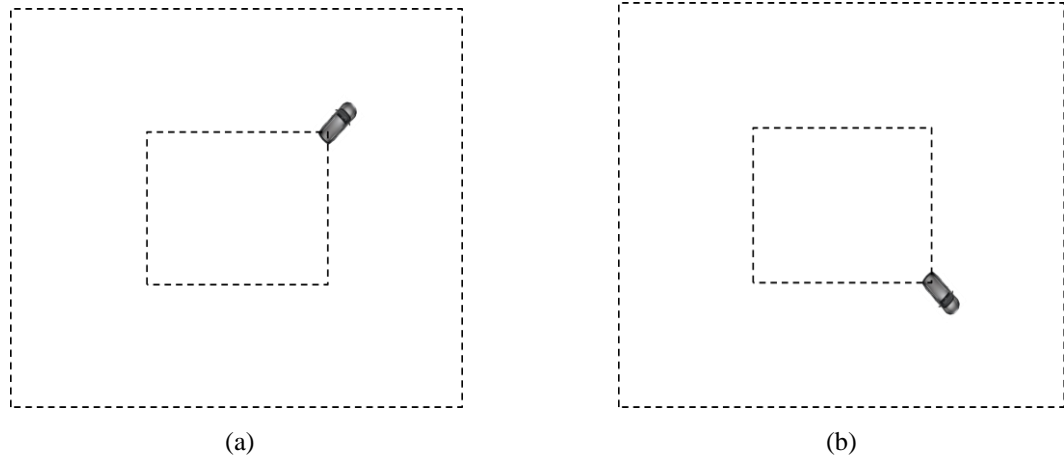


Figure 2.34 Leaving the center frame in the direction of the diagonal axis (a) Through right-forward diagonal direction (b) Through right-backward diagonal direction

In other cases, the system will remain stable.

### ***2.3.5 Producing and Sending the Necessary Commands to UAV System by Mini PC for Tracking the Target***

The UAV attempts to keep the target at the center of the camera's field of view within the specified limits. Mini PC sends commands to the flight controller using the MavLink protocol to make the UAV move in the direction the target leaves the specified frame. The flight controller moves with respect to latitude angle and longitude angle. Therefore, some calculations are needed to obtain the required latitude and longitude angle to track the target. Distance difference from the current location should be converted from metric unit to the latitude and longitude angle. In order to make these calculations, it is necessary to make use of geographical information. Figure 2.35, Figure 2.36 and Figure 2.37 are drawn to illustrate the relation of geographical shapes and geometry in order to help this calculation.

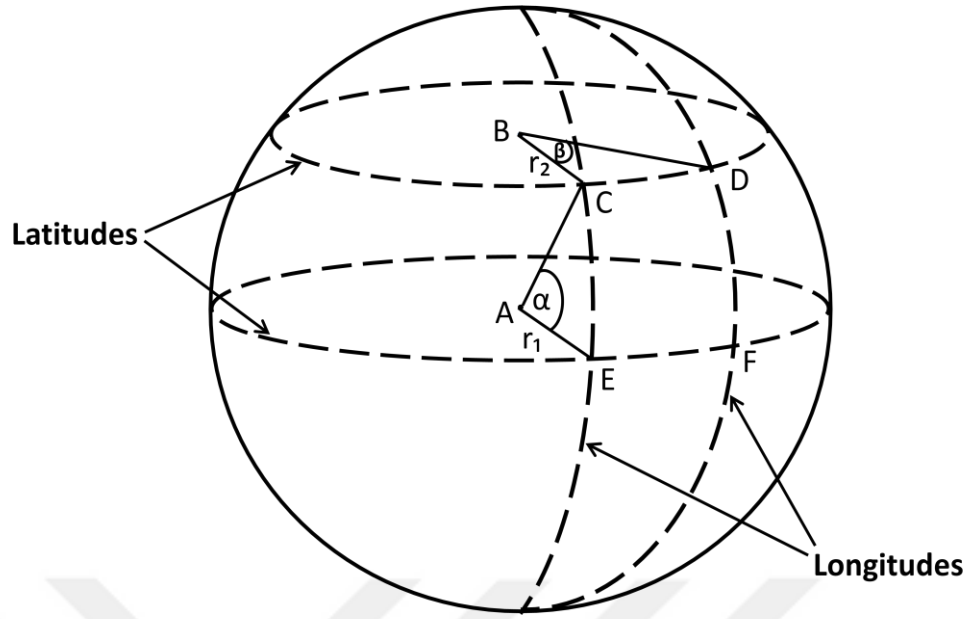


Figure 2.35 View of the world in terms of latitude and longitude

$\alpha$  = Latitude angle  
 $\beta$  = Longitude angle  
A, B = Center points

The latitude angle indicates the difference between two latitudes over the same longitude and the longitude angle indicates the difference between two longitudes over the same latitude. Moving in any direction from starting point to the end point gives the arc length for related direction. The angles mentioned above are used to calculate the arc lengths, and vice versa.

On the other hand, flight controller firmware uses the angles in degree unit while the python calculates in radians. Thus, all angle values are converted to radians and the new latitude and longitude angle values are calculated in radian unit. After obtaining the new angle values, these values are converted back to degree unit to communicate with flight controller correctly. Eq. 2.3 is used for the conversion between radian and degree.

$$\text{Radian} = \text{Degree} \frac{\pi}{180} \quad (2.3)$$

Movement on the vertical axis (North/South directions) causes a change in the latitude angle and this angle can be found from Eq. 2.10. "C" and "n" notations were used to represent the circumference of the circle and the angle across the arc, respectively in the auxiliary equations between Eq. 2.4 and Eq. 2.8 (Case, 2011).

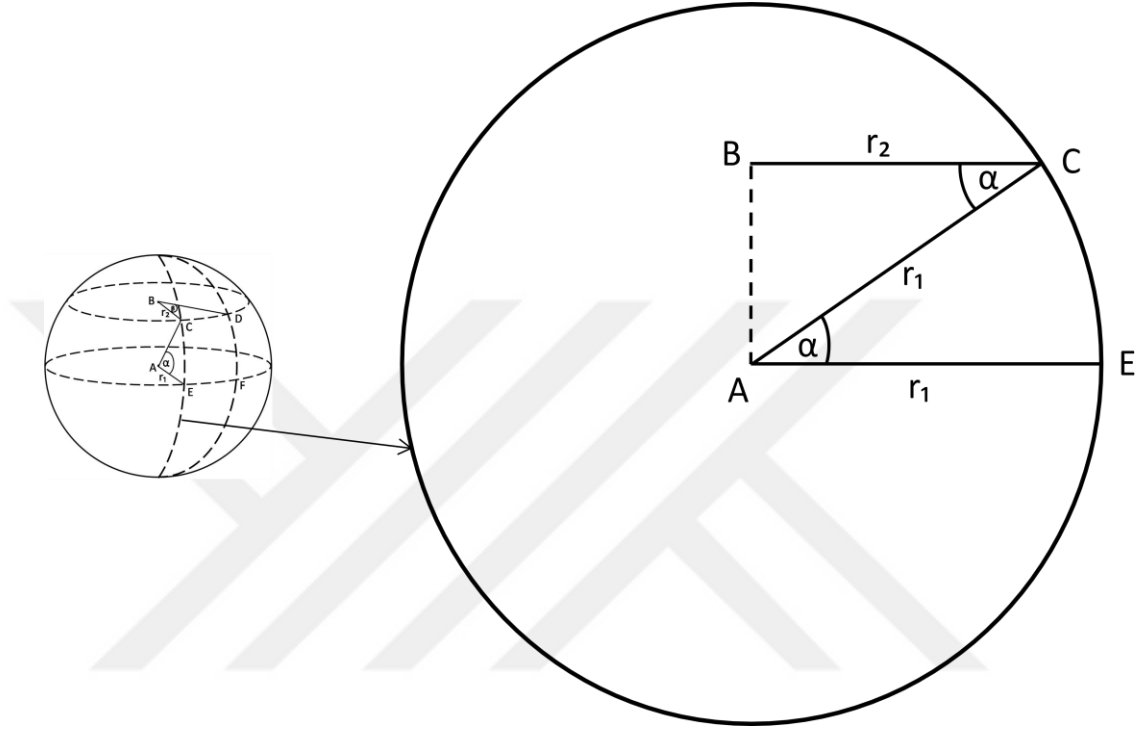


Figure 2.36 Illustration of radii according to latitude angle

$$C = 2\pi r \quad (2.4)$$

$$\text{Arc} = 2\pi r \frac{n}{360} \quad (2.5)$$

$$\text{Arc} = \pi r \frac{n}{180} \quad (\text{in degrees}) \quad (2.6)$$

From Eq. 2.3,  $\text{Arc} = rn \quad (\text{in radians}) \quad (2.7)$

$$n = \frac{\text{Arc}}{r} \quad (2.8)$$

$$\text{Arc } |EC| = r_1 \alpha \quad (2.9)$$

$$\alpha = \frac{\text{Arc}|EC|}{r_1} \quad (2.10)$$

Moving along the horizontal axis (West/East directions) causes a change in the longitude angle. To calculate the new value of the longitude angle, it is important that in which latitude the movement took place. Moving same distance over the latitude gives different longitude angles according to the latitude angle. Because the radius varies with the latitudes and the radius affects the calculation of the arc length. The longer the radius, the longer the arc length. This difference in radius can be obviously seen in Figure 2.36. To find the position difference over the latitude, the radius should be calculated according to the latitude angle. Eq. 2.11 is written in the light of Figure 2.36 and this equation can be used for radius calculation. On the other hand, the longitude angle can be found from Eq. 2.14 with the help of Eq. 2.11.

$$r_2 = r_1 \cos \alpha \quad (2.11)$$

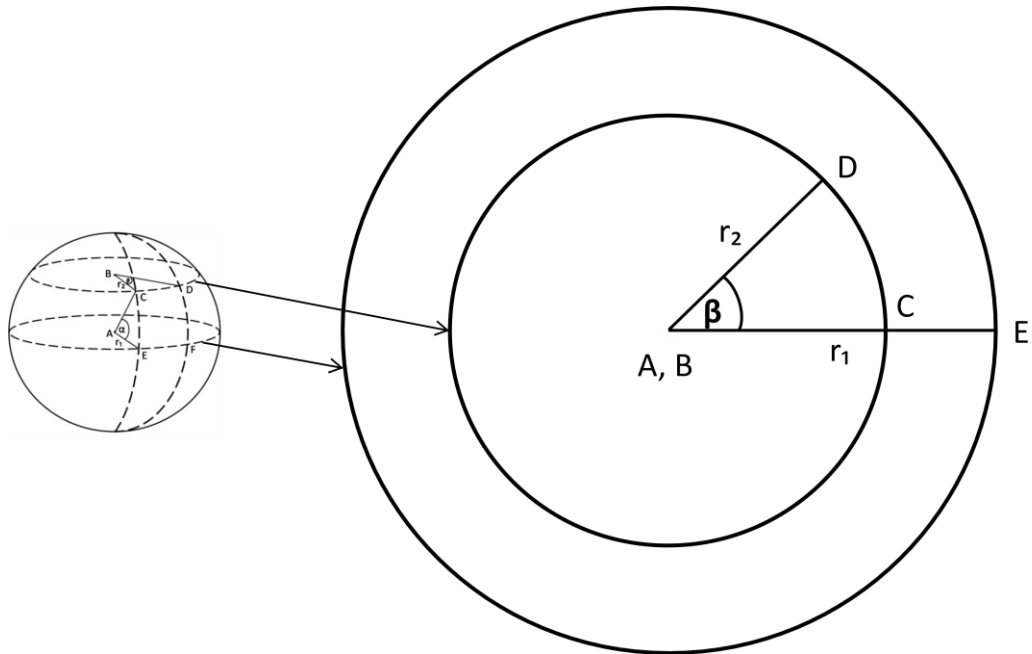


Figure 2.37 Illustration of arcs in terms of radii

$$\text{Arc } |CD| = \beta r_2 \quad (\text{in radians}) \quad (2.12)$$

$$\text{From Eq. 2.11,} \quad \text{Arc } |CD| = \beta r_1 \cos \alpha \quad (2.13)$$

$$\beta = \frac{\text{Arc } |CD|}{r_1 \cos \alpha} \quad (2.14)$$

Here,  $r_1 = r_{\text{earth}} = 6,378,137$  meters.

The code snippets were written according to these equations. The code snippets can be seen from appendix 5.

### ***2.3.6 Moving to Updated Target Location via UAV and Keeping Tracking***

The UAV that moves with the incoming tracking command continues to follow the target at the specified height after the target was centered again. In case of any possible movement of the target out of the frame, the above-mentioned steps starts from the relevant place and continues. A photo of the autonomous UAV system while tracking the target can be seen from Figure 2.38. Mini PC command output screen of an example mission is shown in Figure 2.39 and the routing of the UAV during mission is shown in Figure 2.40.



Figure 2.38 A photo of the UAV during the tracking operation (Personal archive, 2019)

20 m. x 30 m. field scanning commands:

```
(right)
(backward)
(left)
(backward)
(right)
(backward)
(left)
(backward)
(right)
(backward)
(left)
(backward)
(right)
```

Target detected!

Tracking commands:

```
(backward)
(stable)
(stable)
(stable)
(stable)
(stable)
(stable)
(backward)
(stable)
(stable)
(stable)
(left)
(right)
(stable)
(stable)
(stable)
(stable)
(stable)
(stable)
(stable)
(left)
(stable)
(stable)
(stable)
(left)
(left)
(backward)
(stable)
(left)
(backward)
(stable)
(stable)
(left)
(backward)
(stable)
(stable)
(stable)
(right)
(stable)
(stable)
(right-forward diagonal)
(stable)
(forward)
(stable)
(left-forward diagonal)
```

Figure 2.39 Mini PC command output during searching and tracking mission

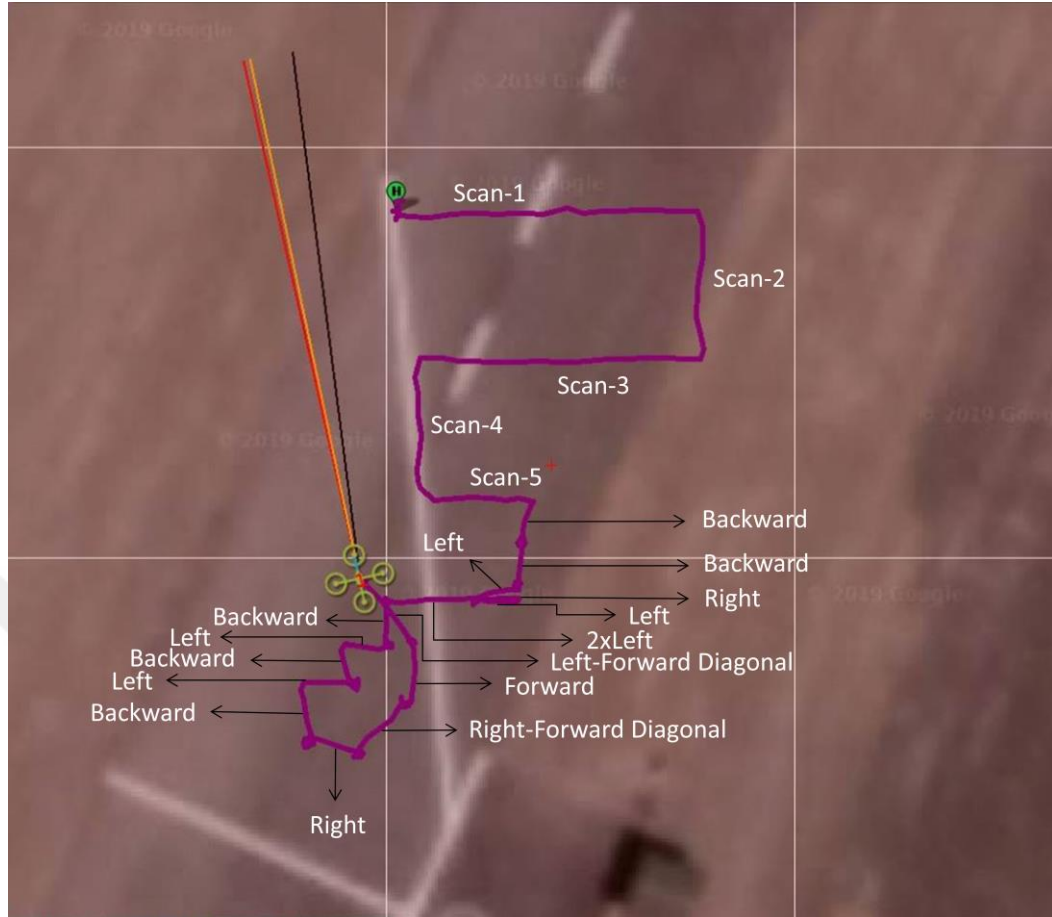


Figure 2.40 Routing output of mission planner during searching and tracking mission

The "H" letter represents "Home" and it is starting point of the mission. The UAV starts scanning from the Home position and scans the related field until the target is detected or entire scanning area is scanned. The UAV proceeds according to scanning commands seen in Figure 2.39. It starts scanning from Scan-1 path and continues until the scanning of Scan-5 path to find the target. The target is detected when the system on the way of Scan-5. After detection, UAV hovers over the target and these steps can be observed from "stable" commands. When the system detected any movement of the target to out of the center frame, it tracks the target according to tracking commands such as "Left", "Backward" or "Right-Forward Diagonal". All received tracking commands during mission can be seen from Figure 2.39. When the mission is completed, end-point of the mission is shown with quadcopter UAV symbol. All behaviours of the system are noted in Figure 2.40 and illustration of the mission can be seen from there.

The maximum speed of the UAV is about 50 km/h, so the designed UAV system can not track any target which moves faster than 50 km/h.

## 2.4 Cost Analysis

The whole system consists of many components and the total cost of the system is \$465. The detailed cost analysis is written in Table 2.4.

Table 2.4 Detailed cost analysis of the autonomous UAV system

<b>Component</b>	<b>Cost (\$)</b>
Flight Controller	53
GPS + Compass	22
Motors (4 pieces)	47
ESCs (4 pieces)	49
Battery + Charger	96
Power Switch	5
Propellers (4 pieces)	17
Telemetry Transceiver	33
Remote Controller (R/C) (inc. Receiver)	53
Mechanical Parts	24
Mini PC	41
Camera	25
<b>Total</b>	<b>465</b>

## **CHAPTER THREE**

### **CONCLUSION**

Usage of the UAVs is increasing day by day and UAVs are gaining in popularity in many sectors, even in daily life. On the other hand, a new trend is started in technology called as artificial intelligence recently and some works are done autonomously with the help of this technique. In many cases, doing works autonomously is easier, more accurate and safer.

In this thesis, a whole system is designed which successfully performs autonomous field scanning, autonomous object detection and autonomous object tracking missions. The study was carried out in two main stages and six sub-groups as described in methodology section.

The product produced in the light of this study can be used in many areas such as defense industries, pursuit, search and rescue, agricultural purpose, hobby, etc. There are many studies related to object detection and tracking, but this system provides ease of changing scanning area and target definition, and adds target specification ability to make the target more specific. In addition to these features, doing all works autonomously without human assistance provides ease of use to end-user.

In some cases, image processing prevents to overlook the detection of the desired target. Also, small autonomous UAV system with image processing feature is cost effective against the use of aircraft such as helicopters or bigger UAVs. All these advantages make the use of this system inevitable.

This study can be developed by using a group of UAV in order to scan larger area autonomously within shorter duration as seen in Figure 3.1.

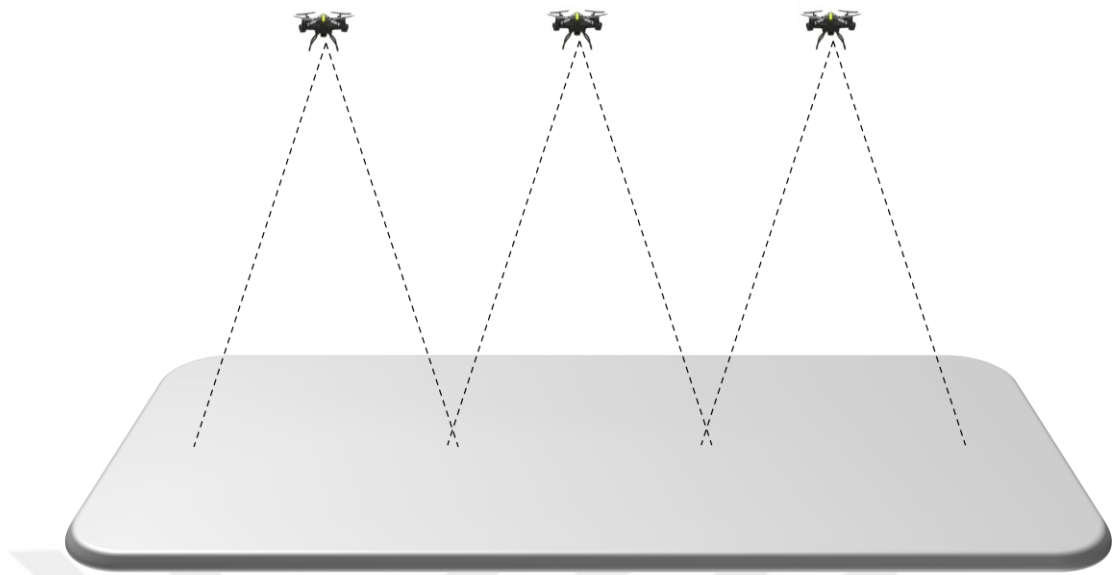


Figure 3.1 Illustration of searching mission by UAV group

## REFERENCES

- Alemdar, A. (2019). *Akıncı taarruzi insansız hava aracı (TİHA) ilk uçuşa hazırlanıyor*. Retrieved September 09, 2019, from <https://www.defenceturk.net/akinci-taarruzi-insansiz-hava-araci-tiha-ilk-ucusa-hazirlaniyor>.
- Bennett, A. E., Rickert, A. A., Crawford, W. B., Lamppin, E. D., & Warman, D. D. (2017). Autonomous aerial robot localization and target detection and manipulation in a GPS-denied environment. *2017 International Aerial Robotics Competition*, 1-12.
- Bogotobogo (n.d.). *Filters A - Average and GaussianBlur*. Retrieved April 11, 2019, from [https://www.bogotobogo.com/OpenCV/opencv\\_3\\_tutorial\\_imgproc\\_gaussian\\_median\\_blur\\_bilateral\\_filter\\_image\\_smoothing.php](https://www.bogotobogo.com/OpenCV/opencv_3_tutorial_imgproc_gaussian_median_blur_bilateral_filter_image_smoothing.php).
- Bradski, G. & Kaehler, A. (2008). *Learning OpenCV*. California: O'Reilly Media.
- Case, J. (2011). *Astro navigation demystified*. Weymouth: Bookcase Learning Resources.
- Codevilla, F., Müller, M., López, A., Koltun, V., & Dosovitskiy, A. (2018). End-to-end driving via conditional imitation learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 4693-4700.
- Custers, B. (2016). *The Future of Drone Use*. The Hague: Asser Press by Springer.
- Dronesxpress (n.d.). *4pcs DJI phantom 3 drone propeller 9450*. Retrieved February 15, 2019, from <https://www.dronesxpress.com.au/products/4pcs-dji-phantom-3-drone-propeller-9450>.

Dronetrest (n.d.). *Brushless motors - how they work and what the numbers mean*. Retrieved January 12, 2019, from <https://www.dronetrest.com/t/brushless-motors-how-they-work-and-what-the-numbers-mean/564/1>.

Engineerstoy (n.d.). *FlySky FS-i6 2.4 GHz 6 channel remote radio for quadcopter Tx Rx mode 2 FSi6*. Retrieved February 15, 2019, from <http://www.engineerstoy.com/quickview/index/view/path/robots-drones/flysky-fs-i6-2-4-ghz-6-channel-remote-radio-for-quadcopter-tx-rx-mode-2-fsi6.html>.

Fahlstrom, P. G., & Gleason, T. J. (2012). *Introduction to UAV systems* (4th ed.). United Kingdom: Wiley.

GeeksforGeeks (n.d.). *Erosion and dilation of images using OpenCV in python*. Retrieved May 18, 2019, from <https://www.geeksforgeeks.org/erosion-dilation-images-using-opencv-python/>.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing* (2nd ed.). New Jersey: Prentice Hall.

Holybro (n.d.). *Micro transceiver telemetry radio set*. Retrieved January 23, 2019, from <http://www.holybro.com/product/micro-transceiver-telemetry-radio-set/>.

Keipour, A., Mousaei, M., & Scherer, S. (2019). ALFA: A dataset for UAV fault and anomaly detection. *International Journal of Robotics Research*, 1-5.

Khurana, S. (2018). *A beginners' guide to understanding drones*. Retrieved March 30, 2019, from <https://medium.com/@sukantkhurana/a-beginners-guide-to-drones-38d215701c4e>.

Kumar, C., Vaddi S., & Sarkar, A. (2018). A case study on optimal deep learning model for UAVs. *ICLR 2019 Conference International Conference on Learning Representations*, 1-9.

- Lin, S., Wang, J., Peng, R., & Yang, W. (2019). Development of an autonomous unmanned aerial manipulator based on a real-time oriented-object detection method. *Sensors*, 1-19.
- Lutz, M. (2009). *Learning Python* (4th ed.). California: O'Reilly Media.
- Marshall, D. M., Barnhart, R. K., Shappee, E., & Most, M. (2016). *Introduction to unmanned aircraft systems* (2nd ed.). Florida: CRC Press.
- Mathews, N. (2015). *India seeks to strengthen unmanned fleet with Heron UAVs*. Retrieved August 07, 2019, from <https://www.ainonline.com/aviation-news/defense/2015-09-29/india-seeks-strengthen-unmanned-fleet-heron-uavs>.
- McClure, J. (2019). *A low-cost search-and-rescue drone platform*. Master Thesis, Rochester Institute of Technology, New York.
- Militaryfactory. (2019). *TAI Aksungur*. Retrieved August 07, 2019, from [https://www.militaryfactory.com/aircraft/detail.asp?aircraft\\_id=2150](https://www.militaryfactory.com/aircraft/detail.asp?aircraft_id=2150).
- Moran, L. M., Borja, G. A. B., & Casupanan, Y. H. A. (2014). *Automated indoor surveillance quadcopter with image recognition using OpenCV via Canny edge detection for avoiding stationary obstacles*. Bachelor's Thesis, Mapua Institute of Technology, Manila.
- Nonami, K., Kendoul, F., Satoshi, S., Wang W., & Nakazawa, D. (2010). *Autonomous flying robots*. Tokyo: Springer.
- OpenCV (n.d.). *Smoothing images*. Retrieved April 11, 2019, from [https://docs.opencv.org/3.4/dc/dd3/tutorial\\_gaussian\\_median\\_blur\\_bilateral\\_filter.html](https://docs.opencv.org/3.4/dc/dd3/tutorial_gaussian_median_blur_bilateral_filter.html).

OpenCV (n.d.). *Smoothing images*. Retrieved April 11, 2019, from [https://docs.opencv.org/3.4.2/d4/d13/tutorial\\_py\\_filtering.html](https://docs.opencv.org/3.4.2/d4/d13/tutorial_py_filtering.html).

Pisupply (n.d.). *Raspberry pi camera board v1.3 (5MP, 1080p)*. Retrieved January 23, 2019, from <https://uk.pi-supply.com/products/raspberry-pi-camera-board-v1-3-5mp-1080p>.

TutorialKart (n.d.). *OpenCV python image smoothing – Gaussian Blur*. Retrieved April 12, 2019, from <https://www.tutorialkart.com/opencv/python/opencv-python-gaussian-image-smoothing/>.

Tutorialspoint (n.d.). *OpenCV - Gaussian Blur*. Retrieved May 18, 2019, from [https://www.tutorialspoint.com/opencv/opencv\\_gaussian\\_blur.htm](https://www.tutorialspoint.com/opencv/opencv_gaussian_blur.htm).

Vaddi, S., Jannesari, A., & Kumar, C. (2019). Efficient object detection model for real-time UAV applications. 1-10.

Valavanis, K. P. (2007). *Advances in unmanned aerial vehicles*. The Netherlands: Springer.

Valavanis, K. P., & Vachtsevanos, G. J. (2015). *Handbook of unmanned aerial vehicles*. The Netherlands: Springer.

Valdes, R. (2004). *How the Predator UAV works*. Retrieved September 09, 2019, from <https://science.howstuffworks.com/predator.htm>.

Villán, A. F. (2019). *Mastering OpenCV 4 with Python*. Birmingham: Packt publishing.

## APPENDICES

### APPENDIX 1: Code Snippets for Filtering

```
real_image = frame.array
smoothing = cv2.GaussianBlur(real_image, (11, 11), 0)
hsv_converted = cv2.cvtColor(smoothing, cv2.COLOR_BGR2HSV)
masked = cv2.inRange(hsv_converted, hsv_l, hsv_u)
masked = cv2.dilate(masked, None, iterations=2)
cnts = cv2.findContours(masked.copy(),
                        cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if imutils.is_cv2() else cnts[1]
```

## APPENDIX 2: Code Snippets for Dimensional Detection

```
if len(cnts) == 4:  
    (x, y, w, h) = cv2.boundingRect(cnts)  
    aspect_ratio = w / float(h)  
  
    if (aspect_ratio >= 0.35 and aspect_ratio <= 0.44) or (aspect_ratio >= 2.3 and  
        aspect_ratio <= 2.8):  
        cv2.drawContours(imgOriginal, [c], -1, (0, 255, 0), 2)
```



### APPENDIX 3: Code Snippets for Finding HSV Values

For black color:

```
black_color = np.uint8([[[0,0,0 ]]])  
hsv_value_of_black_color = cv.cvtColor(black_color,cv.COLOR_BGR2HSV)  
print(hsv_value_of_black_color)
```

Output:

```
[[[ 0 0 0]]]
```

For green color:

```
green_color = np.uint8([[[0,255,0 ]]])  
hsv_value_of_green_color = cv.cvtColor(green_color,cv.COLOR_BGR2HSV)  
print(hsv_value_of_green_color)
```

Output:

```
[[[ 60 255 255]]]
```

#### APPENDIX 4: Code Snippets to Start Field Scanning

```
waypoint = new_coordinates (waypoint, 0, 20, 0);  
cmd = Command(0,0,0,  
    mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT,  
    mavutil.mavlink.MAV_CMD_NAV_WAYPOINT, 0, 1, 0, 0, 0, 0,  
    waypoint.lat, waypoint.lon, waypoint.alt)  
cmds.add(cmd)
```

```
waypoint = new_coordinates (waypoint, -5, 0, 0);  
cmd = Command(0,0,0,  
    mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT,  
    mavutil.mavlink.MAV_CMD_NAV_WAYPOINT, 0, 1, 0, 0, 0, 0,  
    waypoint.lat, waypoint.lon, waypoint.alt)  
cmds.add(cmd)
```

## APPENDIX 5: Code Snippets to Compute Necessary Parameters for Moving to Updated Location

```
def new_coordinates(current_location, distance_in_north, distance_in_east, altitude):

    radius_of_terrestrial_globe=6378137
    Longitude_in_radians = distance_in_east/(radius_of_
        terrestrial_globe*math.cos(math.pi*current_location.lat/180))

    New_Latitude = current_location.lat + ((distance_in_north/earth_radius) *
(180/math.pi))
    New_Longitude = current_location.lon + (Longitude_in_radians * 180/math.pi)
    New_Altitude = current_location.alt+altitude
    return LocationGlobal(New_Latitude, New_Longitude, New_Altitude)

waypoint = new_coordinates (waypoint, 0, 1, 0);
cmd = Command(0,0,0,
    mavutil.mavlink.MAV_FRAME_GLOBAL_RELATIVE_ALT,
    mavutil.mavlink.MAV_CMD_NAV_WAYPOINT, 0, 1, 0, 0, 0, 0,
    waypoint.lat, waypoint.lon, waypoint.alt)
cmds.add(cmd)
cmds.upload()
```