

JUNE 2023

M.Sc. in Mechanical Engineering

TARIK SEV

**REPUBLIC OF TÜRKİYE
GAZİANTEP UNIVERSITY
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES**

**DETERMINING THE DISTANCE AND POSITION OF MOVING AND
STATIONARY TARGETS WITH GPS AND SATELLITE DATA AND ADJUSTING
THE FIRING RANGES OF MILITARY WEAPON WITH IMAGE PROCESS
TECHNIQUES**

**M.Sc. THESIS
IN
MECHANICAL ENGINEERING**

**BY
TARIK SEV
JUNE 2023**

**DETERMINING THE DISTANCE AND POSITION OF MOVING AND
STATIONARY TARGETS WITH GPS AND SATELLITE DATA AND
ADJUSTING THE FIRING RANGES OF MILITARY WEAPON WITH
IMAGE PROCESS TECHNIQUES**

**M.Sc.
in
Mechanical Engineering
Gaziantep University**

Supervisor

Prof.Dr.Sedat BAYSEÇ

by

Tarık SEV

June 2023



©2023[Tarik SEV]



I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Tarık SEV

ABSTRACT

DETERMINING THE DISTANCE AND POSITION OF MOVING AND STATIONARY TARGETS WITH GPS AND SATELLITE DATA AND ADJUSTING THE FIRING RANGES OF MILITARY WEAPON WITH IMAGE PROCESS TECHNIQUES

SEV Tarık

M.Sc. in Department Of Mechanical Engineering

Supervisor: Prof. Dr. Sedat BAYSEÇ

June/2023

71 pages

In this study, some software languages were used to set the firing ranges of military weapons, such as matlab, and some methods were used in these software to make the gun sight more precise, make the moving object more visible, and some methods were used to track matlab, and the arrangement of the firing ranges of military weapon systems using matlab image techniques, as well as motion prediction methods were used to track moving and stationary objects on the system. Image processing techniques cover the threshold, before that, Gps and satellite systems were used to locate the target, and certain regions and coordinates are also tracked and monitored after satellite camera work, from which the coordinates are obtained, so the location is tracked and monitored, and then after additional moving objects are picked up, they are tracked by the systems here using matlab software targets which movement was tracked and captured. In addition, using optical flow methods based on motion estimation in matlab, the direction of the moving object was determined and the targets were tracked, and the targets moving in the direction of vector arrows were tracked through the system, then blob analysis was performed and the system tracked the target by labeling the moving object in the video as a rectangle

Key Words: Matlab image process, Motion estimation, Matlab Computer vision toolbox

ÖZET

GPS VE UYDU VERİLERİYLE HAREKET EDEN VE SABİT HEDEFLERİN UZAKLIK VE KONUMLARININ BELİRLENMESİ VE GÖRÜNTÜ İŞLEME TEKNİKLERİYLE ASKERİ SİLAHLARIN ATIŞ MENZİLLERİNİN AYARLANMASI

SEV Tarık
Yüksek Lisans Tezi, Makine Mühendisliği
Danışman: Prof.Dr.Sedat BAYSEÇ
Haziran/2023
71 sayfa

Bu çalışmada askeri silahların atış menzillerinin ayarlanması için bazı yazılım dilleri kullanıldı matlab gibi ve bu yazılımlarda bazı methodlar kullanıldı silah nişangahını daha hassas hale getirip hareket eden nesneyi daha görünür hale getirip ve takip etmek için bazı metotlar kulanıldı matlab üzerinde,ve askeri silah sistemlerinin atış menzillerinin matlab görüntü teknikleri kullanılarak düzenlenmesi aynı zamanda sistem üzerinde hareket eden ve sabit nesnelere izlemek için hareket tahmin yöntemleri kullanıldı.Görüntü işleme teknikleri eşiği kapsar, bundan önce Gps ve uydu sistemleri hedefin yerini tespit etmek için kullanıldı ve belirli bölgeler ve koordinatlar, koordinat elde edilen uydu kamera çalışmasından sonra da izlendi, böylece konum izlemesinden sonra hareket eden nesnelere alındıktan sonra buradaki sistemler tarafından matlab yazılımı kullanılarak izlenir hedefler ne tarafa hareket ettiyse takip edilip izlendi ve yakalandı.Ayrıca matlabda hareket tahmin yöntemlerine bağlı optik akış metotları kullanılarak hareketli nesnenin yönünü belirleyip hedefleri takip edildi ve vektör okları yönüne hareket eden hedefler sistem üzerinden takip edildi sonra blob analizi yapıp sistem video içerisindeki hareketli nesneyi dikdörtgen şeklinde etiketleyerek hedefi takip etti.

Anahtar Kelimeler: Matlab Görüntü işleme,Hareket tahmin etme,Matlab bilgisayar görüntü araç kutusu



‘Dedicated to my family’

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Dr.Sedat BAYSEÇ for his guidance and support throughout the study. I am thankful for his encouragement and motivation.

I would like to express my love and gratitude to my family for their support, always best wishes.

My mother, my father, my brother, I am grateful to all of them for their endless support and they encouraged and motivated me always by standing with me.

TABLE OF CONTENTS

ABSTRACT	VI
ÖZET.....	VII
ACKNOWLEDGEMENTS.....	VIII
TABLE OF CONTENTS.....	IX
LIST OF TABLES	X
LIST OF FIGURES	XII
LIST OF ABBREVIATIONS.....	XIV
CHAPTER I:LITERATURE SURVEY	1
1.Literature Research	1
CHAPTER II:INTRODUCTION	3
2.1.Motivation of Study	3
CHAPTER III:GPS AND SATELLITE SYSTEM	4
3.1.Definition of GPS	4
3.2.GPS Working Principle	4
3.3.GPS Waveform Generation	5
3.3.1.Cycle of generation gps signal	5
3.3.2 GPS Signal Structure	6
3.3.3.GPS Data Initialization	7
3.3.4.GPS Signal Generation	7
3.4.Working Principle Of Satellite	7
CHAPTER IV:GPS AND SATELLITE DATA FOR MILITARY APPLICATIONS.	10
4.1. Requirements of a Good Navigation System	10
4.2. Purpose of GPS and Satellite datas for Military applications	10
4.2.1.Navigation	11
4.2.2. Tracking	12
4.2.3.Bomb and missile guidance	12
4.2.4.Rescue	13
4.2.5.Map updation	13
CHAPTER V:SATELLITE IMAGING	14
5.1.Satellite image	14
5.2. Satellites and the EM Spectrum	14
5.3. Analyzing satellite images	14

5.4. Advanced Camera for Surveys	15
5.5. Wide Field Camera	15
CHAPTER VI:RESULTS OF IMAGE PROCESS TECHNIQUES FOR FIRE RANGES.....	17
6.1. Some methods to estimate targets	17
CHAPTER VII: GPS SENSOR USAGE WITH COORDINATES	53
7.1. GPS Sensor data with Plot on Matlab	53
CHAPTER VIII: CREATION OF MODEL OF ABONDING BOUNDARY	56
8.1. Model of abandoning boundary of target on ground	56
8.1.1. Subsystem for storing background data	56
8.1.2. Subsystem Detected	56
CHAPTER:IX: SOME CODES AS TAKING SNAP VIEW FROM SPACE	59
9.1. View and determination of the boundary from space	59
CHAPTER:X:CONCLUSION AND RECOMMENDATIONS	67
10.1. Recommendations,Conclusion and Future Work.....	67
REFERENCES	69
CIRRICULUM VITAE(CV)	71

LIST OF FIGURES

Figure 3.1. Cycle of generation of GPS signal.....	6
Figure 3.2. Satellite block diagram	8
Figure 3.3. Signal results of Satellite block diagram	8
Figure 3.4. Satellite and Earth Signal View	9
Figure 4.1. Block of Navigation.....	11
Figure 4.2. View of Navigation.....	12
Figure 4.3. View of Rescue alert satellite	13
Figure 5.1. Satellite camera	15
Figure 6.1. Code result after getting snap	19
Figure 6.2. View after getting snap	22
Figure 6.3. Object tracking predict result	23
Figure 6.4. Blob analysis object detection results in frame	24
Figure 6.5. Blob analysis results in frame	25
Figure 6.6. opticFlowLK result	28
Figure 6.7. opticalFlowFarneback result	35
Figure 6.8. opticalFlowHS in dark area	37
Figure 6.9. opticFlow block diagram	38
Figure 6.10. opticalFlow block diagram result	38
Figure 6.11. opticFlow block diagram result	39
Figure 6.12. opticFlow block diagram result with autothreshold	39
Figure 6.13. opticFlow block diagram result with autothreshold and filtering	40
Figure 6.14. opticalFlow result with threshold and region filter	40
Figure 6.15. opticalFlow result with threshold and region filter result	41
Figure 6.16. opticalFlow result with threshold and region filter	42
Figure 6.17. opticalFlow result with threshold and region filter	42
Figure 6.18. opticalFlowLKDoG in dark area	46
Figure 6.19. opticalFlowLKDoG result	47
Figure 6.20. Kalman filter and blob analysis result	50

Figure 6.21. Kalman filter result in dark area	51
Figure 7.1. GPS sensor	54
Figure 7.2. Gps reading reuslt	55
Figure 8.1. Abandoning boundary model	57
Figure 8.2. Abandoning boundary model result	57
Figure 8.3. Abandoning boundary model result	58
Figure 9.1. View of the border from space	59
Figure 9.2. View of the border from space	59
Figure 9.3. View of the border from space	60
Figure 9.4. View and determination of the boundary from space	60
Figure 9.5. War plane with motion estimation	63
Figure 9.6. War plane labelled target	64

LIST OF ABBREVIATIONS

GPS	Global Position System
DGPS	Differential Global Position System
CSEL	Combat Survivor Evader Locator
LLA	Latitude, Longitude and Altitude
CNAV	Civil Navigation
LNAV	Legacy Navigation
MSG	Message
TOW	Time of week
ROI	Return on Investment

CHAPTER I

LITERATURE SURVEY

1.Literature Research

Since the launch of the first Soviet Sputnik satellite in 1957, attempts have been made to employ space-based platforms for radio navigation in order to reduce the need for high frequency, high precision systems to operate in line of sight. The first successful such system was launched in 1959, and it is still in use today.

The early-late gate circuit recovers bit timing without relying on a clocking component in the spectrum of the incoming waveform. The circuit employs a feedback loop in which the magnitude variations in the outputs of matched filters affect the frequency of a local clocking circuit. In 1995, Roddy and Coolen Franks 1980 and Gagliardi 1991 provide extensive assessments of these and other approaches.

Ahmed El-Rabbany directs several research projects involving GPS, integrated navigational chart systems, and integrated navigation systems for land navigation and hydrographic surveying.

Rafael Gonzalez and Richard E. Woods studied Edge segments were first located, and then methods for directly dividing images into regions based on intensity values and attributes of these values were used to attempt to link the segments into borders. Although this finding holds for any number of classes, when more classes are added, it becomes less meaningful because there is only one variable (intensity) to consider. In reality, Fukunaga (1972) found that the between-class variance is typically described in terms of numerous variables written as vectors. When there is reason to suppose that the problem may be solved successfully with two thresholds, adopting multiple global thresholding is generally seen as a viable strategy. Usually, more than only intensity measurements are used to solve applications with more than two thresholds.

A threshold is computed at each point, (x, y) in the image depending on one or more specified parameters in the area around (x, y) Richard E. Woods' variable

thresholding research. x y despite the fact that this may appear like a time-consuming process, current algorithms and hardware enable quick neighborhood processing, especially for routine operations like logical and arithmetic ones.

In his 2005 study, Zhu, Y., Comaniciu, D., Ramesh, V., Pellkofer, M., and Koehler, T. Various vision algorithms have been developed to address the challenge of object detection. To detect the rear-view appearance of cars, several algorithms have extensively used empirical knowledge about item appearance, such as symmetry, horizontal and vertical occluding boundaries around vehicle border. These methods are computationally efficient, but they lack resilience since the parameters used in edge detection and hypothesis generation are sensitive to lighting conditions and picture collection dynamic range. To achieve reliable object detection, several appearance-based techniques have been proposed.

In 1997, James R. Parker used histogram-based algorithms for different applications, and color histograms have been used in a variety of ways for QBE in previous work.

CHAPTER II

INTRODUCTION

2.1 Motivation of Study

The purpose of the study is to arrange fire ranges of military weapons, describe target's coordinates and detect where targets are as using some methods such as image process techniques in Matlab with GPS sensor. GPS(Global Positioning System) is using to detect targets,coordinates,position and placement therefore all high technology weapons such as fighter plane,missiles,ammos, according to GPS system.Actually GPS is going to be used as taking signal on some location and that location will be viewed and scanned with satellite camera. In image process techniques of Matlab special methods are going to use such as kalman filter,optical flow,threshold,autothreshold,blob analysis,edge enhancement and while using some codes related to track and detect objects,targets which move in real time or in any video.In 1964, the Space Detector Ranger 7 captured thousands of photos of the moon that were processed using image manipulation techniques, including geometry correction, gradation transformation, and noise reduction. These images were evaluated according to the moon's environment and position relative to the sun, resulting in a highly successful mapping of the moon's surface. The goal of early image processing was to increase image quality. It was intended for humans to boost their visual impact. In image processing, the input is a low-quality image, and the output is a higher-quality image. Actually, the other purpose of this thesis is to use these types of methods to adjust the sighting of the military weapon using matlab image processing techniques to make the ranges of some special military weapons used to protect the borders of the country more precise and to focus on the target more precisely, and to make the image more sensitive and track moving targets. To capture by determining and following the direction of the moving object or target by making motion prediction at the same time.

CHAPTER III

GPS AND SATELLITE SYSTEM

3.1. Definition of GPS

GPS means that the World Position System is a satellite navigation system that locates the bottom of associated objects and it is removed from the only radio navigation systems in existence. The area units are Omega, Loran, VOR/DME, ILS, Transit and finally GPS. The four main ground-based area units, transit and GPS area units, each space-based. In general, I tend to use two systems. Sputnik-I (October 4, 1957) and Sputnik-II (October 3, 1957) were the first artificial satellites to be launched into space. These satellites were used by the Russians in subsequent years, providing essential knowledge for the space missions they will carry out. It is the most frequently utilized and widely used satellite system for position determination reasons, and it has NAVSTAR-GPS to its audience. The Navy Navigation Satellite System is the system on which this system is based. It was created in the United States, and the first satellite was launched on April 18, 1960 by the Navy Navigation Satellite. The system was being used for military purposes. [3]

3.2. GPS Working Principle

GPS is a system. It consists of three parts: satellites, ground stations, and receivers.

I. Satellites move around like stars in constellations, and we know where they are because they keep sending out alerts.

II. Floor stations use radar to ensure that the satellites are truly where we think they are.

III. A receiver is a device you would possibly find to your telephone or on your automobile that keeps searching out signals from the satellites. The receiver receives records about how far away it is from some of these satellites. Once the receiver has calculated the gap to 4 or more satellites, it is aware of exactly where you are.

Measure several GPS satellites within line-of-sight of a receiver in the world square. The transmitter's GPS sends its position and time to the receiver's GPS at more or less regular intervals. The signal sent to the receiver measures the square of the radio

wave. The gap between the GPS receiver and the satellites can also be calculated using the time difference between the characters sent by the GPS satellite TV for laptops and the time the GPS receiver could not inherit.[6-3]

3.3.GPS Waveform Generation

GPS waveform technology shows the truth and how to generate complex baseband waveforms including lateral navigation, civil navigation. Record spread ends with C/A code, P code and L2 CM/L2 CL code.[6]

3.3.1.Cycle of generation of gps signal

Follow the steps below to create a GPS Shield.

- I. Receive GPS data bits using the configuration parameters described in the following. Statistics bits are generated at the selected rate of about 50 bits per second.
- II. They may use overcharge spreading code to implement low-value statistical bits. GPS specifies several types of spread codes including C/A codes, P codes, and L2 CM/L2 CL codes. When the Associated Degreeti spoof operation mode is enabled, you can add codes such as the Y code, which is the spatial unit used in place of the P code. The P code and the Y code Called the PY code, the area unit is equal to the P-code. by choosing code with internal branching based on parameter configuration, we generate tough GPS baseband waveforms from unopened bits using spread code and hence split design.

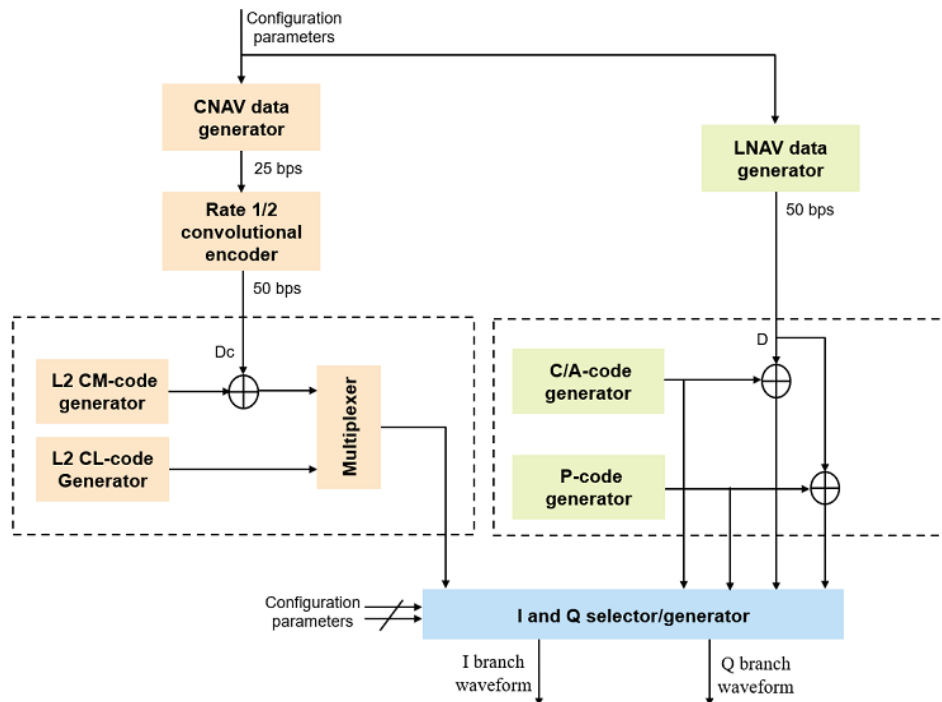


Figure 3.1. Cycle of generation of GPS signal [3-13]

3.3.2. GPS Signal Structure

The GPS was started by the U.S. Department of Defense in 1973. The first prototype spacecraft was launched in 1978 and the full constellation of 24 satellites became operational in 1993. Some houses transmit identical lateral and civil navigation but have completely different body shapes. LNAV data is transmitted in 1500-bit frames, with each body consisting of five three-hundred-bit subframes. Each subframe takes 6 seconds to transmit and each part takes 30 seconds to transmit at a data rate of 50 bits per second. For each sentence, each subframe consists of 10 30-bit words (24 register bits and 6 parity bits). GPS stats includes information on the clock and satellite position. This parent element defines the LNAV record's frame geometry. The following fields are present in the message types depicted in this diagram: Some terms are defined further below.

- I. PRN identification: Pseudo-random noise index (PRN)
- II. MSG: Message
- III. TRAILER: Time zone
- IV. CRC: Consider cyclic redundancy. "Show visible objects" allows you to view spectral and correlation graphs, and the "Write waveforms for reports" property allows you to write complex baseband waveforms for customized reports. GPS time is established by the Control Segment and is referenced to Coordinated Universal Time (UTC) as maintained by the U.S. Naval Observatory (UTC (USNO)) zero time-point defined as midnight on the night of January 5, 1980/morning of January 6, 1980. [3]

3.3.3.GPS Data Initialization

Allows generation of CNAV information, initialization of "recording" configuration items, and formation of configuration items from auxiliary GPS navigation configuration items. Encoding is shown in the following facts. The Cnavconfig code is used to generate CNAV records and initialize statistical configuration items to get configuration items from auxiliary GPS navigation configuration items. [6-3]

3.3.4.GPS Signal Generation coding

The GPS L1 main carrier signal, at a frequency of approximately 1575 MHz, is modulated by two codes: a Coarse/Accept (C/A) code also known as a Civil Code and a Precision/Safety (P) code. /Y), reserved by encryption for authorized military and civilian users.[6-3]

3.4.Working Principle Of Satellite

Along with technical advances in numerous sectors of research, scientists from several countries developed various proposals for space communications. Scott Madry and Joseph N Pelton present a complete review of this interesting new field. It covers small satellite technologies, applications and services, design and manufacture, launch arrangements, ground systems, and economic and regulatory arrangements.[1]

A satellite is a frame that orbits another frame in a mathematically predictable route called an orbit. Communications satellites for PCs are important not only in their own right, but also as regional microwave repeaters that serve communications, radio, and other television and internet programming.

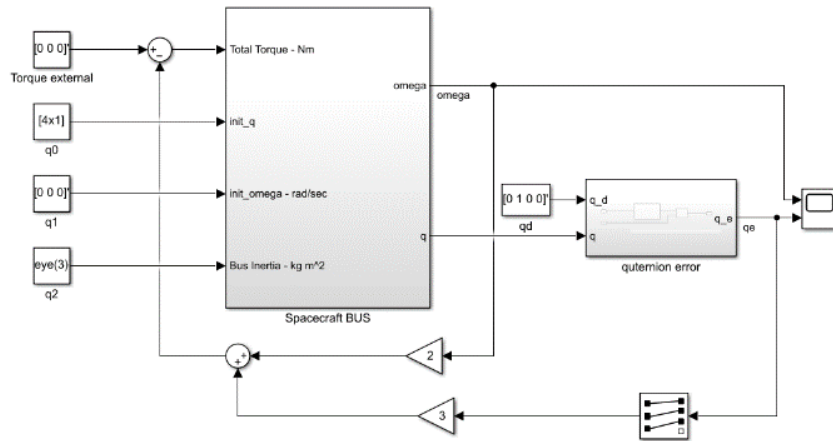


Figure 3.2. Satellite block diagram

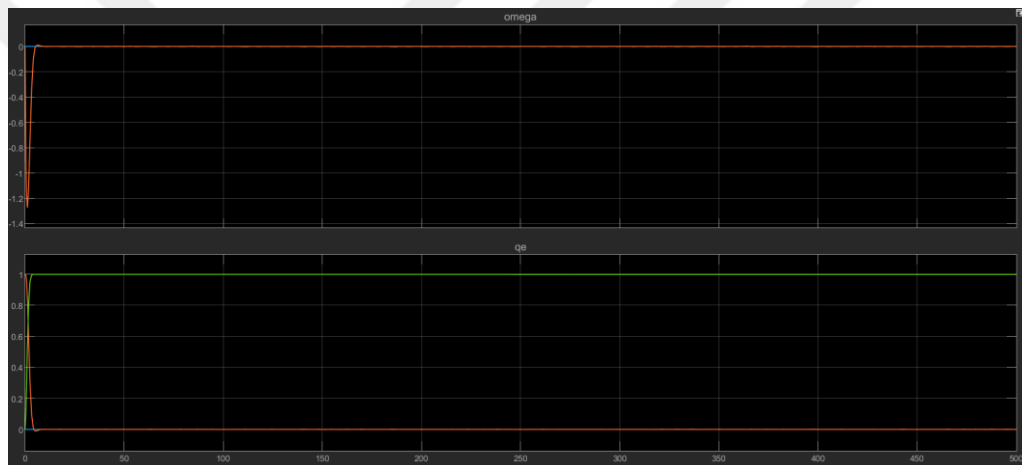


Figure 3.3. Signal results of Satellite block diagram

A repeater is a circuit that amplifies received characters and retransmits them. However, the repeater works like a transponder, translating the frequency band of the transmitted characters from the received characters. The frequency on which the beacon is broadcast within the area is known as the uplink frequency, while the frequency on which it is broadcast far from the transponder is known as the downlink frequency. Attempts have been made to use area-based forests for radionavigation to eliminate line-of-sight of high-frequency, high-precision systems. Communications satellites are in orbits that are synchronous with the rotation of the Earth. This allows for complete simplification of the machine, each ground terminal always working with the same communication satellite TV for PC.

The following figure illustrates this concept clearly.

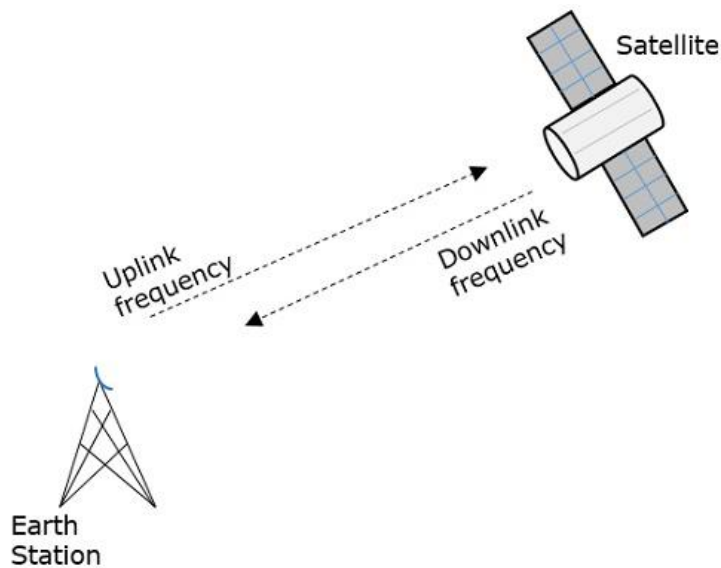


Figure 3.4. Satellite and Earth Signal View

With satellite communications, we know that we can use satellites in space to transmit knowledge from end to end. It works so that a signal is sent from a ground station to a satellite, which then forwards the signal to another ground station as well. However, in our previous content on satellite transmission, we said that the signal is transmitted from the earth station to the satellite. However, when the uplink signal received by the satellite is transmitted to the downlink, it is called downlink. Signal frequencies across uplink and downlink are denoted by uplink frequency and downlink frequency respectively. Note at this point that the uplink and downlink frequencies must be different. This is due to the fact that if 2 are identical, signal interference will occur if the transmit and downlink antennas are in close proximity. Another important factor in separating uplink and downlink frequency registrations in PC satellite TV is the application of filters. Here we can use a low bypass filter that can block the high frequency outputs of the uplink symbols and skip the low frequency outputs that form the downlink symbols.

CHAPTER IV

GPS AND SATELLITE DATA FOR MILITARY APPLICATIONS

4.1. Requirements of a Good Navigation System

The fundamental wishes of all defense force with appreciate to an amazing navigation gadget can be enumerated as follows.

- I. Correct
- II. All-climate successful
- III. Smooth to operate
- IV. Portable

GPS devices in use today meet these requirements. GPS accuracy ranges from meters to tens of meters, meeting Army requirements for navigation capabilities. However, precision target placement, such as air movement and missile attacks, requires mm-level accuracy. This is achieved with the help of differential GPS (DGPS). However, achieving this level of accuracy requires good error modeling. Similarly, GPS satellite displays do not suffer as severe weather conditions as traditional terrestrial radio displays. This is an important requirement as the military requires an all-weather navigation system. Al-Bayari and Sadoun (2005) report that attempts have been made to identify novel applications that will improve the quality of our lives while utilizing GNSS advantages.

Maximum of state-of-the-art GPS receivers are pretty clean to use and offer role in each latitude and longitude and neighborhood map projection system coordinates, and also provide information in the WGS-eighty four coordinate machine. Further, GPS receivers have dramatically reduced in size and weight through the years, making them more transportable. As an example, GPS receivers are actually built into commercially to be had wristwatches.

4.2. Purpose of GPS and Satellite datas for Military applications

There are numerous kinds of applications for military purposes, such as the necessities for a very good navigation system, navigation, monitoring, bomb and

missile guidance, rescue, map updating, image display of a couple of missile structures. Actually gps system is also used to send signal in order to describe point that is sent signal and location is described on gps system. When satellite camera only track those points ,on the other words camera systems are focused to described locations by satellite as data that get information gps coordinates and may show on navigation system therefore zones which is described are found and shoot by weapon systems.

4.2.1 Navigation

Navigation is the most enjoyable task for soldiers operating under cover of darkness in opposite territory, as there are no easily identifiable landmarks on unfamiliar territory and on the ground. Squads have long used the night sky to determine direction. but cannot determine its role on earth. GPS replaces the traditional compass in a soldier's backpack. Special forces and crack groups also use these devices to reach and destroy key enemy centers. Such teams can accurately deflect air and artillery fire with proper role data. Similarly, we can quickly get the position of the weapon with the help of GPS. This is because in modern combat, artillery batteries must cycle regularly to catch up with attacking forces and avoid counterattacks. Convoy actions can also be efficiently tracked and planned using GPS. The INTELSAT VII-VII/A series was launched over a period from October 1993 to June 1996

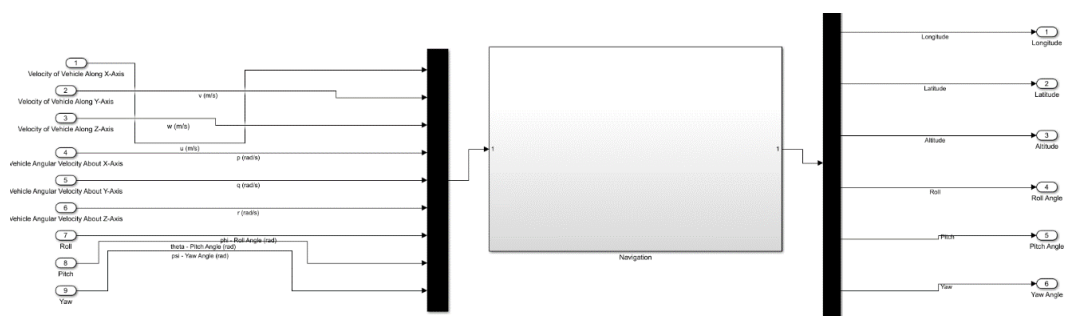


Figure 4.1. Block of Navigation[13]

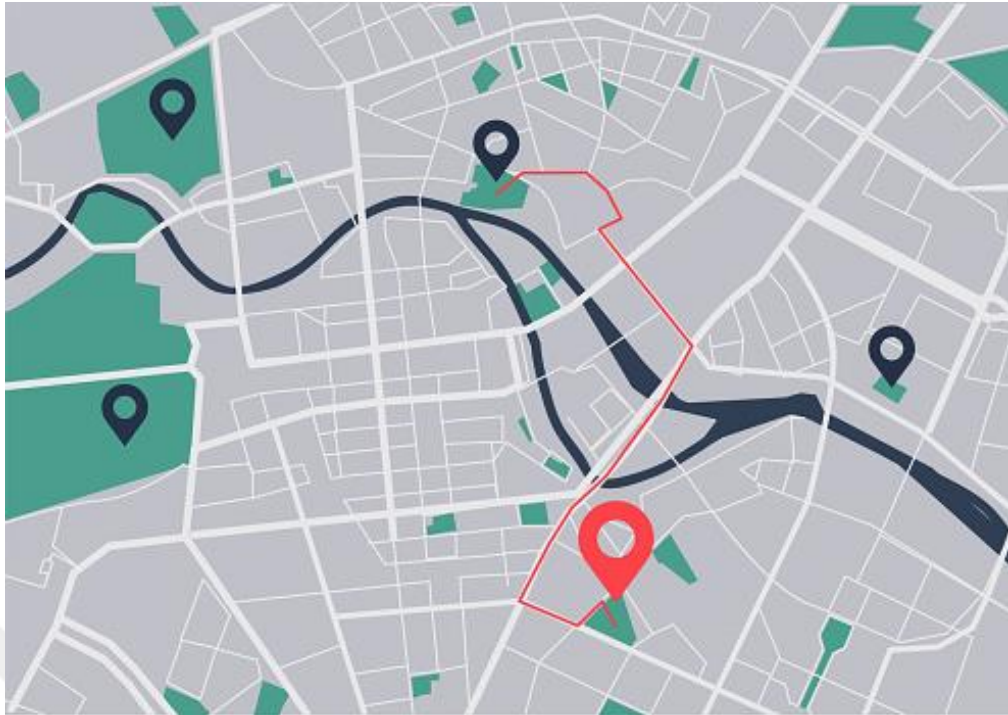


Figure 4.2. View of Navigation

4.2.2. Tracking

In the navy, ability targets have to be continuously tracked earlier than they're labeled as adverse and engaged with the aid of several guns structures, with this understanding serving as enter to superior weapons systems including missiles, bombs, and so forth. It is a compact, light-weight, low-value, and without problems transferable or cell GPS -based monitoring device that makes use of modern GPS -information, radio circuits, and technology to provide extremely correct time-space function facts (TSPI) over up to 10 gadgets, along with floor vehicles, helicopters, and fixed-wing plane. The gadget is surprisingly standardised, geared up with industry-fashionable hardware, and can be effortlessly modified to fulfill the particular desires of person take a look at and monitoring packages.

4.2.3. Bomb and Missile guidance

Modern weapon structures are designed to use traces from GPS for target acquisition and guidance. Cruise missiles used by some nations to strike targets at long ranges use multi-channel GPS receivers to keep track of their position on the wings. A rocket engine (MLRS) is fired several times and uses GPS-based route location information to position itself and align the launch field with the target in a very short

time, reducing the chances of detection and counterattack. Using his DGPS in the Army's Guidance Development (Area) application at several international sites led to the development of the 2000 Avoirdupois Drift Bomb, which uses his GPS seeker for guidance instead of an optical maser. . The bomb was controlled via four of his DGPS base stations at a distance of about 1000 nautical miles, allowing him to accurately hit a target 11 miles away from the drop target.

4.2.4. Rescue

Relief and emergency operations are all other areas where GPS will prove valuable to the Army. Responders use GPS to reduce latency in order to determine the condition of victims during operations. For example, some countries' air forces are already using GPS-based generations and developing Survivor Evader Locator (CSEL) fighter aircraft. By integrating a GPS receiver and communication radio, the new device will allow search and rescue teams to locate the crew of a downed aircraft faster and more accurately than before.



Figure 4.3. View of Rescue alert satellite

4.2.5. Map Updation

For planning at various stages of military headquarters, protection forces want correct and up-to-date maps at numerous scales.

CHAPTER V

SATELLITE IMAGING

5.1. Satellite Image

Faraway sensing is the method of the scanning of the Earth using a satellite or excessive-flying plane/plane. The earth has a range of satellites which might be equipped with exceptional-supposed systems ones scanning the earth. These designs have a completely unique reason. Satellites collect electromagnetic radiation reflected from the Earth the use of various kinds of sensors. Passive sensors, which do no longer require electricity, accumulate radiation emitted by way of the sun and contemplated by the Earth. Active sensors produce their very own radiation and examine it after it's far meditated back from the Earth. Lively sensors require a widespread quantity of electricity to emit radiation, but they're useful because they can be utilized in any season and at any time of day and they could emit radiation that the solar does no longer provide.

5.2.Satellites and the EM Spectrum

Whilst humans can only perceive a small portion of the electromagnetic spectrum (seen mild), satellite sensors can use infrared, ultraviolet, or maybe microwaves. Those invisible kinds of light are assigned a visible coloration when satellite tv for pc pix are created. As a end result, satellite pictures with "unnatural" colorations, which includes the one at the right, are commonplace.

5.3.Analyzing satellite images

Satellite for PC imaging is useful because it can identify unusual surfaces and objects based on their response to radiation. Smooth surfaces like lines reproduce nearly all of the energy applied to them in a single path. This is called mirror reflection. On the other hand, rough surfaces like wood reflect electricity in all directions. This is called diffuse reflection. Capturing unique reflection types can help measure changes in forest density, abundance, and cover.

Objects also respond differently to certain wavelengths of radiation.

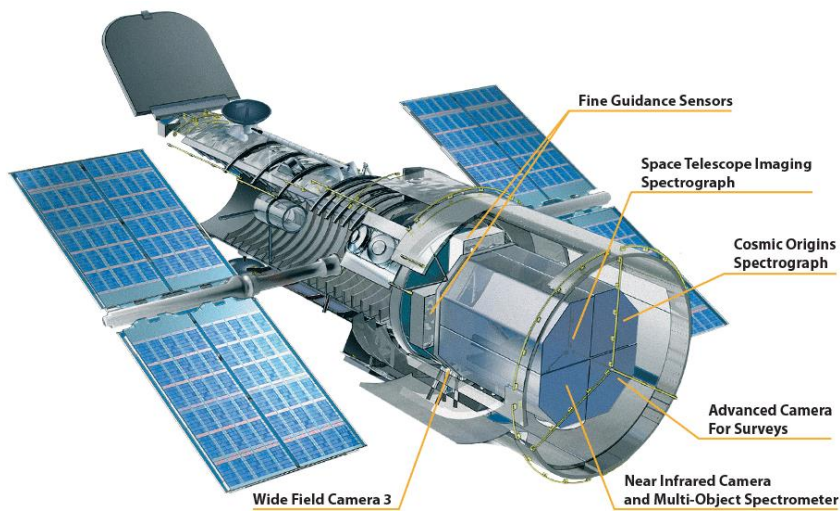


Figure 5.1. Satellite camera

5.4. Advanced Camera for Surveys

3rd technology imaging camera. This digital camera is optimized for campaign research and large-scale image processing. The initial design and scientific capabilities of ACS were defined by a team based at Johns Hopkins University. ACS was assembled and tested extensively at Ball Aerospace & Technologies Corp.

5.5. Wide Field Camera

Good discipline the digital camera is the telescope's primary image generator. Equipped with cameras that record visible and ultraviolet (UV) wavelengths, it is 35 times more sensitive in the UV range than its predecessors, the Widefield and Planetary Digicam 2. A second digicam was developed that can display infrared (IR) light. IR resolution increases from 65,000 to 1 million pixels. The combination of field of view, sensitivity, and low detector noise makes Hubble a 15-20x improvement over previous IR cameras. I have these codes for capturing objects that seem to snap.

The Video Labeler application is used to neatly label square ROI (ROI) labels, polyline ROI labels, pixel ROI labels, and video scenes.

Rectangle - Draws a 2D rectangular bounding box label around objects in an image
Vehicles, boats, helicopters, tanks, etc.



CHAPTER VI

RESULTS OF IMAGE PROCESS TECHNIQUES FOR FIRE RANGES

6.1. Some methods to estimate targets

There are a few methods such as `vidDevice` and `videoinput` to take real time image. Firstly when `vidDevice` is used, obtained in the following code group in order to get snap any image.[13]

```
vidDevice = imaq.VideoDevice('winvideo', 1, 'YUY2_1280x720', ...
    'ReturnedColorSpace', 'rgb', ...
    'DeviceProperties.Brightness', 64, ...
    'DeviceProperties.Sharpness', 7);

nextId=1
nFrames = 0;
while (nFrames<2000)
frame = vidDevice();
frame=T
data = getsnapshot(T);
data_0 = getsnapshot(T);
data_0 = rgb2gray(data_0);
data_1 = getsnapshot(T);
data_1 = rgb2gray(data_1);
data_2 = imabsdiff(data_0,data_1);
data_2 = medfilt2(data_2, [3 3]);
data_2= im2bw(data_2,0.05);
data_2 = bwareaopen(data_2,320);
pause(0.1);
red_01 =data( :, : ,1);
green_02 =data(:, :,2);
```

```

blue_03=data( :,:,3);
Filt_1 = 16+(0.2567890625 * red_01)+( 0.50412890625 * green_02) +( 0.09790625
* blue_03);
Filt_2= 128+(-0.14822265625 * red_01)- (0.2909921875 * green_02) +
(0.43921484375* blue_03);
Filt_3 = 128+(0.43921484375 * red_01)- (0.3677890625 * green_02) -(
0.07142578125 * blue_03);
pause(0.01);
[R3r R3c]= find((Filt_1>Filt_2) & (Filt_2>Filt_3) );
ruleIIIpixel=size(R3r);
Ir3= uint8(zeros(size(data)));
bk = bwlabel(data_2, 8);
dedector_1 = regionprops(bk, 'BoundingBox', 'Centroid','Area');
handles.output = hObject;
axes(handles.axes3);
imshow(data_2);
handles.output = hObject;
axes(handles.axes1);
imshow(data_0);
set(handles.text8,'BackgroundColor','blue');
set(handles.OperatorText,'String','Scanning');
for object = 1:length(dedector_1)
bk = dedector_1(object).BoundingBox;
yk(1) = dedector_1(object).Centroid;
yk(2) = dedector_1(object).Area;
rectangle('Position',bk,'EdgeColor','g','LineWidth',2)
plot(yk(1),yk(1), '-m+')

ba = stats(object).BoundingBox;
bd(1) = stats(object).Centroid;
bc(2)=stats(object).Area;
rectangle('Position',bb,'EdgeColor','r','LineWidth',2)
plot(bd(1),bd(2), '-m+')

```

```

for i4=1:ruleIIIpixel-1
Ir3(R3r(i4),R3c(i4),1) =red_01(R3r(i4),R3c(i4));
Ir3(R3r(i4),R3c(i4),2) =green_02(R3r(i4),R3c(i4));
Ir3(R3r(i4),R3c(i4),3) =blue_03(R3r(i4),R3c(i4));
i4=i4+1;
B_1= im2bw(Ir3,0.9);
B_1 = bwareaopen(B_1,300);
C_1 = bwlabel(B_1, 8);
dedector_1 = regionprops(C_1, 'BoundingBox', 'Centroid','Area');
C_1 = dedector_1(object).BoundingBox;
y(1) = dedector_1(object).Centroid;
y(2) = dedector_1(object).Area;
rectangle('Position',C_1,'EdgeColor','r','LineWidth',2)
plot(y(1),y(1), '-m+')

```

as using vidVideo,taken real time image ,track and scan ground and sky therefore obtained in the following image and capture target by labelled on image process technique.

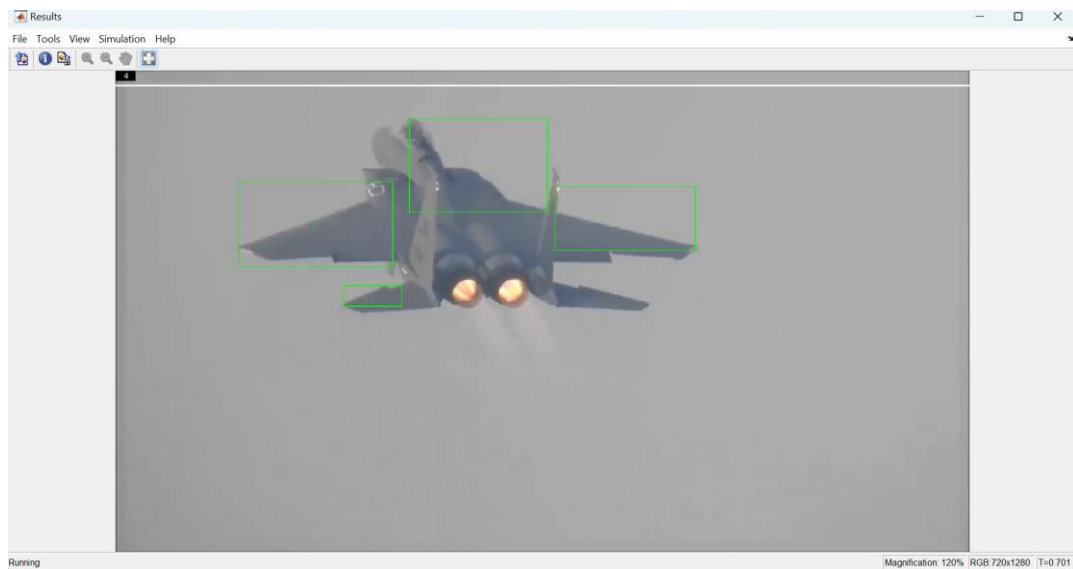


Figure 6.1. Code result after getting snap

Other one method is to take real time image as using videoinput ,as using that method obtained in the following code group in order to get snap image.[13]

Matlab applications[13] have designed variable object getsnap block or code members in order to label and track objects which move.

```
vid = videoinput('winvideo',1);
set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb')
vid.FrameGrabInterval = 3;
start(vid)
nextId=1
nFrames = 0;
while (nFrames<2000)
frame = vidDevice();
frame=T
data = getsnapshot(T);
data_0 = getsnapshot(T);
data_0 = rgb2gray(data_0);
data_1 = getsnapshot(T);
data_1 = rgb2gray(data_1);
data_2 = imabsdiff(data_0,data_1);
data_2 = medfilt2(data_2, [3 3]);
data_2= im2bw(data_2,0.05);
data_2 = bwareaopen(data_2,320);
pause(0.1);
red_01 =data( :, ,1);
green_02 =data(:, ,2);
blue_03=data( :, ,3);
Filt_1 = 16+(0.2567890625 * red_01)+( 0.50412890625 * green_02) +( 0.09790625
* blue_03);
Filt_2= 128+(-0.14822265625 * red_01)-(-0.2909921875 * green_02) +
(0.43921484375* blue_03);
Filt_3 = 128+(0.43921484375 * red_01)- (0.3677890625 * green_02) -(
0.07142578125 * blue_03);
pause(0.01);
```

```

[R3r R3c]= find((Filt_1>Filt_2) & (Filt_2>Filt_3) );
ruleIIIpixel=size(R3r);
Ir3= uint8(zeros(size(data)));
bk = bwlabel(data_2, 8);
dedector_1 = regionprops(bk, 'BoundingBox', 'Centroid','Area');
handles.output = hObject;
axes(handles.axes3);
imshow(data_2);
handles.output = hObject;
axes(handles.axes1);
imshow(data_0);
set(handles.text8,'BackgroundColor','blue');
set(handles.OperatorText,'String','Scanning');
for object = 1:length(dedector_1)
    bk = dedector_1(object).BoundingBox;
    yk(1) = dedector_1(object).Centroid;
    yk(2) = dedector_1(object).Area;
    rectangle('Position',bk,'EdgeColor','g','LineWidth',2)
    plot(yk(1),yk(1), '-m+')

    ba = stats(object).BoundingBox;
    bd(1) = stats(object).Centroid;
    bc(2)=stats(object).Area;
    rectangle('Position',bb,'EdgeColor','r','LineWidth',2)
    plot(bd(1),bd(2), '-m+')

for i4=1:ruleIIIpixel-1
    Ir3(R3r(i4),R3c(i4),1) =red_01(R3r(i4),R3c(i4));
    Ir3(R3r(i4),R3c(i4),2) =green_02(R3r(i4),R3c(i4));
    Ir3(R3r(i4),R3c(i4),3) =blue_03(R3r(i4),R3c(i4));
    i4=i4+1;
    B_1= im2bw(Ir3,0.9);
    B_1 = bwareaopen(B_1,300);
    C_1 = bwlabel(B_1, 8);

```

```

dedector_1 = regionprops(C_1, 'BoundingBox', 'Centroid','Area');
C_1 = dedector_1(object).BoundingBox;
y(1) = dedector_1(object).Centroid;
y(2) = dedector_1(object).Area;
rectangle('Position',C_1,'EdgeColor','r','LineWidth',2)
plot(y(1),y(1), '-m+')

```

and obtained real time image and track and scan objects which acts by labelled on ground and sky.



Figure 6.2. View after getting snap

The image used in the threshold and contrast tests is divided into white and black objects. White objects are those that move, have tracking applied, and then acquire targets. Thresholding is, in fact, a simple but effective method of separating an image into foreground and background. This image analysis technique is a type of image segmentation in which objects are separated by converting grayscale images to binary images. Image thresholding works best with images that have a high contrast.

return;

end

```
invisibleForTooLong = 26;
```

```
ageThreshold = 9;
```

```
ages = [tracks(:).age];
```

```
totalVisibleCounts = [tracks(:).totalVisibleCount];
```

```
visibility = totalVisibleCounts ./ ages;
```

```
lostInds = (ages < ageThreshold & visibility < 0.7) | ...
```

```
[tracks(:).consecutiveInvisibleCount] >= invisibleForTooLong;
```

```

tracks = tracks(~lostInds);
end

```



Figure 6.3. Object tracking predict result

Data which is obtained by some image process such as bbox,opticalflow,blob analysis,opticalFlowLK,opticalFlowFarneback,opticalFlowHS,autothreshold process, opticalFlowLKDoG, configureKalmanFilter.

Matlab application [13] some code groups such as predict and detection are used to track objects but here Studied on blob analysis are used with below codes.

```

v = VideoReader("C:\Users\Dell\Desktop\VIDEOLAR\VIDEO22.mp4")
detector = vision.ForegroundDetector(...
'NumTrainingFrames', 4, ...
'InitialVariance', 25*25);
blob = vision.BlobAnalysis(...
'CentroidOutputPort', false, 'AreaOutputPort', false, ...
'BoundingBoxOutputPort', true, ...
'MinimumBlobAreaSource', 'Property', 'MinimumBlobArea', 240);
shapeInserter = vision.ShapeInserter('BorderColor','White');

```

```

videoPlayer = vision.VideoPlayer();
while hasFrame(v)
frame = readFrame(v);
fgMask = detector(frame);
bbox = blob(fgMask);
output = shapeInserter(frame,bbox);
videoPlayer(output);
pause(0.2);
end

```

To remove objects from video code, such as fgMask, vision. A ForegroundDetector compares a color or grayscale video frame to a background model to determine whether individual pixels are in the foreground or background. The foreground mask is then computed. Background subtraction can be used to detect foreground objects in camera images.

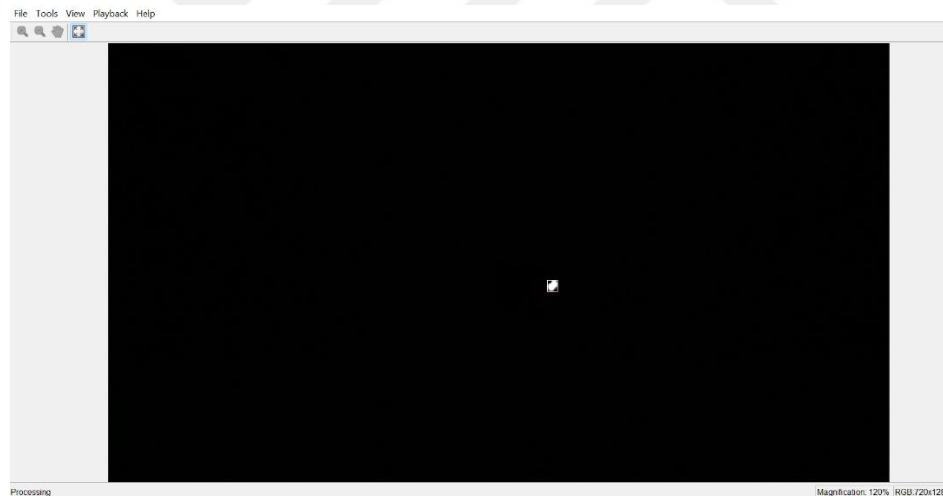


Figure 6.4. Blob analysis object detection results in frame

```

v = VideoReader("C:\Users\Dell\Desktop\VIDEOLAR\VIDEO8.mp4")
detector = vision.ForegroundDetector(...
'NumTrainingFrames', 4, ...
'InitialVariance', 25*25);

```

```

blob = vision.BlobAnalysis(...
'CentroidOutputPort', false, 'AreaOutputPort', false, ...
'BoundingBoxOutputPort', true, ...
'MinimumBlobAreaSource', 'Property', 'MinimumBlobArea', 240);
shapeInserter = vision.ShapeInserter('BorderColor','White');
videoPlayer = vision.VideoPlayer();
while hasFrame(v)
frame = readFrame(v);
fgMask = detector(frame);
bbox = blob(fgMask);
output = shapeInserter(frame,bbox);
videoPlayer(output);
pause(0.2);
end

```

Matlab application [13] some code groups such as predict and detection are used to track objects but here Studied on blob analysis are used with below codes.



Figure 6.5. Blob analysis results in frame

There are many vehicles in the image and video player, and we tracked them using blob analysis and detector codes, so any target or object was tracked. That is where codes come in handy.

```
detector = vision.ForegroundDetector(...  
'NumTrainingFrames', 4, ...  
'InitialVariance', 25*25);  
To trace and track them
```

In real time on matlab;

```
v = imaq.VideoDevice with properties:  
Device: 'Integrated Webcam (winvideo-1)'  
VideoFormat: 'YUY2_1280x720'  
ROI: [1 1 1280 720]  
ReturnedColorSpace: 'rgb'  
ReturnedDataType: 'single'  
ReadAllFrames: 'off'  
DeviceProperties: [1×1 imaq.internal.DeviceProperties]  
nextId = 1
```

```
vidDevice = imaq.VideoDevice('winvideo', 1, 'YUY2_1280x720', ...  
    'ReturnedColorSpace', 'rgb', ...  
    'DeviceProperties.Brightness', 64, ...  
    'DeviceProperties.Sharpness', 7);
```

```
detector = vision.ForegroundDetector(...  
'NumTrainingFrames', 4, ...  
'InitialVariance', 25*25);  
blob = vision.BlobAnalysis(...  
'CentroidOutputPort', false, 'AreaOutputPort', false, ...  
'BoundingBoxOutputPort', true, ...  
'MinimumBlobAreaSource', 'Property', 'MinimumBlobArea', 240);  
shapeInserter = vision.ShapeInserter('BorderColor', 'White');  
videoPlayer = vision.VideoPlayer();  
nextId=1  
nFrames = 0;
```

```

while (nFrames<2000)
    frame = vidDevice();
    fgMask = detector(frame);
    bbox = blob(fgMask);
    output = shapeInserter(frame,bbox);
    videoPlayer(output);
    pause(0.2);
end

```

There is tracking in addition to optical flow, and the code group is as follows:
 optical flow LK The Lucas-Kanade method is used to estimate optical flow objects. To estimate the victim direction and velocity of moving objects, optical flow objects are created using the Lucas-Kanade method. To calculate the optical flow vector, use the flow object estimator. The concept of optical flow was introduced by the American psychologist James J. Gibson in the 1940s to describe the visual stimulus provided to animals moving through the world.[8]

Optical Flow LK

The Lucas-Kanade technique is a commonly used differential method for optical flow estimates in computer vision that was created by Bruce D. Lucas and Takeo Kanade. It assumes that the flow is essentially constant in a local neighbourhood of the pixel under examination and uses the least squares criterion to solve the basic optical flow equations for all pixels in that neighbourhood. [12]

The Lucas-Kanade method is used to estimate optical flow and using the Lucas-Kanade method, create an optical flow object for estimating the direction and speed of a moving object. To calculate the optical flow vectors, use the object function estimate flow. The optical flow object's internal state can be reset using the reset object function.[8-12]

```
v = VideoReader("C:\Users\Dell\Desktop\VIDEOLAR\VIDEO22.mp4")
```

```

opticFlow = opticalFlowLK('NoiseThreshold',0.002);

f = figure;
movegui(f);
View = uipanel(f,'Position',[0 0 2 2],'Title', 'opticalFlowLK ');
Plot = axes(View);

while hasFrame(v)
    frame = readFrame(v);
    Grayimage = im2gray(frame);
    flow = estimateFlow(opticFlow, Grayimage);
    imshow(frame)
    hold on
    plot(flow,'DecimationFactor',[6 6],'ScaleFactor',12,'Parent',Plot);
    hold off
    pause(0.2)
end

```

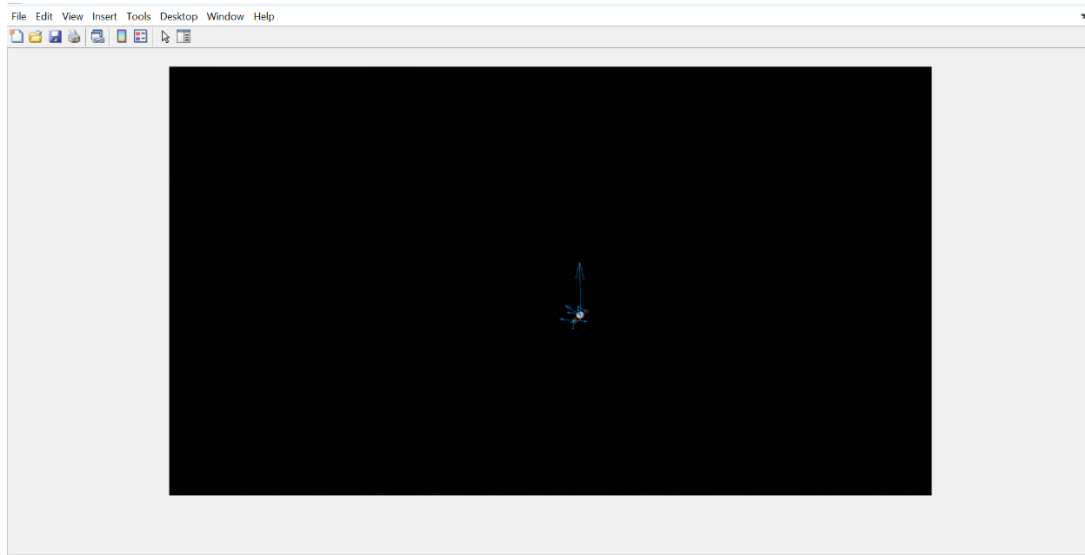


Figure 6.6. opticalFlowLK result

As you seen here optical flow LK method is used and target which move estimating while scanning parameters in video actually all processes are defined as ; The Lucas-Kanade approach. The concept of optical flow was proposed by Aires, Kelson R. T.; Santana, Andre M.; Medeiros, Adelardo A. D. [12-13]

The Lucas-Kanade approach is used to estimate optical flow and build an optical flow object for determining the direction and speed of a moving object using the Lucas-Kanade method. Use the object function estimate flow to compute the optical flow vectors. The internal state of the optical flow object can be reset using the reset object function and on matlab as applying optical flow LK is run to estimate and frame unit is coded in parameter below.

Firstly introduced `opticFlow = opticalFlowLK('NoiseThreshold',0.002);`

on matlab.

```

while hasFrame(v)
    frame = readFrame(v);
    Grayimage = im2gray(frame);
    flow = estimateFlow(opticFlow, Grayimage);
    imshow(frame)
    hold on
    plot(flow,'DecimationFactor',[6 6],'ScaleFactor',12,'Parent',Plot);

```

Algorithms

To determine the optical flow between two frames, we must solve the optical flow constraint equations listed below. It does not search the second image for a match for a specific pixel. It operates by attempting to guess the direction an object has moved in order to explain local changes in intensity.

$$I_x u + I_y v + I_t = 0 \tag{6.1}$$

The spatiotemporal image brightness derivatives are I_x , I_y , and I_t .

The horizontal optical flow is denoted by the letter U .

The vertical optical flow is denoted by the letter V .

The Lucas-Kanade method solves the u and v optical float constraint equations by dividing the unique image into smaller sections and assuming a uniform tempo in

each section. Then, for the stationary version of $[uv]^T$ for each bin, reproduce the optical floating point constraint equation's weighted least square color.

This suit is achieved by minimizing the expression.

$$\sum_{x \in \Omega} W^2 [I_x u + I_y v + I_t]^2 \quad (6.2)$$

W is a window feature that highlights the intermediate limits of each section. A method that reduces overhead is

According to Lucas and T. Kanade, this system is frequently over-determined because it has more equations than unknowns. To arrive at a compromise solution, the Lucas-Kanade technique applies the least squares principle.[12]

$$\begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_y I_x & \sum W^2 I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum W^2 I_x I_t \\ \sum W^2 I_y I_t \end{bmatrix} \quad (6.3)$$

It is computed using a difference filter in the Lucas-Kanade method, $[-1 \ 1]$.

The following is how u and v are solved:

- Use the kernel $[1 \ 8 \ 0 \ 8 \ 1]/12$ and its transposed form to compute I_x and I_y , and the $[11]$ kernel to compute I_t between images 1 and a couple of. Using a separable and isotropic five-by-five detail kernel with $[1 \ 4 \ 6 \ 4 \ 1]/16$ powerful 1-D coefficients, smooth the gradient components, I_x , I_y , and I_t .

Use the following method to solve the 2-by-2 linear equations for each pixel:

$$\text{If } A = \begin{bmatrix} a & b \\ b & c \end{bmatrix} = \begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_y I_x & \sum W^2 I_y^2 \end{bmatrix} \quad (6.4)$$

Lucas and T. Kanade studied, because this system contains more equations than unknowns, it is generally over-determined.

The Lucas-Kanade technique employs the least squares principle to arrive at a compromise answer.

Furthermore, A's eigenvalues are;

Lucas and T. Kanade found that this system often has too many equations compared to unknowns. To solve this, the Lucas-Kanade technique uses the least squares principle to find a compromise solution.

$$\lambda_i = \frac{a+c}{2} \pm \frac{\sqrt{4b^2 + (a-c)^2}}{2}; i = 1, 2 \quad (6.5)$$

The edge and the eigenvalues are compared. This is the cost of the noise reduction edge that you specified. When either of the following occurs, there is an impact:

Condition: $\lambda_1 \geq \tau$ and $\lambda_2 \geq \tau$

Because A is nonsingular, Cramer's rule is used to solve the system of equations.

Condition: $\lambda_1 \geq \tau$ and $\lambda_2 < \tau$

Because A is singular, the gradients associated with drift are normalized to compute u and v.

Condition : $\lambda_1 < \tau$ and $\lambda_2 < \tau$

The optical flow, u and v, are both zero.

Motion in real time of dark area can express as using optical flow on matlab ;

Code groups which is below are enable to estimate objects that move in dark area or normal area ,actually function of system is focused as motion and vector arrows track moved objects.In dark zone assump flying fighter plane and fire of behind exhaust of plane take place any motion by fire light those vector arrows track that light by using optical flow.Actually system only focus motion and predict.

```

vidDevice = imaq.VideoDevice('winvideo', 1, 'YUY2_1280x720', ...
'ReturnedColorSpace', 'rgb', ...
'DeviceProperties.Brightness', 64, ...
'DeviceProperties.Sharpness', 7);
nextId=1
nFrames = 0;
while (nFrames<2000)
frame = vidDevice();
opticFlow = opticalFlowLK('NoiseThreshold',0.002);
f = figure;
movegui(f);
View = uipanel(f,'Position',[0 0 2 2],'Title','Optical Flow ');
Plot = axes(View);
Grayimage = im2gray(frame);
flow = estimateFlow(opticFlow, Grayimage);
imshow(frame)
hold on
plot(flow,'DecimationFactor',[6 6],'ScaleFactor',12,'Parent',Plot);
hold off
pause(0.2)
end

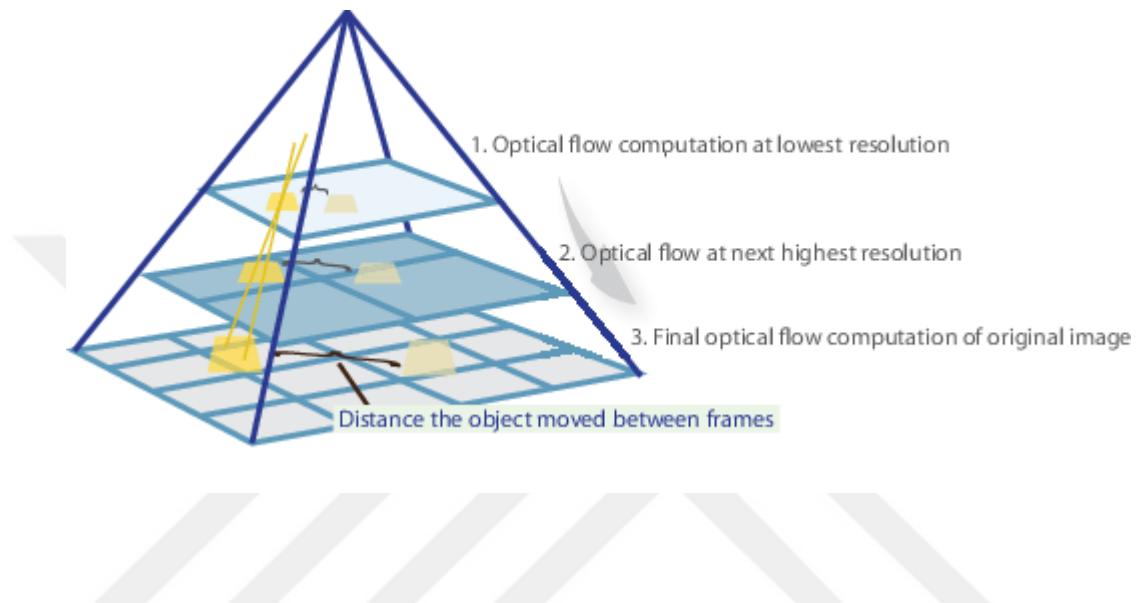
```

Optical flow farneback

In his paper two-frame mobility estimation based on polynomial expansion, Gunnar Farneback suggested an efficient method for estimating the mobility of important features by comparing two consecutive frames. Object for evaluating optical flow as the farneback method. In order to create an optical flow object to estimate the direction and velocity of a moving object using the farneback method and estimate the optical flow vector using the EstimateFlow objective function.

opticFlow = opticalFlowFarneback returns an optical flow object that can be used to estimate the direction and velocity of moving objects in a video.

Farneback's algorithm generates an image pyramid, with each level having a lower resolution than the one before it. By selecting a pyramid level greater than one, the algorithm can track points at multiple resolution levels, beginning with the lowest. The algorithm can handle larger point changes between frames by increasing the number of pyramid levels. Simultaneously, the number of calculations grows. This diagram depicts a three-level image pyramid.



Tracking starts at the lowest resolution and continues until convergence. Points found in one level are passed as keypoints to the next level. In this way, the algorithm refines the tracking at each level. Because of pyramid decomposition, the algorithm can handle large pixel motions. This distance could be greater than the size of the neighborhood.

```
v = VideoReader("C:\Users\Dell\Desktop\VIDEOLAR\VIDEO8.mp4")
```

```
opticFlow = opticalFlowFarneback
```

```
f = figure;
```

```
movegui(f);
```

```
View = uipanel(f,'Position',[0 0 2 2],'Title','opticalFlowFarneback ');  
Plot = axes(View);
```

```
while hasFrame(v)  
    frame = readFrame(v);  
    Grayimage = im2gray(frame);  
    flow = estimateFlow(opticFlow, Grayimage);  
    imshow(frame)  
    hold on  
    plot(flow,'DecimationFactor',[6 6],'ScaleFactor',12,'Parent',Plot);  
    hold off  
    pause(0.2)  
end
```

```
opticFlow =  
opticalFlowFarneback with properties:  
NumPyramidLevels: 3  
PyramidScale: 0.5000  
NumIterations: 3  
NeighborhoodSize: 5  
FilterSize: 15
```

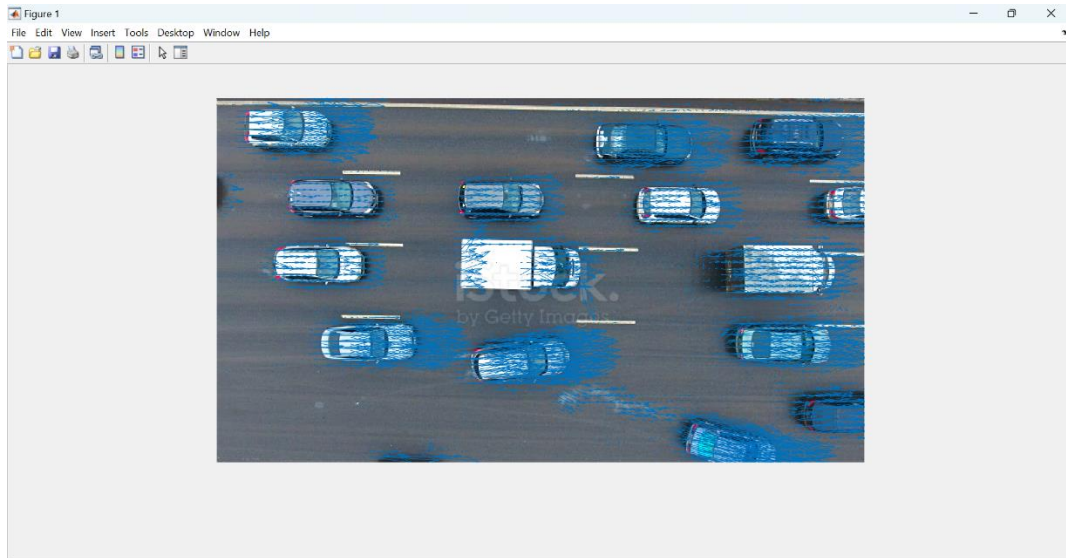


Figure 6.7. opticalFlowFarneback result

As you seen in the application here all objects which move estimate as utilizing optical flow farneback method this method defined as; object used to evaluate optical flow using the farneback method. To generate an optical flow object, use the farneback method to estimate the direction and velocity of a moving object and the EstimateFlow objective function to estimate the optical flow vector.

Optical flow HS

For specifying direction which move objects Kelson R. T. Aires studied on that method and to provide ImageFlow tries to smooth out distortions in the flow over the whole image. So it prefers solutions that show more smoothness. A classic optical current estimation algorithm is the Horn-Schunck method. This assumes that the current is constant across the image. As a result, they tend to promote more uniform solutions while minimizing current distortion. This problem employs a modern version of the Horn-Schunck method, which employs the rough-to-fine pyramid method, similar to the Lucas-Kanade method, for efficiency. This framework serves as the foundation for many current optical flow algorithms.

When it comes to overall energy production, targets are set and minimized.

Let the image be $p = (x,y)$ and the underlying flow field $w(p) = (u(p),v(p), 1)$.

where $u(p)$ and $v(p)$ are horizontal and vertical components of the flow field. Assuming constant luminance, the pixel values in the flow vector should be uniform, and the flow field should be piecewise uniform. This yields the objective function in the continuous spatial domain object for optical flow estimation using the Horn-Schunck method. To create an optical flow object, estimate the direction and velocity of a moving object using the Horn-Schunck method, and the optical flow vector using the EstimateFlow objective function.

```
v = VideoReader("C:\Users\Dell\Desktop\VIDEOLAR\VIDEO8.mp4")
```

```
opticFlow = opticalFlowHS
```

```
f = figure;
```

```
movegui(f);
```

```
View = uipanel(f,'Position',[0 0 2 2],'Title','opticalFlowFarneback ');
```

```
Plot = axes(View);
```

```
while hasFrame(v)
```

```
    frame = readFrame(v);
```

```
    Grayimage = im2gray(frame);
```

```
    flow = estimateFlow(opticFlow,Grayimage);
```

```
    imshow(frame)
```

```
    hold on
```

```
    plot(flow,'DecimationFactor',[6 6],'ScaleFactor',12,'Parent',Plot);
```

```
    hold off
```

```
    pause(0.2)
```

```
end
```

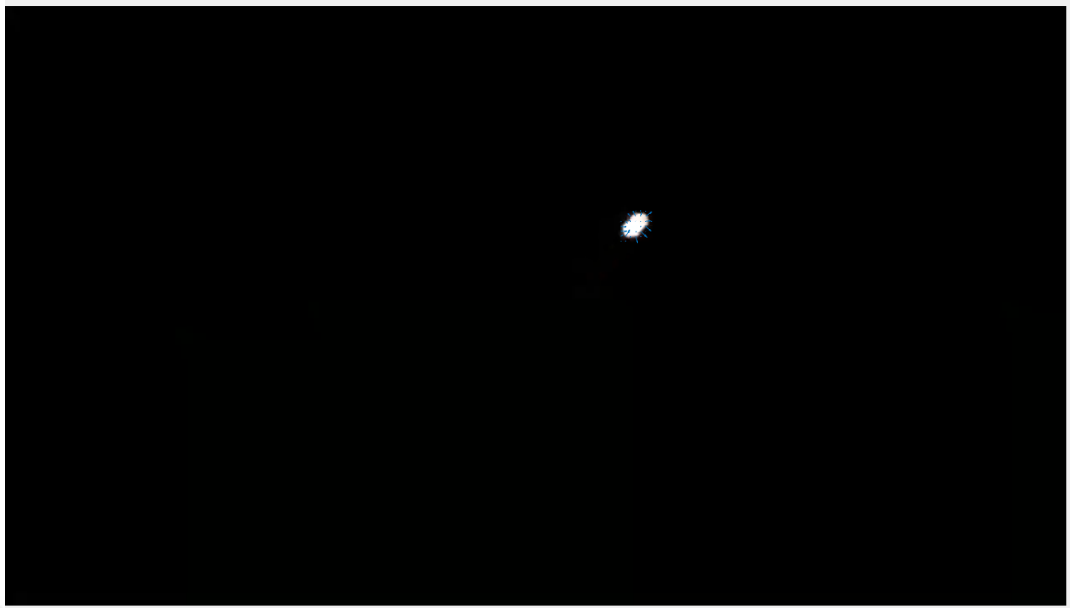


Figure 6.8. opticalFlowHS in dark area

As you seen in that figure, when optical flow HS method applied vectors are follow objects which move on video and this is specified as; The Horn-Schunck technique is a well-known optical current estimate methodology. This is based on the assumption that the current is constant across the picture. As a result, they tend to favor more consistent solutions while limiting current distortion. For efficiency, this problem employs a current version of the Horn-Schunck approach, which uses the rough-to-fine pyramid method, similar to the Lucas-Kanade method. Many modern optical flow algorithms are built on this framework.

in application on matlab firstly defined opticFlow = opticalFlowHS

and then frame unit is introduced ;

while hasFrame(v)

frame = readFrame(v);

Grayimage = im2gray(frame);

flow = estimateFlow(opticFlow,Grayimage);

imshow(frame)

in simulink block

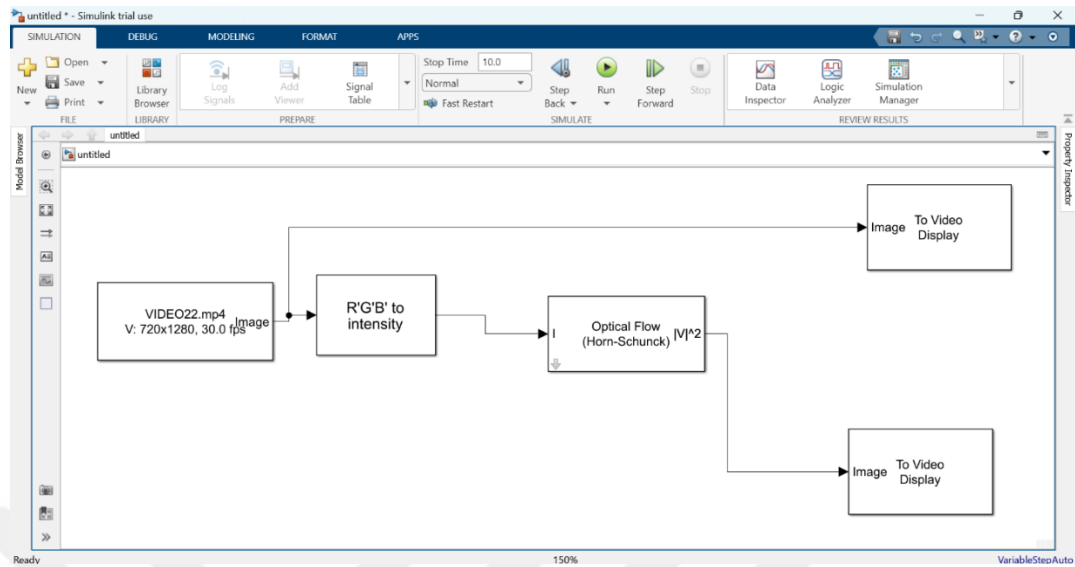


Figure 6.9. opticalFlow block diagram

In here by utilizing Simulink block optical flow HS method is used in order to estimate objects which move and you seen in dark area flying any fire fighter plane and fire that take place on behind of plane is tracked firstly scanning And then objects is followed by optical flow vector.

Original video, fighter plane in dark area

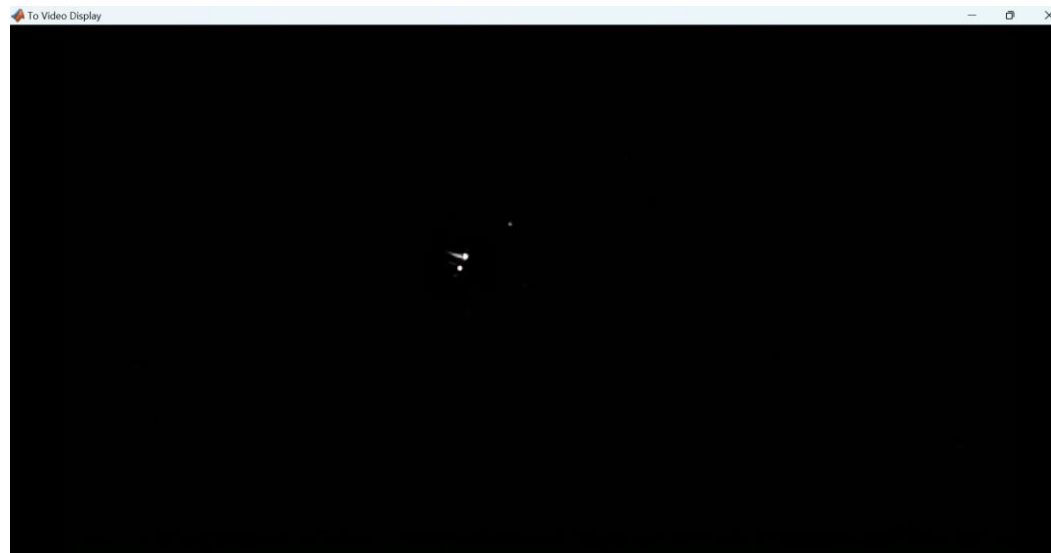


Figure 6.10. opticalFlow block diagram result

After process

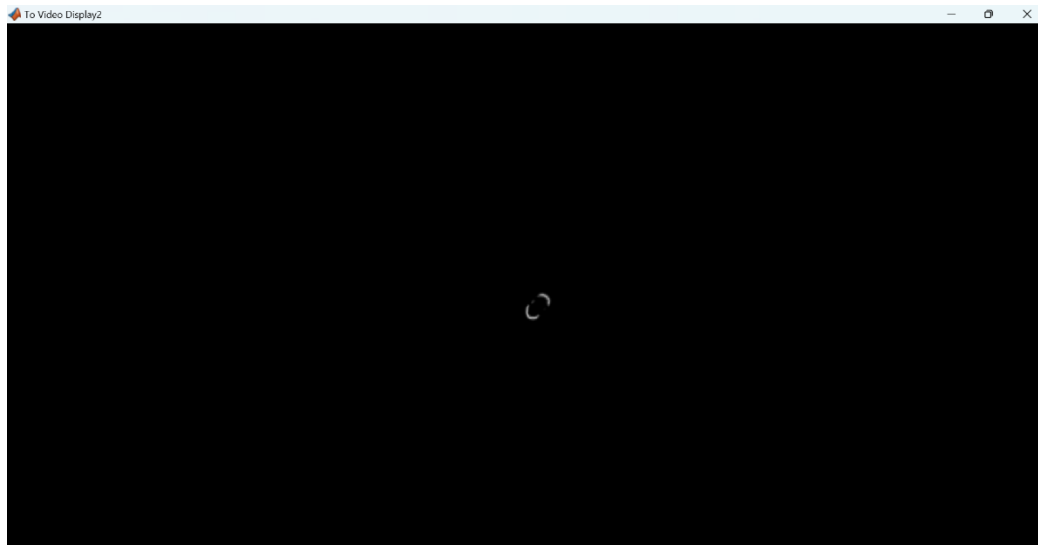


Figure 6.11. opticalFlow block diagram result

For autothreshold process

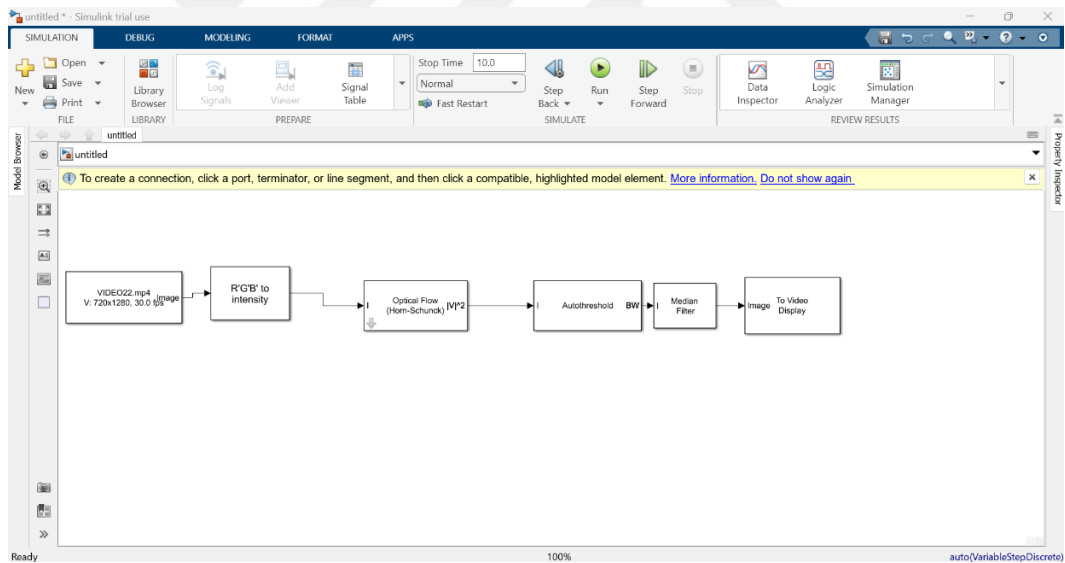


Figure 6.12. opticFlow block diagram result with autothreshold

After autothreshold process

Here as using autothreshold and optical flow HS method objects scanning and estimate targets that move.

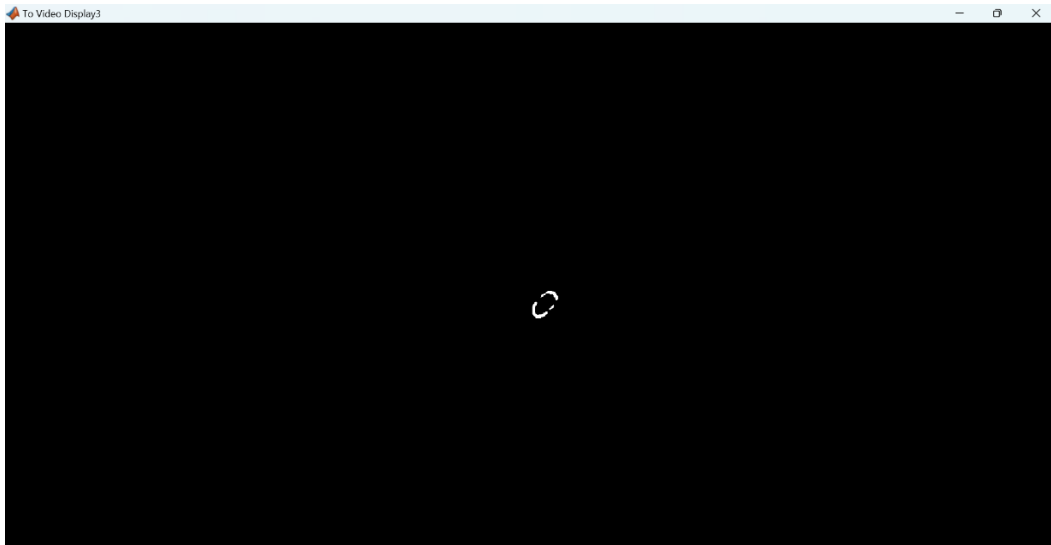


Figure 6.13. opticalFlow block diagram result with autothreshold and filtering

Compare the objects as some parameters such as optical flow,threshold,bbox

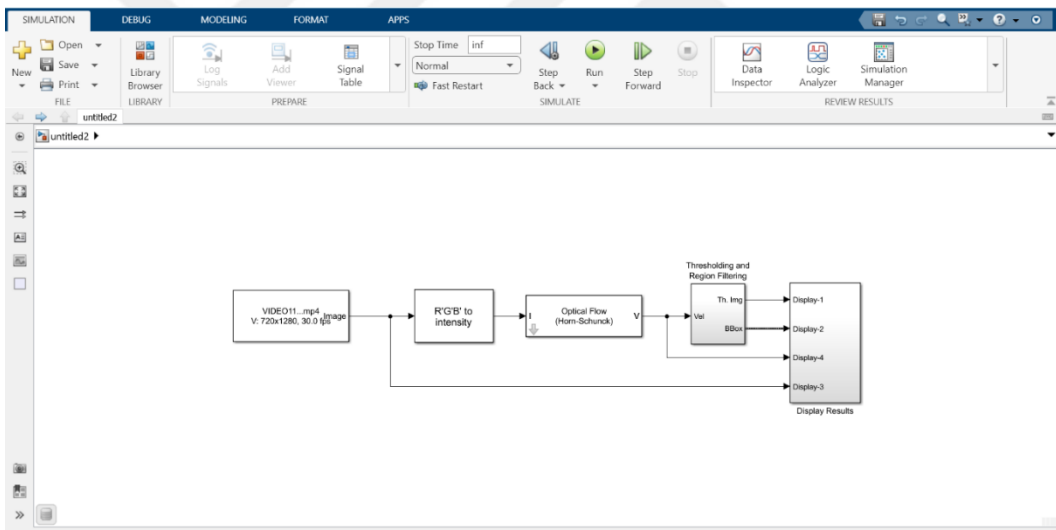


Figure 6.14. opticalFlow result with threshold and region filter

Here as utilizing matlab Simulink block,applied image process techniques such as threshold,region and filtering.

Thresholding a photograph is a simple yet effective way of separating it into foreground and background components. This technique to image analysis isolates items by converting grayscale photographs to binary images. Alasdair McAndrew studied on some applications.

After threshold,object tracking

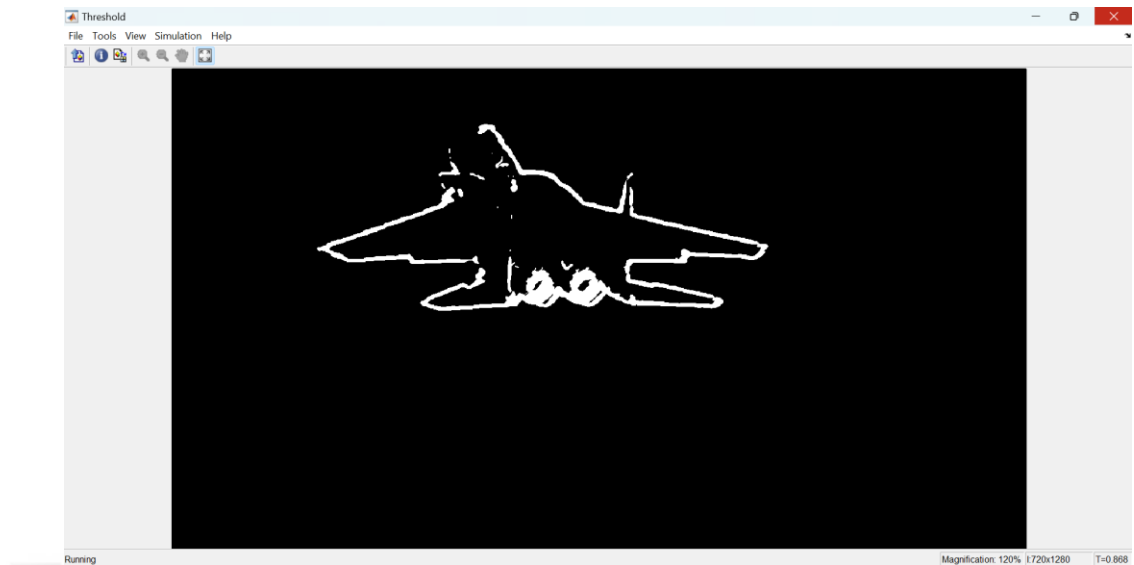


Figure 6.15. opticalFlow result with threshold and region filter result

Image regions have attributes such as area, center of mass, direction, and bounding box. They are also known as objects, linked components, or blobs. You may use the Image Region Analyzer app or the `regionprops` function to determine these attributes. Filtering is a method of changing or improving a picture. For example, you may filter an image to accentuate some aspects while removing others. Image processing procedures accomplished using filtering include smoothing, sharpening, and edge enhancement.

After `bbox` ,object tracking

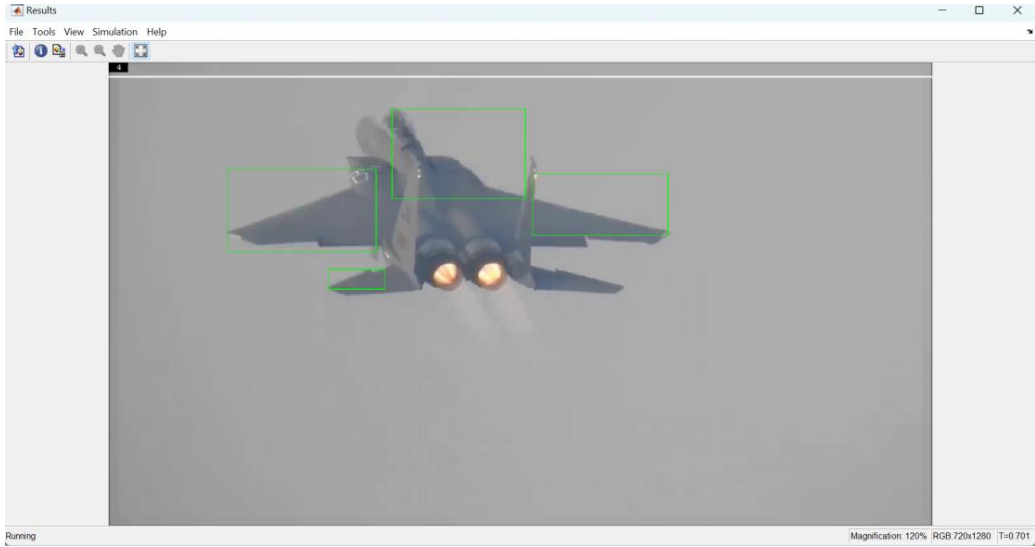


Figure 6.16. opticalFlow result with threshold and region filter

After optical flow

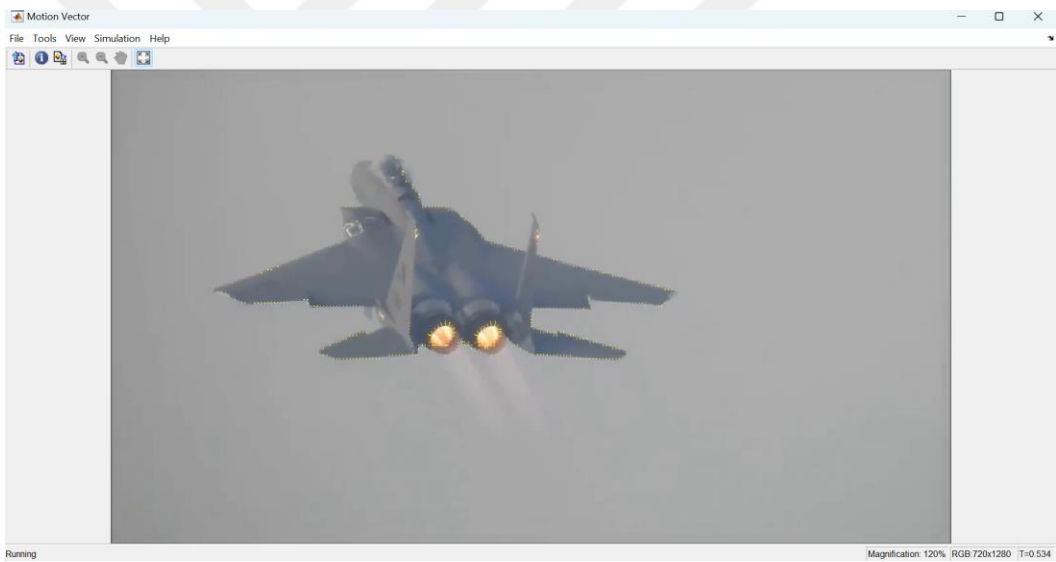


Figure 6.17. opticalFlow result with threshold and region filter

As a result we obtained object tracking as using optical flow block and other blocks such as threshold, bbox

Optical flow LKDoG

Lucas-Kanade Gaussian derivatives are used to estimate optical flow. The Lucas-Kanade derivative of the Gaussian (DoG) method is used to create optical flow objects by estimating the direction and speed of moving objects. Calculate the optical flow vector using the AssessmentFlow object function and use the object reset function to reset the internal state of the optical flow object. It is studied by Kelson R. T. Aires for specify direction which move objects.

Algorithms

You need to solve the following optical flow constraint equation to calculate the optical flow between two frames. According to Lucas try to guess the direction of motion of the object to account for local variations in intensity.[10]

$$I_x u + I_y v + I_t = 0 \quad (6.6)$$

I_x , I_y , and I_t are the brightness derivatives in the spatiotemporal image.

The letter U represents horizontal optical flow.

Vertical optical flow is represented by the letter V .

The Lucas-Kanade method and the derivation of the Gaussian filter are used for the calculation.

Use the following formula to solve the optical flow limiting equation for u and v :

1. Compute I_x and I_y using the steps below:

The Gaussian filter is used for temporal filtering. The NumFrames property defines the time filter parameters, such as B , the standard deviation and the number of filter coefficients.

Gaussian filter and its derivative are used for spatial filtering of the image. The ImageFilterSigma property lets you specify the standard deviation and length of the image smoothing filter.

2. Determine it by carrying out the operations

Use the derivative of a Gaussian filter to filter time. Use the NumFrames property to set filter timing specifications, including number of filter coefficients and standard deviation.

3. Apply the gradient smoothing filter defined in step 1b to the I_x , I_y , and I_t components of the gradient to spatially filter the temporal filter output. The GradientFilterSigma property specifies the number of gradient smoothing filter coefficients and the standard deviation.

4. Follow these steps to solve the 2-by-2 linear equations for each pixel.

$$\text{if } A = \begin{bmatrix} a & b \\ b & c \end{bmatrix} = \begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_y I_x & \sum W^2 I_y^2 \end{bmatrix}$$

$$\text{Then the eigenvalues of A are } \lambda_i = \frac{a+c}{2} \pm \frac{\sqrt{4b^2 + (a-c)^2}}{2}; i = 1, 2$$

(6.7)

When the algorithm finds the eigenvalues, it compares them to the threshold, which is the value you set for the NoiseThreshold property. The results fall into one of three categories:

Condition 1: $\lambda_1 \geq \tau$ and $\lambda_2 \geq \tau$

Because A is nonsingular, Cramer's rule is used by the algorithm to solve the system of equations.

Condition 2: $\lambda_1 \geq \tau$ and $\lambda_2 < \tau$

Because A is singular (noninvertible), the algorithm normalizes the gradient flow in order to compute u and v.

Condition 3: $\lambda_1 < \tau$ and $\lambda_2 < \tau$

The optical flow, u and v, are both zero.

In application of Optical flow LKDoG

v =

VideoReader with properties:

General Properties:

Name: 'VIDEO20.mp4'

Path: 'C:\Users\Dell\\Desktop\VIDEOLAR'

Duration: 63.3
CurrentTime: 0
NumFrames: <Calculating...> learn more
Video Properties:
Width: 1280
Height: 720
FrameRate: 30
BitsPerPixel: 24
VideoFormat: 'RGB24'

```
opticFlow =  
opticalFlowLKDoG with properties:  
  
NumFrames: 3  
ImageFilterSigma: 1.5  
GradientFilterSigma: 1  
NoiseThreshold: 0.0005  
v = VideoReader("C:\Users\Dell\Desktop\VIDEOLAR\VIDEO20.mp4")
```

```
opticFlow = opticalFlowLKDoG('NoiseThreshold',0.0007)
```

```
f = figure;  
movegui(f);  
View = uipanel(f,'Position',[0 0 2 2],'Title','opticalFlowFarneback ');  
Plot = axes(View);
```

```
while hasFrame(v)  
frame = readFrame(v);  
Grayimage = rgb2gray(frame);  
flow = estimateFlow(opticFlow,Grayimage);  
imshow(frame)  
hold on
```

```
plot(flow,'DecimationFactor',[6 6],'ScaleFactor',12,'Parent',Plot);
```

```

hold off
pause(0.2)
end

```



Figure 6.18. opticalFlowLKDoG in dark area

As you see figure above target which is moved is tracking sensing object by vector arrows also defined as ;

optical flow=opticalflowLKDoG('NoiseThreshold',0.0007) and estimate object by defining as ; specifying frame parameters while using some code groups

```

while hasFrame(v)
frame = readFrame(v);
Grayimage = rgb2gray(frame);
flow = estimateFlow(opticFlow,Grayimage);
imshow(frame)
hold on

```

therefore when running codes videos which is selected are opened and objects which move in video are estimated and capture targets actually firstly scanning all view and then estimate and focusing targets that move.As a result from opticalflowLKDoG property utilizes and estimate objects ,density of arrows on objects which move introduce all tracked targets.



Figure 6.19. opticalFlowLKDoG result

In this case, as you know, the property of optical flow is introduced, and the arrows determine the movement of the fire along the exhaust system behind the firefighter. Actually try to evaluate the movements, the arrows mean that here we are dealing with the vector optical flow.

The Lucas-Kanade Gaussian derivative (DoG) is used to create optical flow objects by estimating the direction and speed of moving objects and using the AssessmentFlow object function to calculate the optical flow vector and the function "Reset Object" is used to reset the internal state of the optical flow object.

Kalman Filter

The Kalman filter is used to track objects. This feature provides an easy way to customize your view. The Kalman filter object, which is an implementation of a linear discrete-time state space system, is used for tracking. The configuration view properties are determined by the Kalman filter function. The start location property corresponds to the measurement vector in the state space model of the Kalman filter. This table establishes the relationship between the dimension vector M and the state space model of the kalman filter. Paul Zarchan investigate the filter is named after Rudolf E. Kálmán, who was one of the primary developers of its theory. This digital filter is sometimes termed the Stratonovich–Kalman–Bucy filter because it is

a special case of a more general, nonlinear filter developed somewhat earlier by the Paul zarchan.[10]

Algorithm

State transition model and H dimension model

The state space model state transition model A and the dimension model H are specified as M block diagonal matrices of identical submatrices As and Hs:

According to Howard Musoff the dimension model H and the state space model state transition model A are given as M block diagonal matrices of identical submatrices. As and Hs, the estimate and its quality are determined by the system parameters and noise statistics provided into the estimator.[10]

$$A = \text{blkdiag}(As_1, As_2, \dots, As_M) \quad (6.8)$$

$$H = \text{blkdiag}(Hs_1, Hs_2, \dots, Hs_M) \quad (6.9)$$

The following describes the submatrices As and Hs:

MotionModel	As	Hs
'ConstantVelocity'	[1 1; 0 1]	[1 0]
'ConstantAcceleration'	[1 1 0.5; 0 1 1; 0 0 1]	[1 0 0]

The First State, x:

MotionModel	Initial state, x
'ConstantVelocity'	[InitialLocation(1), 0, ..., InitialLocation(M), 0]
'ConstantAcceleration'	[InitialLocation(1), 0, 0, ..., InitialLocation(M), 0, 0]

The error covariance matrix for initial state estimation, P:

$$P = \text{diag}(\text{repmat}(\text{InitialError}, [1, M]))$$

The noise covariance of the process, Q:

$$Q = \text{diag}(\text{repmat}(\text{MotionNoise}, [1, M]))$$

R is the measurement noise covariance.

```
R = diag(repmat(MeasurementNoise, [1, M])).
```

Matlab simulink block was used to track object in here I studied on code members actual aim is to track objects that move.

```
videoReader=vision.VideoFileReader("C:\Users\Dell\Desktop\VIDEOLAR\VIDEO11..mp4");
```

```
videoPlayer = vision.VideoPlayer('Position',[150,150,600,500]);  
foregroundDetector = vision.ForegroundDetector('NumTrainingFrames',15,...  
'InitialVariance',0.07);  
blobAnalyzer = vision.BlobAnalysis('AreaOutputPort',false,...  
'MinimumBlobArea',90);
```

```
kalmanFilter = []; isTrackInitialized = false;  
while ~isDone(videoReader)  
colorImage = step(videoReader);
```

```
foregroundMask = step(foregroundDetector, rgb2gray(colorImage));  
detectedLocation = step(blobAnalyzer,foregroundMask);  
isObjectDetected = size(detectedLocation, 1) > 0;
```

```
if ~isTrackInitialized  
if isObjectDetected  
kalmanFilter = configureKalmanFilter('ConstantAcceleration',...  
detectedLocation(1,:), [1 1 1]*1e6, [30, 15, 15], 30);  
isTrackInitialized = true;  
end  
label = ""; circle = zeros(0,3);  
else  
if isObjectDetected
```

```

predict(kalmanFilter);
trackedLocation = correct(kalmanFilter, detectedLocation(1,:));
label = 'Target';
else
trackedLocation = predict(kalmanFilter);
label = 'Predicted Target';
end
end
circle= [trackedLocation, 100];
end

colorImage = insertObjectAnnotation(colorImage,'circle',...
circle,label,'Color','blue');
step(videoPlayer,colorImage);
end

```



Figure 6.20. Kalman filter and blob analysis result

How to use the Kalman filter object tracking method in a Matlab moving target application to scan all views, evaluate moving objects, and lock on targets. The blob parsing method is coded first, and then the kalman filter object is traced.

Blob Analysis calculates statistics for labeled parts of a binary image. This block returns the centroid, bounding box, label matrix, and number of blobs. The Blob Analysis block accepts variable-size signals as input and output.

```
blobAnalyzer = vision.BlobAnalysis('AreaOutputPort',false,...  
'MinimumBlobArea',90);
```

```
kalmanFilter = []; isTrackInitialized = false;
```

```
while ~isDone(videoReader)
```

```
colorImage = step(videoReader);
```

```
results by using kalman filter as you seen in the figure 5.20 and 5.21
```

This feature provides an easy way to customize your view.

The Kalman filter object, which is an implementation of a linear discrete-time state space system, is used for tracking. The Kalman filter function defines the visual properties of a configuration. The dimension vector in the Kalman filter state space model is associated with the initial position attribute. This table shows the relationship between the dimension vector M and the state space model of the Kalman filter.

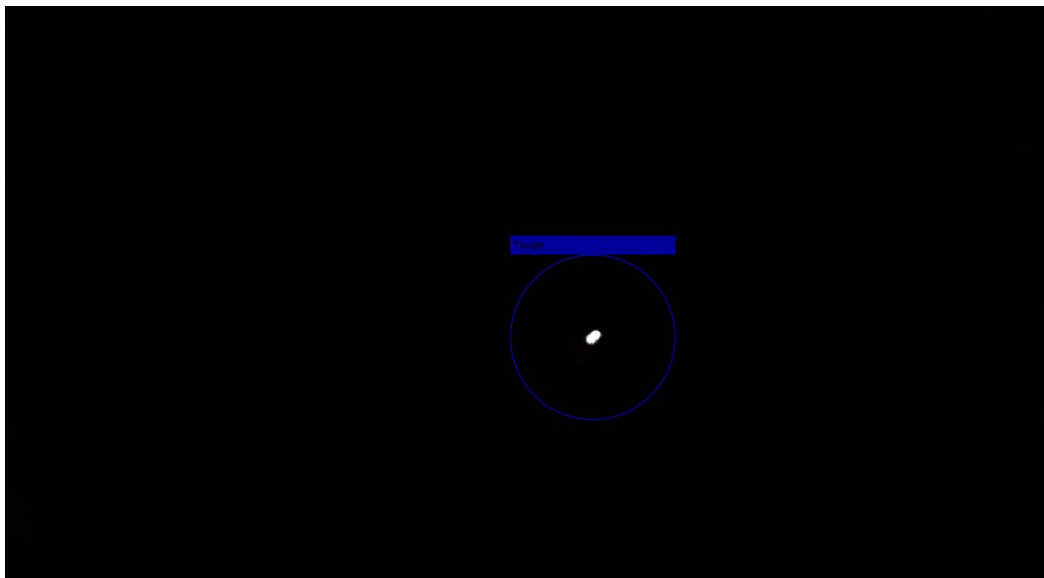


Figure 6.21. Kalman filter result in dark area

As you have seen, this character system scans the entire frame and focus the movement, which predicts the moving and capturing objects, so the weapon's firing

range is adjusted to find the object in the circle and the intended objects in the center of the fire the circle system weapon will detect and destroy the target. In the above condition, the fighter flies in a dark area, and the system focuses the firelight behind the aircraft's exhaust, thereby judging it based on the movement.



CHAPTER VII

GPS SENSOR USAGE WITH COORDINATES

7.1. GPS Sensor data with Plot on Matlab

Video processing in Matlab necessitates stream processing, which involves simultaneously processing video frames from a continuous stream.

A GPS sensor provide to take signals by using parameters such as truePosition, trueVelocity,LLA,ground speed, and fs values such as 1 and so on.[12]

The following parameters and charts are written, and the condition is controlled.

```
duration = 1000;
numSamples = duration*fs;
refLoc = [37.0288 35.8124 30];
truePosition = zeros(numSamples,3);
trueVelocity = zeros(numSamples,3);
gps = gpsSensor('UpdateRate',fs,'ReferenceLocation',refLoc);
position = gps(truePosition,trueVelocity);
t = (0:(numSamples-1))/fs;
subplot(3, 1, 1)
plot(t, position(:,1), ...
t, ones(numSamples)*refLoc(1))
title('GPS Sensor Readings')
ylabel('Latitude (degrees)')
subplot(3, 1, 2)
plot(t, position(:,2), ...
t, ones(numSamples)*refLoc(2))
ylabel('Longitude (degrees)')
```



Figure 7.1. GPS sensor

Here, used a gps sensor certainly placed this sensor on any factor and get statistics approximately region consequently display us 37.0288,35.8124 coordinate after which after gps ship data,satellite tv for pc digital digicam activate and start to hint that region and accordingly take snap photos which flow on floor certainly at unique region arise tracing and tracking. When the usage of GPS, 4 satellites are required due to the fact the magic variety is 4 because of the manner GPS calculates your unique function. Each satellite tv for pc transmits a sign that consists of 1) the best time of transmission and 2) the best function of the satellite tv for pc relative to the middle of the Earth. Also covered are frequency periods L1, 1575.forty two MHz, and L2, 1227.6 MHz. And GPS antenna is a tool that gets and amplifies radio alerts transmitted with the aid of using GNSS satellites on precise frequencies and converts them to an digital sign to be used with the aid of using a GNSS or GPS receiver.

```
subplot(3, 1, 3)
plot(t, position(:,3), ...
t, ones(numSamples)*refLoc(3))
ylabel('Altitude (m)')
xlabel('Time (s)')
```

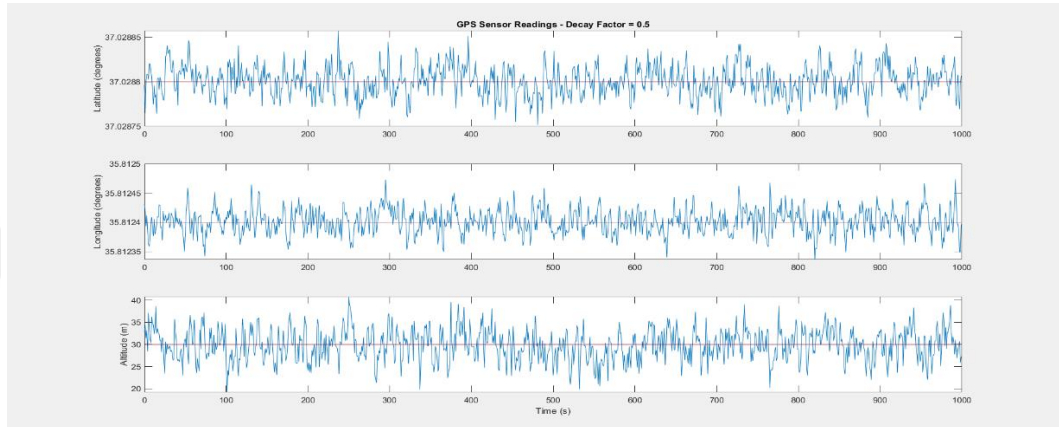


Figure 7.2. Gps reading result

Here gps sensor data results are used to see coordinates where it is.

CHAPTER VIII

CREATION OF MODEL OF ABONDING BOUNDARY

8.1.Model of abandoning boundary of target on ground

To track and estimate objects abandoned object detection method is used in here simulink model was obtained and objects tracked. Yamaguchi studied object detection has emerged as a hot subject in video surveillance and estimated to motion.

8.1.1.Subsystem for storing background data

The background for this mockup is the first frame of the video. The example uses intensity and color information to subtract the background to improve accuracy. The Cb and Cr color channels are stored in a complex array during this operation. When developing a professional surveillance system, a more sophisticated segmentation algorithm should be used.

8.1.2.Subsystem Detected

The main algorithm resides in the detection subsystem. The luminance segmentation and color segmentation subsystems in this subsystem perform background subtraction using intensity and color data. Using the binary OR operator, the example combines two segmentation results. The blob parsing block calculates statistics for scene objects.

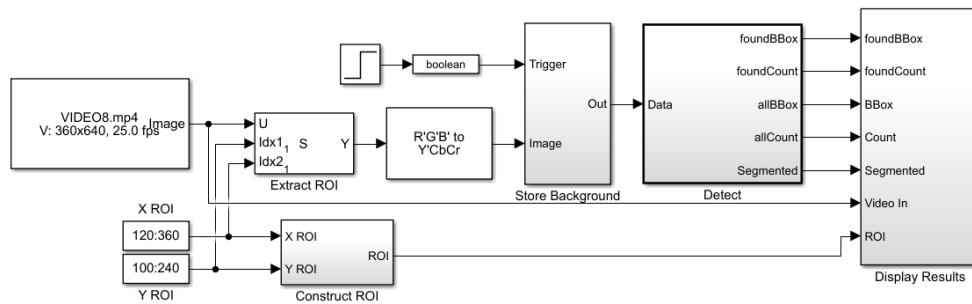


Figure 8.1. Abandoning boundary model

Here target detection and evaluation is introduced using both target acquisition and boundary detection methods and image processing techniques, and then frames are formed, objects are moved to the side and tracked.

bboxes = Detection uses the Input Aggregate Channel Object Detector (ACF) to detect elements within the image. The detected item positions are returned as a set of bounding boxes.

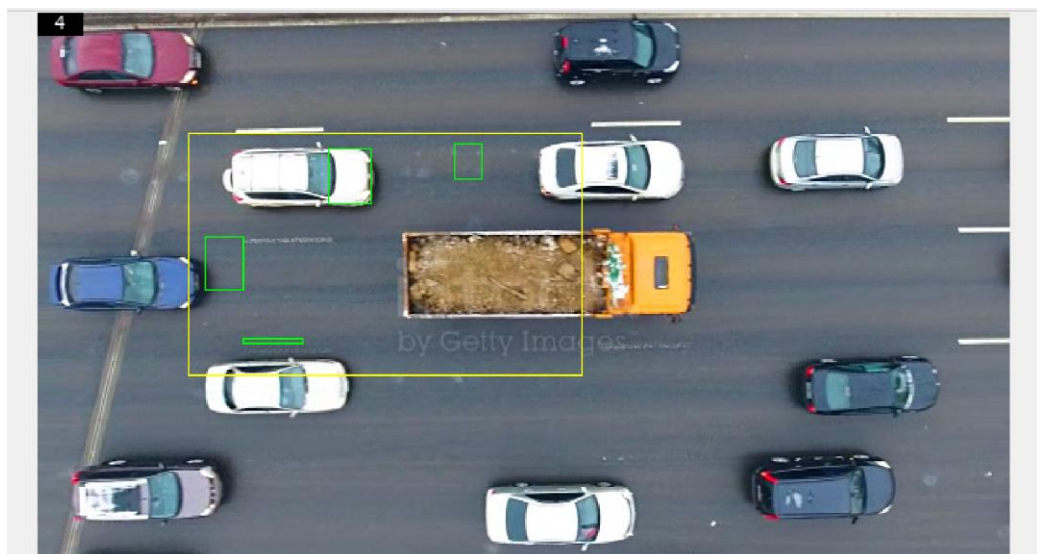


Figure 8.2. Abandoning boundary model result

The Threshold window displays the outcome of the ROI(the region of interest) background subtraction,here Simulink model was used.



Figure 8.3. Abandoning boundary model result

CHAPTER IX

SOME CODES AS TAKING SNAP VIEW FROM SPACE

9.1. View and determination of the boundary from space



Figure 9.1. View of the border from space

Figure 9.2. View of the border from space

We can manipulate the mechanisms of the guns and song the route and function of the goal in addition to the hearthplace stages of the guns for twenty-four hours while we take a photo on a stay video. Some instructions can help us in finding the goal and figuring out the route wherein it's far transferring.

Even if we handiest use a satellite tv for pc digital digicam device to ship photo data, a laser variety locating device and a GPS monitoring device installed on every weapon can use a particular region, a particular region at the ground.

When we use satellite tv for pc imagery, we're searching down from the sky at a particular region. Weapon hearthplace stages goal at the peak and duration of transferring targets, and the goal distance is measured with a laser rangefinder, with a few munitions who prefer to ruin handiest a part of the goal.



Figure 9.3. View of the border from space



Figure 9.4. View and determination of the boundary from space

processDetections - read the image processing detections that have been recorded.

This command will only extract the functions listed below:

1. Vehicle coordinates in two dimensions [x,y] array

2. Width: The object's width as reported by the video sensor.
3. Classification: the report's object classification

```

vidDevice = imaq.VideoDevice('winvideo', 1, 'YUY2_1280x720', ...
    'ReturnedColorSpace', 'rgb', ...
    'DeviceProperties.Brightness', 64, ...
    'DeviceProperties.Sharpness', 7);

nextId=1
nFrames = 0;
while (nFrames<2000)
frame = vidDevice();
frame=T
data = getsnapshot(T);
data_0 = getsnapshot(T);
data_0 = rgb2gray(data_0);
data_1 = getsnapshot(T);
data_1 = rgb2gray(data_1);
data_2 = imabsdiff(data_0,data_1);
data_2 = medfilt2(data_2, [3 3]);
data_2= im2bw(data_2,0.05);
data_2 = bwareaopen(data_2,320);
pause(0.1);
Red_01 =data( :, : ,1);
Green_02 =data(:, :,2);
Blue_03=data( :, :,3);
Filt_1 = 16+(0.2567890625 * Red_01)+ (0.50412890625 * Green_02) +(
0.09790625 * Blue_03);
Filt_2= 128+(-0.14822265625 * Red_01)-(-0.2909921875 * Green_02) +
(0.43921484375* Blue_03);
Filt_3 = 128+(0.43921484375 * Red_01)- (0.3677890625 * Green_02) -(
0.07142578125 * Blue_03);
pause(0.01);
[R3r R3c]= find((Filt_1>Filt_2) & (Filt_2>Filt_3) );
ruleIIIpixel=size(R3r);

```

```

Ir3= uint8(zeros(size(data)));
bk = bwlabel(data_2, 8);
dedector_1 = regionprops(bk, 'BoundingBox', 'Centroid','Area');
handles.output = hObject;
axes(handles.axes3);
imshow(data_2);
handles.output = hObject;
axes(handles.axes1);
imshow(data_0);
set(handles.text8,'BackgroundColor','blue');
set(handles.OperatorText,'String','Scanning');
for object = 1:length(dedector_1)
    bk = dedector_1(object).BoundingBox;
    yk = dedector_1(object).Centroid;
    ak = dedector_1(object).Area;
    rectangle('Position',bk,'EdgeColor','g','LineWidth',2)
    plot(yk(1),yk(2), '-m+')

    ba = stats(object).BoundingBox;
    bd = stats(object).Centroid;
    ary=stats(object).Area;
    rectangle('Position',bb,'EdgeColor','r','LineWidth',2)
    plot(bd(1),bd(2), '-m+')
    for i4=1:ruleIIIpixel-1
        Ir3(R3r(i4),R3c(i4),1)=Red_01(R3r(i4),R3c(i4));Ir3(R3r(i4),R3c(i4),2)
        =Green_02(R3r(i4),R3c(i4));
        Ir3(R3r(i4),R3c(i4),3) =Blue_03(R3r(i4),R3c(i4));
        i4=i4+1;
        B_1= im2bw(Ir3,0.9);
        B_1 = bwareaopen(B_1,300);
        C_1 = bwlabel(B_1, 8);
        dedector_1 = regionprops(C_1, 'BoundingBox', 'Centroid','Area');
        C_1 = dedector_1(object).BoundingBox;
        yk = dedector_1(object).Centroid;

```

```

ak = dedector _1(object).Area;
rectangle('Position',C_1,'EdgeColor','r','LineWidth',2)
plot(yk(1),yk(2), '-m+')
Matlab detection and prediction code members were used[12]

```



Figure 9.5. War plane with motion estimation

Here,dedector codes activite targets which fly on sky are tracking and so labels run after that capture object and track.

Both object tracking and fire tracking are carrying out for those code group such as;label,dedector,bounding thus code function process in any video real or normal video.

ImageFileNames: {'C:\Users\DeIII\Desktop\f16-atesleme.jpg'}

Datastore: []

SelectedImageIdx: 1

IsDataBlockedImage: 0

IsChanged: 0

FileName: 'C:\Users\DeIII\Desktop\imageLabelingSession.mat'

PixelLabelDataPath: []

TempDirectory: []

ROILabelSet: [1×1 vision.internal.labeler.ROILabelSet]

ROISublabelSet: [1×1 vision.internal.labeler.ROISublabelSet]

ROIAttributeSet: [1×1 vision.internal.labeler.ROIAttributeSet]
FrameLabelSet: [1×1 vision.internal.labeler.FrameLabelSet]
ROIAnnotations: [1×1 vision.internal.labeler.ROIAnnotationSet]
FrameAnnotations: [1×1 vision.internal.labeler.FrameAnnotationSet]
AlgorithmInstances: {}
ShowROILabelMode: 'hover'
HasROILabels: 1
NumROILabels: 1
HasFrameLabels: 0
NumFrameLabels: 0
NumROIsublabels: 0
NumAttributes: 0



Figure 9.6. War plane labelled target[12]

Here matlab image labelled application was used and objects were labelled.

Here the jet engine exhaust flame is monitored with some codes and also if the flame appears suddenly, the detection system and the active markers as you see in the picture above.

In matlab application some code members were used to track some colour

Fire label is tracking object that is moving as that code group;

```
Red_01 =data( :, : ,1);
```

```

Green_02 =data(:,2);
Blue_03=data(:,3);
Filt_1 = 16+(0.2567890625 * Red_01)+ (0.50412890625 * Green_02) +(
0.09790625 * Blue_03);
Filt_2= 128+(-0.14822265625 * Red_01)-(-0.2909921875 * Green_02) +
(0.43921484375* Blue_03);
Filt_3 = 128+(0.43921484375 * Red_01)- (0.3677890625 * Green_02) -(
0.07142578125 * Blue_03);
pause(0.01);
[R3r R3c]= find((Filt_1>Filt_2) & (Filt_2>Filt_3) );
ruleIIIpixel=size(R3r);
Ir3= uint8(zeros(size(data)));
bk = bwlabel(data_2, 8);
dedector _1 = regionprops(bk, 'BoundingBox', 'Centroid','Area');
handles.output = hObject;
axes(handles.axes3);
imshow(data_2);
handles.output = hObject;
axes(handles.axes1);
imshow(data_0);
set(handles.text8,'BackgroundColor','blue');
set(handles.OperatorText,'String','Scanning');

for i4=1:ruleIIIpixel-1
Ir3(R3r(i4),R3c(i4),1) =Red_01(R3r(i4),R3c(i4));
Ir3(R3r(i4),R3c(i4),2) =Green_02(R3r(i4),R3c(i4));
Ir3(R3r(i4),R3c(i4),3) =Blue_03(R3r(i4),R3c(i4));
i4=i4+1;
B_1= im2bw(Ir3,0.9);
B_1 = bwareaopen(B_1,300);
C_1 = bwlabel(B_1, 8);
dedector _1 = regionprops(C_1, 'BoundingBox', 'Centroid','Area');
C_1 = dedector _1(object).BoundingBox;
yk = dedector _1(object).Centroid;

```

```
ak = dedector _1(object).Area;  
rectangle('Position',C_1,'EdgeColor','r','LineWidth',2)  
plot(yk(1),yk(2), '-m+')
```

Humans, vehicles, animals, and drones moving on the ground or in the air are detected and tracked in Matlab using this method, allowing weapon targets to be tracked while moving object data is written to the media. By recording data, you can determine the range of weapons and continuously track targets. This technology is widely used by the military and can be used to protect national borders.



CHAPTER X

CONCLUSION AND RECOMMENDATIONS

10.1.Recommendations,Conclusion and Future Work

The goal of this thesis is to arrange military technology by utilizing matlab software and some electronic aspects. For this study, Used some methods such as image process techniques and motion estimation techniques in matlab application ,in here firstly took data as using Gps and satellite actually satellite and Gps provide us coordinates and location of targets which moving or stationary and then satellite camera is tracked those zones and getting snap and after getting snap some processes are applied like threshold, and objects revealed and become more prominent therefore targets are tracked and capture other ones methods are motion estimation is to detect objects and find their direction and where they arrive. Result of this study shows us as utilizing some methods military weapon firing ranges adjusting. There are a few recommendations to improve that study you can use various software in order to specify more where targets are and their coordinates and locations. Firstly we must introduce artificial intelligence, humans make object recognition appear simple. We can quickly recognize items in our environment, regardless of their situations, such as being upside down, different in color or texture, partially obscured, and so on. Even things that appear in a variety of shapes or are vulnerable to significant shape variations, such as any target, may be easily identified. Our brain has generalized to one type of item. A fundamental issue in identifying a shape is that it can be in any position, rotated, mirrored, and so on. These transformations are classified as affine transformations, which encompass any transformation that preserves distance ratios as well as co-linearity. These alterations can soon lead to difficulty identifying the item for a neural system. The present generation of robots is not yet capable of thinking about the particulars of its environment. However, as this subject advances, contact with the environment will become an increasingly important element of study. However, for the time being, only rudimentary object recognition is employed in some circumstances Our system first processes images with preprocessing steps to extract entities that can be used for recognition by a neural system. This is done

because the system for recognition we are proposing mainly relies on the fact that an object can be decomposed into many shapes that can be extracted from an image. These forms will then serve as the foundation for our adaptive system. The descriptions are produced once the forms have been retrieved. This is yet another critical problem that will be investigated in this thesis. These descriptors are designed to be invariant representations that include as much critical shape information as feasible. They are designed in such a way that they may be fed into a neural network and easily classified. The adaptive system is made up of feed-forward neural networks as well as a self-organizing map (SOM). The system is trained in multiple stages, both supervised and unsupervised. Our adaptive technique, like other adaptive systems, may still be separated into two phases. The first step is the learning phase, in which the system processes training data and adjusts to its particular. The second stage is recognition. When presented with images that it has never seen before, the system's recognition capabilities are put to the test. Thus after using artificial intelligence recognize objects and classify them and because of that ,introduce easily objects.I have a few recommendation more,in future when use other software to track objects which moving and stationary ,absolutely it should be noted that there is a wider range and it should be noted that the objects introduced to artificial intelligence are perceived more precisely and get accurate results. In fact, when artificial intelligence is used, all the data obtained should be compared with other programs used, whether they are perceived more accurately in the application area or not, and which of them gives more accurate results.

REFERENCES

- [1] Handbook of Satellite Applications Joseph N Pelton, Scott Madry, Sergio Camacho-Lara Published in 2017, ISBN: 9783319233864
- [2] Understanding the GPS-introduction to Global Positioning System ,Gregory T.French-,published by Georesearch inc. in 1996, ISBN: 0-9655723-O-7
- [3] NAVSTAR GPS Space Segment/Navigation User Segment Interfaces,Fyfe,Peter Kovach,Karl
- [4] Satellite Communications Dennis Roddy, Published by McGraw-Hill Professional, in 2001,ISBN : 9780071371766
- [5] Satellite Communications Systems: Design Principles ,M.Richharia,ISBN: 978-0070523746, Published by McGraw-Hill Professional Publishing; First Edition in 1995
- [6] Introduction to Global Position System 2nd Edition, Ahmed El-Rabbany, Published by Artech House Publishers in 2006,ISBN: 978-1596930162
- [7] An Introduction to Digital Image Processing with Matlab-Alasdair McAndrew, Published by Brooks/Cole in 2004,ISBN : 978-0534400118
- [8] Determining optical flow,article,Horn, Berthold K.P.; Schunck, Brian G. (August 1981).
- [9] Digital Image Processing ,Rafael Gonzalez and Richard E. Woods.. Addison-Wesley, 2nd edition, 2002. ISBN 0-201-18075-8 Library of Congress Cataloging-in-Publication Data,Gonzalez,Rafael C.Digital Image Processing
- [10] Fundamentals of Kalman Filtering: A Practical Approach, Paul Zarchan; Howard Musoff, Published by American Institute of Aeronautics and Astronautics in 2000,ISBN:1563474557
- [11] Algorithms for Image Processing and Computer Vision ,James R. Parker. Published by Wiley Publishing,Inc,in 2011 ,ISBN: 978-0-470-64385-3,
- [12] Optical Flow Using Color Information,Aires, Kelson R. T.; Santana, Andre M.; Medeiros, Adelardo A. D. 2008.

[13] Matlab Applications

[14] Vehicle ego-motion estimation and moving object detection using a monocular camera ,Yamaguchi, K., Kato, T., and Ninomiya, Y.. Published in: 18th International Conference on Pattern Recognition in 2006,ISBN:0-7695-2521-0

[15] Digital Image Processing,Rafael C. Gonzalez Published by pearson in 2018,ISBN: 978-1292223049



CIRRICULUM VITAE(CV)

Name SURNAME :Tarık SEV

Title :Mechanical Engineer

Current Institution: Gaziantep University Natural & Applied Science

Degree	Department	University	Year
Bachelor Degree	Mechanical Engineering	Çukurova University	2016
Master Degree	Mechanical Engineering	Gaziantep University	2023