PEOPLE’S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

Saad Dahlab Blida 1 University

Institute of Aeronautics and Space Studies IAES

Option: Avionics

Theme:

« Swarm of drones for forest fire detection»

Dissertation

Submitted in partial fulfillment of the requirements for Master Degree in Aeronautical engineering and the attainment of the diploma "a startup" / "a patent" under Ministerial Decree

No. 1275

Presented by:	**Supervised by:**	**Instructors:**
1- Chegrani Akram	**Supervisor: Dr. Choutri Kheireddine**	**President : Dr. Zabot Amar**
2- Yahiaoui Mohamed	**Co- Supervisor : Prof. Lagha Mohand**	**Examiner : Dr. Bousaad Azmedroub**

Academic Year 2022/2023

Information card

About the supervision team

Of the working group

Information card

About the supervision team of the working group

Main Group Supervision Team	SPECIALTY	FACULTY	INSTITUTION
Supervisor 01	Avionics	Aeronautics	Blida 1
Supervisor 02	installation	Aeronautics	Blida 1

About the supervision team of the working group

About the project team of the working group	SPECIALTY	FACULTY	INSTITUTION
Student 01	Avionics	Aeronautics	Blida 1
Student 02	Avionics	Aeronautics	Blida 1

ملخص

في عصر يتسم بالتقدم التكنولوجي غير المسبوق، أدى دمج التعلم الآلي، والرؤية الحاسوبية، والمركبات الجوية بدون طيار إلى عصر جديد من الإمكانيات عبر مختلف المجالات. لقد تجاوز استخدام والمركبات الجوية بدون طيار، المعروفة باسم الطائرات بدون طيار، عالم الأدوات الترفيهية وظهر كأداة تحويلية في مجالات تتراوح من المراقبة والزراعة إلى إدارة الكوارث. تبدأ أطروحة الماستر هذه في رحلة إلى تقاطع هذه التقنيات المتطورة، بهدف معالجة قضية ملحة في عصرنا: الكشف عن حرائق الغابات وإدارتها. وفي هذا السياق، تسعى العديد من الأنظمة المقترحة إلى تعزيز الكشف عن حرائق الغابات باستخدام الطائرات بدون طيار. تهدف أطروحة الماستر هذه إلى مواصلة تطوير هذه الأنظمة من خلال تسخير إمكانات أسراب الطائرات بدون طيار ومعالجة التحديات التي يفرضها الاستخدام المحدود للطائرات بدون طيار في الهواء الطلق في بلدنا. ويسعى البحث إلى إنشاء نظام تحديد المواقع الداخلي ليحل محل نظام تحديد المواقع العالمي (GPS)، مما يسهل عملية التطوير داخل البيئات الداخلية المغلقة. بالإضافة إلى ذلك، تبدأ الأطروحة محاولة إنشاء تشكيلة من الطائرات بدون طيار مكون من طائرتين قائد ومتابع، وهو ما يمثل خطوة محورية نحو تحقيق حلول الكشف عن حرائق الغابات وإدارتها القائمة على السرب.

الكلمات المفتاحية: الطائرات بدون طيار، التعلم الآلي، الكشف عن حرائق الغابات، رؤية الكمبيوتر، سرب الطائرات بدون طيار، نظام تحديد المواقع، تقدير المسافة، معايرة الكاميرا.

Résumé

À une époque marquée par des avancées technologiques sans précédent, l’intégration de l’apprentissage automatique, de la vision par ordinateur et des véhicules aériens sans pilote a marqué le début d’une nouvelle ère de possibilités dans divers domaines. L’utilisation des véhicules aériens sans pilote, communément appelés drones, a transcendé le domaine des gadgets récréatifs et est devenue un outil de transformation dans des domaines allant de la surveillance et de l’agriculture à la gestion des catastrophes. Ce mémoire de master s'embarque dans un voyage à l'intersection de ces technologies de pointe, visant à répondre à un problème urgent de notre époque : la détection et la gestion des incendies de forêt. Dans ce contexte, divers systèmes proposés cherchent à améliorer la détection des incendies de forêt à l’aide de drones. Ce mémoire de master vise à développer davantage ces systèmes en exploitant le potentiel des essaims de drones et en relevant les défis posés par l'utilisation restreinte des drones en extérieur dans notre pays. La recherche vise à créer un système de positionnement intérieur pour remplacer l'estimation de position basée sur le GPS, facilitant ainsi le processus de développement dans les environnements intérieurs. De plus, la thèse initie une tentative d'établir une formation de drones leader-suiveur à deux drones, ce qui représente une étape cruciale vers la réalisation de solutions de détection et de gestion des incendies de forêt basées sur des essaims.

Mots clés : UAV, Apprentissage automatique, détection des incendies de forêt, vision par ordinateur, essaim de drones, système de positionnement, estimation de la distance, calibrage de la caméra,yolo,positionnement intérieur,essaim

Abstract

In an era marked by unprecedented technological advancements, the integration of machine learning, computer vision, and unmanned aerial vehicles has ushered in a new era of possibilities across various domains. The utilization of UAVs, commonly known as drones, has transcended the realm of recreational gadgets and has emerged as a transformative tool in fields ranging from surveillance and agriculture to disaster management. This work embarks on a journey into the intersection of these cutting-edge technologies, aiming to address a pressing issue of our times: wildfire detection and management. Within this context, various proposed systems seek to enhance wildfire detection using drones. This work aims to further develop these systems by harnessing the potential of drone swarms and addressing the challenges posed by restricted outdoor drone utilization in our country. The research endeavors to create an indoor positioning system to replace GPS-based position estimation, facilitating the development process within indoor environments. Additionally, the thesis initiates an attempt to establish a two-drone leader-follower drone formation,representing a pivotal step towards the realization of swarm-based wildfire detection and management solutions.

Keywords: UAV, Machine learning, wildfire detection, Computer vision, swarm of drones, positioning system, distance estimation, camera calibration,yolo,indoor positioning,swarm

.

Acknowledgement

In the name of Allah, the Most Gracious and the Most Merciful. Above all, I express my gratitude to Almighty Allah for providing me with the strength, knowledge, capability, and opportunity to embark on this study and successfully bring it to fruition.

I wish to convey my heartfelt appreciation to my advisors, Dr.Choutri kheireddine and Prof. Lagha Mohand, for their invaluable guidance,endless patience and unwavering support . Their expertise,encouragement and mentorship have been instrumental in the successful completion of this dissertation.

I extend my deep gratitude to my beloved family for their constant support and keen interest in my academic achievements. I also cannot forget the dedication and support of my parents, who have always been by my side.

I would like to express my sincere appreciation to my dear friends,Their presence made this academic endeavor not only manageable but also enjoyable.

Acknowledgement

First and foremost, I would like to praise Allah the Almighty, the Most Gracious, and the Most Merciful for His blessing given to me during my study and in completing this thesis. May Allah's blessing goes to His final Prophet Muhammad (peace be up on him), his family and his companions.

I would like to express my sincere gratitude to my advisors, Dr. Choutri Kheireddine and Prof. Lagha Mohand, for their invaluable guidance, unwavering support, and endless patience throughout the completion of this dissertation. Their expertise, encouragement, and mentorship have played a pivotal role in ensuring the successful culmination of this research endeavor.

Finally i would also like to thank my friends and family for their love and support during this process. Without them, this journey would not have been possible.

Contents

Abstract **3**

Contents **3**

List of Figures **8**

1 State of the art **11**

1.1 Introduction . . . 11

1.2 Background and context . . . 11

1.2.1 Brief overview of wildfires and their impact . . . 11

1.2.2 The role of early fire detection in wildfire management . . . 13

1.2.3 Introduction to UAV technology . . . 13

1.3 Literature review . . . 15

1.3.1 Use of visual, infrared and thermal cameras in forest fire . . . 15

1.3.1.1 Visual-based systems . . . 15

1.3.1.2 Fire detection with infrared images . . . 17

1.3.1.3 Fusion of visual and IR images . . . 19

1.4 Machine learning and computer vision . . . 19

1.4.1 Introduction . . . 19

1.4.2 Computer vision . . . 20

1.4.3 Machine learning . . . 21

1.4.4 Supervised machine learning . . . 23

1.4.5 Convolutional neural network . . . 23

1.4.6 Object detection . . . 25

1.4.6.1 Classification of object detection algorithms . . . 26

1.4.6.2 Common object detection network model . . . 26

1.4.6.3 Evaluation metrics for object detectors . . . 28

2 Indoor positioning system **30**

2.1 Introduction . . . 30
2.2 System architecture . . . 30
2.3 Methodology . . . 32
2.3.1 Configuration . . . 32
2.3.2 Position estimation . . . 34
2.3.2.1 From pixel to film reference . . . 35
2.3.2.2 From film to camera reference . . . 36
2.3.2.3 From camera to world reference . . . 36
2.4 Depth estimation . . . 38
2.4.1 Generalities . . . 39
2.4.1.1 The pinhole camera . . . 39
2.4.1.2 The normalized camera . . . 40
2.4.1.3 Focal length parameters . . . 41
2.4.1.4 Offset and skew parameters . . . 43
2.4.1.5 Position and orientation of camera . . . 43
2.4.1.6 Full pinhole camera model . . . 44
2.4.1.7 Radial distortion . . . 45
2.4.1.8 Camera calibration . . . 47
2.4.1.9 Calibration patterns . . . 47
2.4.1.10 Planar calibration patterns . . . 48
2.4.1.11 Stereo calibration . . . 48
2.4.1.12 Image rectification . . . 48
2.4.2 Method 1 with disparity . . . 48
2.4.3 Method 2 with triangulation . . . 50
2.5 Experimental setup . . . 52
2.5.1 Initialization . . . 57
2.5.2 Calibration process . . . 57
2.5.3 Binocular image correction . . . 58
2.5.4 Post-processing images . . . 59
2.5.5 Performing object detection . . . 59
2.5.6 Measuring the distance with disparity . . . 59
2.5.7 Depth estimation . . . 60
2.5.8 Position estimation . . . 60
2.5.9 Display and send results . . . 60
2.6 System evaluation . . . 61
2.7 Discussion . . . 69
2.8 Conclusion . . . 70

3 Multi UAV formation control **71**
3.1 Introduction . . . 71
3.2 Related work . . . 71
3.3 Multi-drone system architecture . . . 72
3.3.1 Centralized communication architecture . . . 73
3.3.2 Decentralized communication architecture . . . 74
3.4 Initial approach proposed . . . 76
3.4.1 Architecture . . . 76
3.4.2 Drones used . . . 77
3.4.2.1 Pixhawk drone . . . 77
3.4.2.2 Naza drone . . . 78
3.4.2.3 Parrot mambo . . . 79
3.4.3 Drone detection . . . 80
3.4.4 Leader follower algorithm . . . 82
3.5 Conclusion . . . 85

4 Fire detection and localization **87**
4.1 Introduction . . . 87
4.2 Related work . . . 88
4.2.1 UAVs for forest fire detection and monitoring . . . 88
4.2.2 Stereo vision-based fire localization system . . . 88
4.2.3 Vision based automatic forest fire detection techniques . . . 89
4.3 Methodology . . . 89
4.3.1 System description . . . 89
4.3.2 Data collection . . . 90
4.3.3 Yolo architecture . . . 91
4.3.3.1 Yolov4 . . . 92
4.3.3.2 Yolov5 . . . 92
4.3.3.3 Yolov8 . . . 93
4.3.3.4 Yolov-NAS . . . 93
4.3.4 Model training . . . 93
4.3.5 Comprehensive comparison of yolo models . . . 95
4.4 Position estimation . . . 96
4.5 Conclusion . . . 97

General Conclusion **98**

Further work **100**

Bibliography **101**

List of abbreviations

AI	Artificial intelligence
ANN	Artificial neural network
AP	Average Accuracy
CNN	Convolutional Neural Network
DL	Deep learning
FOV	field of view
FPV	First Person View
GCS	ground controle station
GPS	Global Positionning System
IMU	Inertial Measurement Units
IPS	indor positions system
IR	Infrared
MAP	Mean Average Accuracy
ML	machine learning
MPCS	Mission planning and control stations
RGB	red, green and blue
ROS	rate of spread
RTSP	Real Time Streaming Protocol
RPV	Remotely piloted vehicles
SPOF	Point of Failure Single
UAV	Unmmaned Arial Vehicle
YOLO	You Only Look Once
R-CNN	Region-Based Convolutional Neural Network

List of Figures

1.1 Global distribution of active fires from January to December 2019, as detected by MODIS. Source NASA [1] 12
1.2 Generic UAV system [2] 14
1.3 Offline video forest fire monitoring and detection methodologies using visual cameras[3]. 16
1.4 Fire monitoring, detection, and fighting methodologies using visual and infrared cameras[4]. 17
1.5 Vision-based techniques for forest fire monitoring, detection, and fighting using UAVs in near-operational field[3]. 18
1.6 Relationship between AI, ML, and DL[5]. 20
1.7 Categories of Machine Learning Algorithms Separated by Problems they Address[6]. 22
1.8 Supervised learning overview[7] 23
1.9 Structure of a Convolutional Neural Network (CNN)[7]. 25
1.10 The development history of the two-stage object detection network framework[8]. 26
1.11 The development history of the one-stage object detection network framework[8]. 27
1.12 The architecture of the R-CNN framework[8]. 28
1.13 internal architecture of the yolo object detector[9] 28

2.1 System operating diagram 31
2.2 indoor positioning system algorithm diagram 32
2.3 camera's vision 33
2.4 cameras alignment 34
2.5 backward camera projection 35
2.6 difference between film and pixel coordinates 35
2.7 perspective projection 36
2.8 representation of both camera and world references 37
2.9 The pinhole camera model[10] 39
2.10 Pinhole camera model terminology[10] 41

2.11 Normalized camera[10] 42
2.12 Focal length and photoreceptor spacing[10] 44
2.13 Radial distortion[10] 46
2.14 Camera calibration setup[11] 47
2.15 A rectified stereo pair[11] 49
2.16 Stereo vision basics [12] 50
2.17 triangulation representation 51
2.18 estimate angles with camera 52
2.19 an image of the stereo camera pair 53
2.20 align cameras using tier video streams 56
2.21 extrinsic and intrinsic camera parameters 57
2.22 Matlab stereo camera calibrator 58
2.23 capture of the positioning system output 61
2.24 Taking measures to evaluate the system 62
2.25 error in depth estimation 63
2.26 error in position estimation X 64
2.27 error in position estimation Y 64
2.28 error in position estimation Z 65
2.29 3d points of position estimation on XY plane 66
2.30 3d points of position estimation on XZ plane 67
2.31 3d points of position estimation on YZ plane 68
2.32 Depth and disparity are inversely related, so precise depth measurement is restricted to nearby objects 70

3.1 Schematic depicting the centralized communication architecture 73
3.2 Schematic showing a single-group swarm Ad hoc network 75
3.3 Intra-swarm communication architecture: (a) ring architecture, (b) star architecture,(c) meshed architecture 76
3.4 centralized leader-follower proposed architecture 77
3.5 image of Parrot Pixhawk drone 78
3.6 image of Parrot Naza drone 79
3.7 image of Parrot Mambo mini drone 79
3.8 example of the images acquired for training 80
3.9 example of the labeled images 81
3.10 training results of the YOLOv5 model of our custom dataset 82
3.11 illustration of the decision zones relative to the image received from follower drone 83
3.12 illustration of the decision zones relative to reference height y 84

3.13 Diagram of the follower tracking algorithm 85

4.1 Fire detection and geo-localization scheme 90
4.2 Image labeling session 91
4.3 Example of confidence score diversity 94

Chapter 1

State of the art

1.1 Introduction

In recent years, the integration of computer vision and machine learning has revolutionized the field of Unmanned Aerial Vehicle (UAV) based fire detection. This innovative synergy between advanced technologies has ushered in a new era of wildfire monitoring and management. By combining the capabilities of computer vision to analyze visual data with the power of machine learning algorithms to make sense of this information, UAVs have become invaluable tools in the early detection and rapid response to wildfires.

This chapter sets the stage for a comprehensive exploration of computer vision, machine learning and UAV-based fire detection, elucidating the theoretical underpinnings and existing research in this domain.

1.2 Background and context

1.2.1 Brief overview of wildfires and their impact

Forest fire is one of the major global environmental issues, causing havoc in places as disparate as cold Siberia, tropical Amazon, and the temperate HKH region (Fig. 1.1). Recent mega fires in Australia, Brazil, the United States, Greece, and Indonesia have not only destroyed ecosystems but have also triggered climate change through carbon emission. A rise in global temperature by 2 °C has contributed to increased frequency of forest fire, though only 3 % of all forest fires have been caused naturally

the majority of them have been sparked off by anthropogenic activities.
HKH, known as the Third Pole, or the Water Tower, is likely to face an increase in the frequency of forest fires as it is a region sensitive and vulnerable to climate change. The region is currently experiencing an annual increase in temperature by 0.03–0.07 °C. Climate change has had an impact on the increasing number of hot and dry days, thereby aggravating the risk of fire. Also, climate change and forest fire are interrelated and can have an impact on both—while forest fires can cause climate change through carbon emission, an increase in temperature due to climate change can cause forest fires. Forest fires are a major driver in the destruction of biodiversity and habitats of many endangered species in the region and a key factor in environmental transformation by the infusion of substantial amount of greenhouse gas. Besides, vegetation cover and its moisture content are two of the major factors that significantly influence forest fire as it holds fuel. Areas with dry and dense vegetation are more prone to forest fires than those with moist and sparse vegetation (Stevens et al. 2020). It has also got to be noted that forest fires are beneficial for

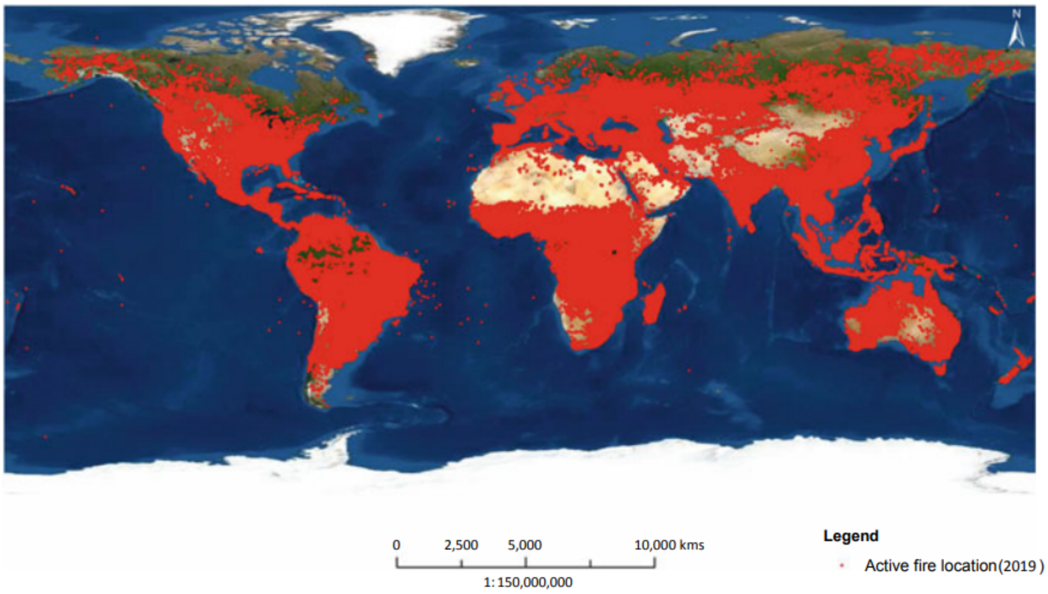


Figure 1.1: Global distribution of active fires from January to December 2019, as detected by MODIS. Source NASA [1]

nutrient recycling and vegetative succession, but an increase in their intensity and frequency could lead to desiccation and death of trees.[1]

1.2.2 The role of early fire detection in wildfire management

Continuous monitoring of open space is of the utmost importance for the protection of forests against fire. Collected data in real time provide fast intervention of relevant services to extinguish the fire. Timely information about the appearance of fire reduce the number of areas affected by this fire and thereby minimizes the costs of fire extinguishing and the damage caused in the woods.

Forest fires usually occur in areas remote from populated places, so that their detection at an early stage and timely reports to the competent services are of extreme importance. Early fire detection reduces the extinction time [6], requires fewer executors and fire-fighting equipment, thus increasing the efficiency and reducing the damage to the lowest possible level. Due to the importance of forest ecosystems, the goal is to prevent forest fires at an early stage.[13]

1.2.3 Introduction to UAV technology

There are three kinds of aircraft, excluding missiles, that fly without pilots. They are unmanned aerial vehicles (UAVs), remotely piloted vehicles (RPVs), and drones. All, of course, are unmanned so the name “unmanned aerial vehicle” or UAV can be thought of as the generic title. Some people use the terms RPV and UAV interchangeably, but to the purist the “remotely piloted vehicle” is piloted or steered (controlled) from a remotely located position so an RPV is always a UAV, but a UAV, which may perform autonomous or preprogrammed missions, need not always be an RPV.

In the past, these aircraft were all called drones, that is, a “pilotless airplane controlled by radio signals,” according to Webster’s Dictionary. Today the UAV developer and user community does not use the term drone except for vehicles that have limited flexibility for accomplishing sophisticated missions and fly in a persistently dull, monotonous, and indifferent manner, such as a target drone. This has not prevented the press and the general public from adopting the word drone as a convenient, if technically incorrect, general term for UAVs. Thus, even the most sophisticated air vehicle with extensive semiautonomous functions is likely to be headlined as a “drone” in the morning paper or on the evening news.

Whether the UAV is controlled manually or via a preprogrammed navigation system, it should not necessarily be thought of as having to be “flown,” that is, controlled by someone that has piloting skills. UAVs used by the military usually have autopilots and navigation systems that maintain attitude, altitude, and ground track automatically.

the vehicle and assume control when the desired course is reached. Navigation systems of various types (global positioning system (GPS), radio, inertial) allow for preprogrammed missions, which may or may not be overridden manually. As a minimum, a typical UAV system is composed of air vehicles, one or more ground control station (GCS) and/or mission planning and control stations (MPCS), payload, and data link. In addition, many systems include launch and recovery subsystems, air-vehicle carriers, and other ground handling and maintenance equipment. A very simple generic UAV system is shown in Figure 1.2 [2]

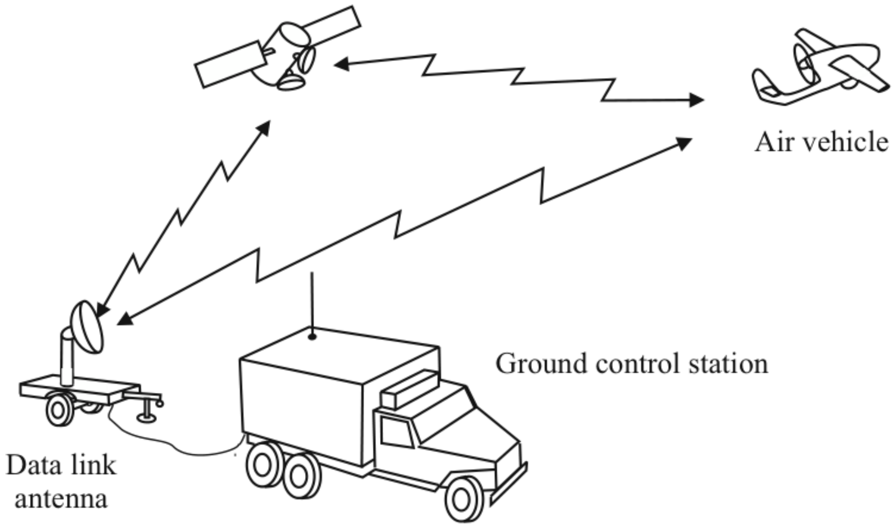


Figure 1.2: Generic UAV system [2]

1.3 Literature review

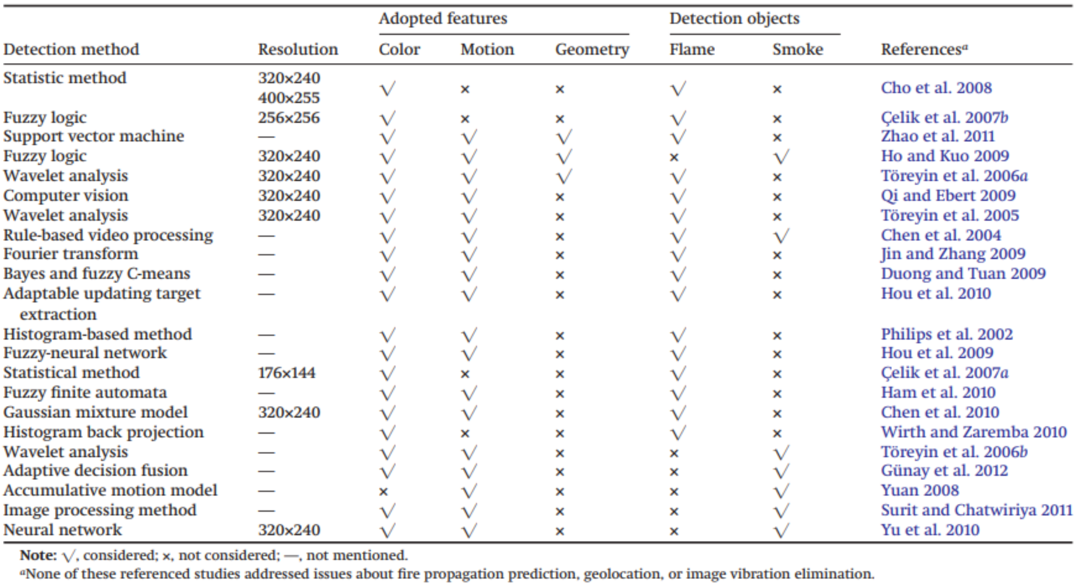
1.3.1 Use of visual, infrared and thermal cameras in forest fire

Traditional forest fire detection methods using watchtowers and human observers usually involve extensive labour forces and potentially threaten personnel safeties. Meanwhile, searching and observing forest fires solely depending on human is a dangerous and time-consuming activity. In order to tackle these disadvantages, more advanced automatic forest fire detection methods are developed using satellites, ground-based equipment's, and manned/unmanned aerial vehicles (UAVs). As a promising substitution of traditional fire detection approach serving as powerful tools for operational fire fighting, the integration of UAVs with remote sensing techniques has been attracted worldwide attention. Increasing research activities have accordingly been conducted for UAV-based firefighting applications in recent years [14]

1.3.1.1 Visual-based systems

Over the last decade, image processing techniques have become widely used for forest fire detection. Based on the spectral range of the camera used, vision-based fire detection technologies can generally be classified as either visual or infrared fire detection systems . Fire detection can be divided into either flame detection or smoke detection in terms of the actual object being detected. Most importantly, the color, motion, and geometry of the fire constitute the three dominant characteristic features of fire detection.

In Tables 1.3 and 1.4, color, motion, and geometry of the detected fire are commonly used in the existing investigation, with color being used mostly in segmenting the fire areas (Rudz et al. 2009; Mahdipour and Dadkhah 2012). As outlined in Table 1.3, considerable effort has gone into the development of offline video-based fire detection. Chen et al. (2004) use color and motion features based on an RGB (red, green, blue) model to extract real fire (flame) and smoke in video sequences. Töreyin et al. (2005) propose a real-time algorithm combining motion and color clues with fire flicker analysis on wavelet domain to detect fire in video sequences. Töreyin et al. (2006a) combine a generic color model based on RGB color space, motion information, and Markov process enhanced fire flicker analysis to create an overall fire detection system. Later, the same fire detection strategy is employed to detect possible smoke samples, which is used as an early alarm for fire detection (Töreyin et al. 2006b). In Çelik and Demirel (2009), a rule-based generic color model for flame pixel classification is presented, with experimental results showing significant improvement in detection performance. In Günay et al. (2009), an approach based on four sub-algorithms for wildfire detection at night is addressed and an adaptive active fusion method is adopted to linearly combine decisions from sub-algorithms. Finally, Günay et al. (2012) develop an entropy-functional-based online adaptive decision fusion framework, the application of which is to detect the presence of wildfires in video.

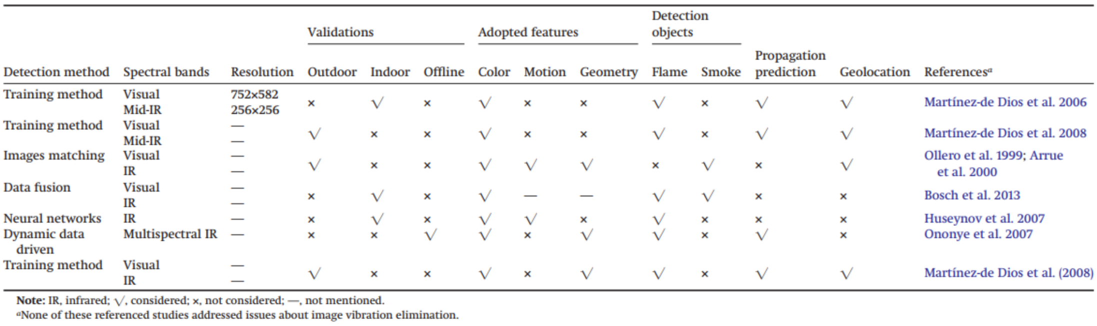


		Adopted features			Detection objects		
Detection method	Resolution	Color	Motion	Geometry	Flame	Smoke	References[a]
Statistic method	320×240 400×255	√	×	×	√	×	Cho et al. 2008
Fuzzy logic	256×256	√	×	×	√	×	Çelik et al. 2007*b*
Support vector machine	—	√	√	√	√	×	Zhao et al. 2011
Fuzzy logic	320×240	√	√	√	×	√	Ho and Kuo 2009
Wavelet analysis	320×240	√	√	√	√	×	Töreyin et al. 2006*a*
Computer vision	320×240	√	√	×	√	×	Qi and Ebert 2009
Wavelet analysis	320×240	√	√	×	√	×	Töreyin et al. 2005
Rule-based video processing	—	√	√	×	√	√	Chen et al. 2004
Fourier transform	—	√	√	×	√	×	Jin and Zhang 2009
Bayes and fuzzy C-means	—	√	√	×	√	×	Duong and Tuan 2009
Adaptable updating target extraction	—	√	√	×	√	×	Hou et al. 2010
Histogram-based method	—	√	√	×	√	×	Philips et al. 2002
Fuzzy-neural network	—	√	√	×	√	×	Hou et al. 2009
Statistical method	176×144	√	×	×	√	×	Çelik et al. 2007*a*
Fuzzy finite automata	—	√	√	×	√	×	Ham et al. 2010
Gaussian mixture model	320×240	√	√	×	√	×	Chen et al. 2010
Histogram back projection	—	√	×	×	√	×	Wirth and Zaremba 2010
Wavelet analysis	—	√	√	×	×	√	Töreyin et al. 2006*b*
Adaptive decision fusion	—	√	√	×	×	√	Günay et al. 2012
Accumulative motion model	—	×	√	×	×	√	Yuan 2008
Image processing method	—	√	√	×	×	√	Surit and Chatwiriya 2011
Neural network	320×240	√	√	×	×	√	Yu et al. 2010

Note: √, considered; ×, not considered; —, not mentioned.
[a] None of these referenced studies addressed issues about fire propagation prediction, geolocation, or image vibration elimination.

Figure 1.3: Offline video forest fire monitoring and detection methodologies using visual cameras[3].

Most of the above-mentioned research focuses on the detection of forest fires by flame, whereas smoke is also an important feature in the early and precise detection of forest fires. Tables 1.3 and 1.4 show some studies that focus on smoke detection. Chen et al. (2006) continue their work from Chen et al. (2004), proposing a chromaticity-based static decision rule and a diffusion-based dynamic characteristic decision rule for smoke pixel judgment. Experimental results indicate that this approach can provide an authentic and cost-effective solution for smoke detection. In Yu et al. (2009), a real-time smoke classification method using texture analysis is developed, and a back-propagation neural network is adopted as a discriminating model.



			Validations			Adopted features			Detection objects				
Detection method	Spectral bands	Resolution	Outdoor	Indoor	Offline	Color	Motion	Geometry	Flame	Smoke	Propagation prediction	Geolocation	References[a]
Training method	Visual Mid-IR	752×582 256×256	×	√	×	√	×	×	√	×	√	√	Martínez-de Dios et al. 2006
Training method	Visual Mid-IR	— —	√	×	×	√	×	×	√	×	√	√	Martínez-de Dios et al. 2008
Images matching	Visual IR	— —	√	×	×	√	√	√	×	√	×	√	Ollero et al. 1999; Arrue et al. 2000
Data fusion	Visual IR	— —	×	√	×	√	—	—	√	√	×	×	Bosch et al. 2013
Neural networks	IR	—	×	√	×	√	√	×	√	×	×	×	Huseynov et al. 2007
Dynamic data driven	Multispectral IR	—	×	×	√	√	×	√	√	×	√	×	Ononye et al. 2007
Training method	Visual IR	— —	√	×	×	√	×	√	√	×	√	√	Martínez-de Dios et al. (2008)

Note: IR, infrared; √, considered; ×, not considered; —, not mentioned.
[a]None of these referenced studies addressed issues about image vibration elimination.

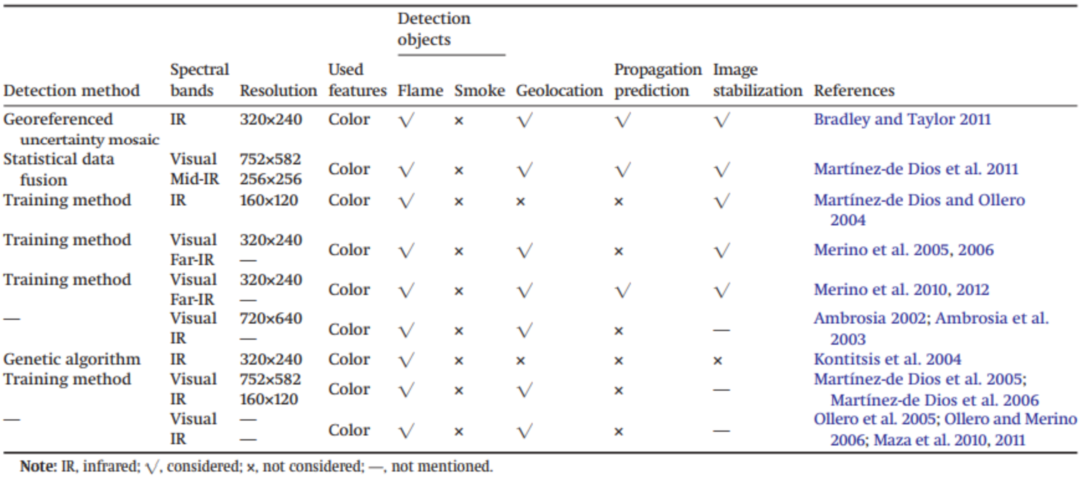
Figure 1.4: Fire monitoring, detection, and fighting methodologies using visual and infrared cameras[4].

Experiments prove that the proposed method is capable of differentiating smoke and non-smoke videos with both a quick fire alarm and low false alarm rate. Yuan (2008) designs a smoke detection system utilizing an accumulative motion model based on integral images by fast estimation of the smoke motion orientation to reduce the rate of false alarms, as the estimation accuracy can affect subsequent critical decisions. Hence, smoke orientation is accumulated over time to compensate for inaccuracy. A fuzzy logic method is employed to detect smoke in a real-time alarm system in Ho and Kuo (2009). Spectral, spatial, and temporal features are used for extracting smoke and for helping with the validation of smoke. Experimental results show that smoke can be successfully detected in different circumstances (indoor, outdoor, simple or complex background image, etc.). Although the results are promising, further development is still needed to integrate such findings with existing surveillance systems and implement them in actual operations. An approach using static and dynamic characteristic analysis for forest fire smoke detection is proposed by Surit and Chatwiriya (2011). Zhang et al. (2007) present an Otsu-based method to detect fire and smoke while segmenting fire and smoke together from the background. In Yu et al. (2010), a color-based decision rule and an optical flow algorithm are adopted for extracting the color and motion features of smoke, with experimental results showing significantly improved accuracy of video smoke detection[3].

1.3.1.2 Fire detection with infrared images

Since infrared (IR) images can be captured in either weak or no light situations while smoke is transparent in IR images, it is therefore applicative and practical to detect fires in both daytime and nighttime. Tables 1.4 and 1.5 list fire detection studies done by IR cameras.take advantage of a training-based threshold selection method to obtain binary images containing fire pixels from IR images. The false alarm rates are significantly reduced, since the appearance of fire is a high intensity region in IR images. Bosch et al. detect the occurrence of forest fires in IR images by using decision fusion. Various useful information for the fire detection can be acquired by this method. Pastor et al.Use linear transformations to precisely calculate the rate of spread (ROS) of forest fires in IR images, while a threshold-value-searching criterion is applied to locate the flame front position. Ononye et al.illustrate a multi-spectral IR image processing method which is capable of automatically obtaining the forest fire perimeter, active fire line, and fire propagation tendency. The proposed method is developed based on a sequence of image processing tools and a dynamic data-driven application system (DDDAS) concept. Huseynov et al. devise a multiple ANNs model for distinguishing flame in IR images. The experimental results show that the proposed approach can reduce training time and improve the success rate of classification.

One issue related to processing images collected by IR cameras is that miniaturized cameras still have low sensitivity. This phenomenon demands an augment in detector exposure periods to produce higher-quality images. In addition, the high frequency of vibration of UAV can lead to blurring images, which remains a major difficulty in their development.[15]



				Detection objects					
Detection method	Spectral bands	Resolution	Used features	Flame	Smoke	Geolocation	Propagation prediction	Image stabilization	References
Georeferenced uncertainty mosaic	IR	320×240	Color	√	×	√	√	√	Bradley and Taylor 2011
Statistical data fusion	Visual Mid-IR	752×582 256×256	Color	√	×	√	√	√	Martínez-de Dios et al. 2011
Training method	IR	160×120	Color	√	×	×	×	√	Martínez-de Dios and Ollero 2004
Training method	Visual Far-IR	320×240 —	Color	√	×	√	×	√	Merino et al. 2005, 2006
Training method	Visual Far-IR	320×240 —	Color	√	×	√	√	√	Merino et al. 2010, 2012
—	Visual IR	720×640 —	Color	√	×	√	×	—	Ambrosia 2002; Ambrosia et al. 2003
Genetic algorithm	IR	320×240	Color	√	×	×	×	×	Kontitsis et al. 2004
Training method	Visual IR	752×582 160×120	Color	√	×	√	×	—	Martínez-de Dios et al. 2005; Martínez-de Dios et al. 2006
—	Visual IR	— —	Color	√	×	√	×	—	Ollero et al. 2005; Ollero and Merino 2006; Maza et al. 2010, 2011

Note: IR, infrared; √, considered; ×, not considered; —, not mentioned.

Figure 1.5: Vision-based techniques for forest fire monitoring, detection, and fighting using UAVs in near-operational field[3].

1.3.1.3 Fusion of visual and IR images

One way of improving fire detection algorithms in terms of accuracy, robustness and reliability, is by fusing visual and IR images together. These improvements can be done by the use of fuzzy logic, probabilistic, statistical and intelligent methods (As illustrated in Table 1.4).Arrue et al.develop an algorithm in which IR image processing, fuzzy logic and artificial neural networks are used to improve fire detection operations. They use matching the information of visual and infrared images to confirm forest fires.authors integrate the information from both visual and infrared cameras for fire front parameter calculation by taking advantage of image processing techniques in both visual and IR images. They did not conduct any experiment out of laboratory. They continue in by introducing a forest fire perception system. By using computer vision techniques, visual and IR images are fused to extract a 3D fire perception. Then the fire propagation can be visualized by remote computer systems.[4]

1.4 Machine learning and computer vision

1.4.1 Introduction

AI is a subcategory of computer science that handles the simulation of intelligent activities in computers. AI is a computer system, which can accomplish responsibilities that usually need human acumen. Generally, a rule engine leads the AI system and a good AI system should have an intelligent rule engine, which is based on a series of meaningful IF–THEN statements. Since the 1950s, AI has been successfully used in visual perception, speech recognition, decision- making, and translation between languages. AI and ML are often used interchangeably, especially in the realm of big data. As shown in Figure 1.6, DL is considered as a subcategory of ML and again ML is a subcategory of AI. In other words, all DL is ML, but not all ML is DL, and so forth[5].

The machine learning and computer vision hopes to bring into the computers the human capabilities for data sensing, data understanding, and action taking based on the past and present outcomes. The machine learning solutions revolves around data gathering, training a model, and use the trained model to make predictions. There are models and services provided by private companies for speech recognition, text analysis, and image classification. One can use their models through application programming interfaces (API). For instance, Amazon Recognition, Polly, Lex, Mi-

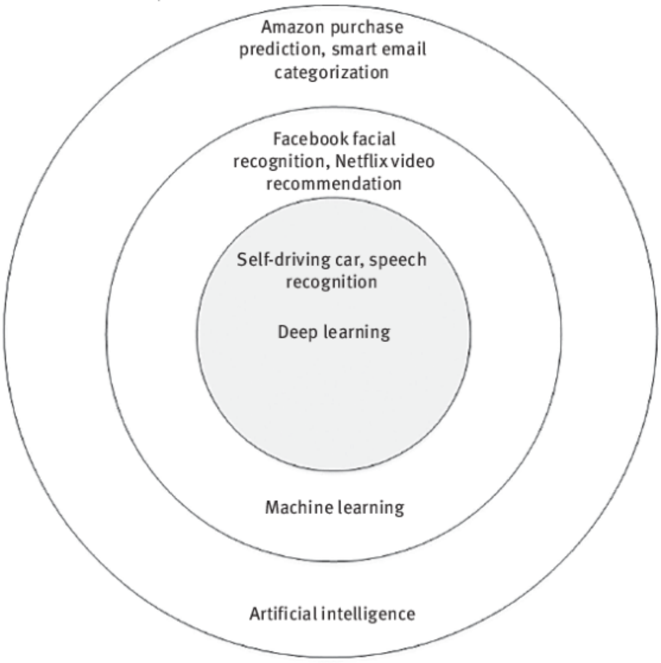

Amazon purchase prediction, smart email categorization
Facebook facial recognition, Netflix video recommendation
Self-driving car, speech recognition
Deep learning
Machine learning
Artificial intelligence

Figure 1.6: Relationship between AI, ML, and DL[5].

crosoft Azure Cognitive Services, IBM Watson The machine learning algorithms are now running on cloud as a “machine learning as a service”, “cloud machine learning” . Moreover, companies, such as Amazon, Microsoft, and Google, have machine learning as a cloud service.[16]

1.4.2 Computer vision

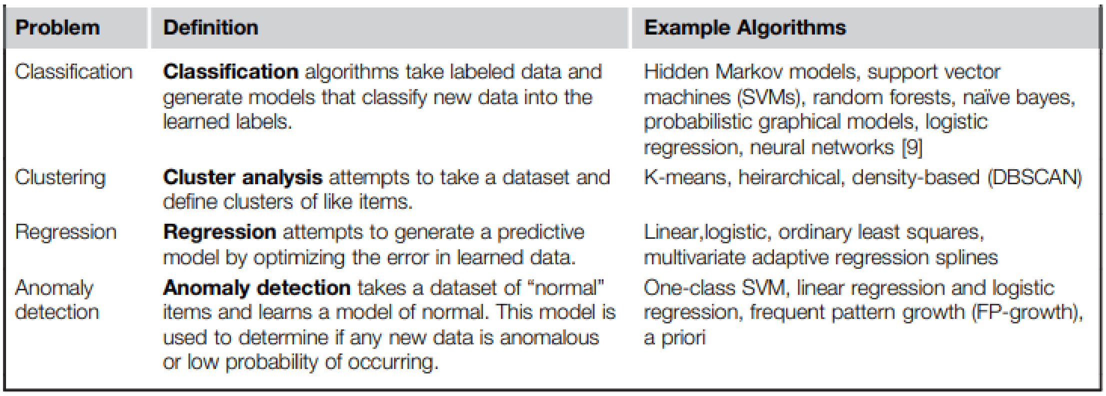
It is obvious to ask what computer vision is and why Machine Learning is a valuable tool for computer vision applications. Computer vision is a computer science field that works on digital images and videos to process and understand visual data. The focus of computer vision is to discover the pattern, objects, and their various properties. To discover these properties, researchers try to build a computer model similar to the human vision process. Computer vision’s core concept is to “teach” or train a computer system to understand the smallest part of an image or video and understand that. Based on that understanding, specific algorithms interpret the results. The finding often used for image and object recognition, segmentation, voice generation, self-driving cars, facial recognition, augmented and mixed reality, and medical image processing. Nevertheless, the task of computer vision is not an easy one. There are several challenges involved. Imitating a computer to human vision is the biggest challenge since we are not entirely sure how human vision processes things—helping computers to see turns out to be very hard. Inventing the machine that sees as we do is a deceptively tricky task, not just because it is hard to make computers do it, but because we are not entirely sure how human vision works in the first place. Studying biological vision requires an understanding of the perception organs like the eyes and the interpretation of the brain's perception. Much progress has already been made, both in charting the process and discovering the tricks and shortcuts used by the system, although like any study that involves the brain, there is a long way to go.

Image and videos are the main available data for computer vision. In current time, everybody has access to cameras, people takes a lot of pictures and capture videos. These huge amount of image and video data can be used for various purposes. For example, the law enforcement official may use those data to identify a person of their interest, a computer scientist may want to develop a gesture based computer access for physically challenged people, a farmer may want to identify a crop disease by taking a picture of a diseased crop. The implementation list can go long and new challenging problems are emerging daily basis. These requirements demand some efficient method that can solve the problem with precision. One of the endeavor to solve challenging computer vision problem is use of machine learning algorithm. [7]

1.4.3 Machine learning

Machine Learning is a process to train computer systems to solve real-world problems, focusing on image classification, object detection, segmentation, and so many. Machine Learning usually has two steps- Training and Testing. In the training stage, Machine Learning approaches "learn" some features from the input data and those learning knowledge used to do the task in the testing data. The training phase is a crucial one. The performance of a Machine Learning algorithm deeply depends on training. Each Machine Learning approach has its way of training. The training phase can be explained with a naïve example- let us assume we want to train a baby to recognize an apple. A baby does not have any prior knowledge of an apple. More specifically- how an apple looks like, the shape, and the color of an apple. To make a baby recognize an apple, what we can do is show a baby an apple. The baby will take a look at the apple and then try to learn some features like the shape, color, and texture of an apple. After that, if we put the apple into a fruit basket and ask the baby, which is an apple, the baby can recognize the apple. Based on the training process, all Machine Learning can be divided into - Supervised, Unsupervised, and semi-supervised. Each of these approaches has its pros and cons and use based on the problem domain. Supervised learning uses labeled data for training. Label data includes knowledge that is already known, and it is used to train a Machine Learning model. Supervised Machine Learning techniques are prevalent nowadays. Many models are developed and used for image classification, object detections. However, there is an adequate amount of available data- text, image, video time series now, but very few are labeled. Label data needs domain expert knowledge, and it is time-consuming and expensive. Semi-supervised Machine Learning takes advantage of both supervised and unsupervised Machine Learning. It takes a small amount of labeled data and a large amount of unlabeled data in semi-supervised learning. For that, it does not depend on the availability of labeled data. Unsupervised Machine Learning does not require any labeled data. So, this approach finds the underlying pattern of a given data. Clustering is one example of such a Machine Learning technique.[7]

There are two dimensions around which machine learning is generally categorized: the process by which it learns and the type of output or problem it attempts to solve. machine learning algorithms can be broken into four categories: classification, clustering, regression, and anomaly detection figure 1.7 [6]



Problem	Definition	Example Algorithms
Classification	**Classification** algorithms take labeled data and generate models that classify new data into the learned labels.	Hidden Markov models, support vector machines (SVMs), random forests, naïve bayes, probabilistic graphical models, logistic regression, neural networks [9]
Clustering	**Cluster analysis** attempts to take a dataset and define clusters of like items.	K-means, heirarchical, density-based (DBSCAN)
Regression	**Regression** attempts to generate a predictive model by optimizing the error in learned data.	Linear,logistic, ordinary least squares, multivariate adaptive regression splines
Anomaly detection	**Anomaly detection** takes a dataset of “normal” items and learns a model of normal. This model is used to determine if any new data is anomalous or low probability of occurring.	One-class SVM, linear regression and logistic regression, frequent pattern growth (FP-growth), a priori

Figure 1.7: Categories of Machine Learning Algorithms Separated by Problems they Address[6].

1.4.4 Supervised machine learning

Supervised learning uses known data set to train an algorithm with features to make the prediction. This known data set is also known as training data which is associated with the desired output. Supervised learning establishes a model by learning the relationship between features and the output that which used afterward to make a prediction of a new data-set (test data-set).
Given a training set of an instance in the form of (t1,f1)......... (tn,fn) where ti is the ith feature vector of training example, and fi is the label (known class). The supervised algorithm tries to map a function: T → F. this function is known as hypothesis space.A supervised algorithm strongly depends on training data. The performance is related to the choice of training data. It is crucial to choose the correct training data set.[7]

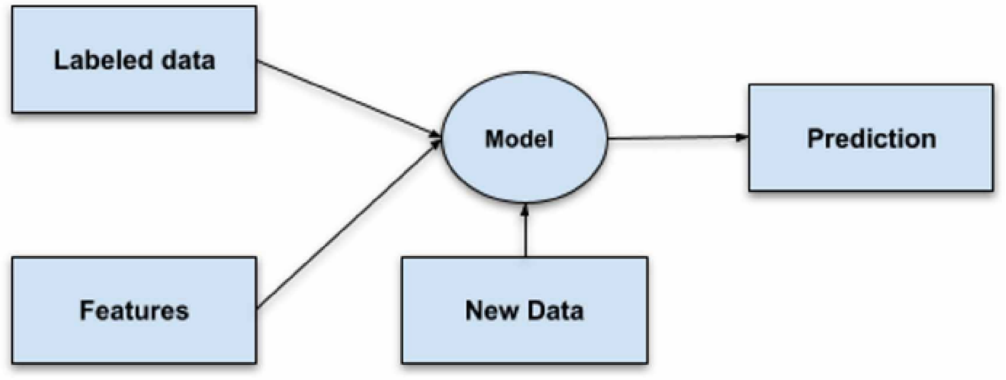


Figure 1.8: Supervised learning overview[7]

1.4.5 Convolutional neural network

Convolutional Neural Network (CNN) is a mostly used neural network for classification problems. It is very popular for groundbreaking accuracy. CNN is a deep learning algorithm that takes the input of an image and assigns weights and biases to various objects to the image. This capability makes different objects separable from each other. The architecture of CNN is related to the connectivity of neurons. It replicates the idea of the human brain to get prediction results. One important distinguishable characteristic of CNN is, it can capture the spatial and temporal dependencies in an image with filters. The basic CNN architecture is composed of convolution layer, pooling layer, and fully connected layer.

The convolution layer is the first part of the CNN. It consists of kernel/filters. The size of filters can be defined dynamically. These filters move from the beginning of an image to the end with a stride value. It traverses the whole image and performs a matrix multiplication, and puts all the results in a stack. All these results are combined with the bias, and that makes convoluted feature output or feature map. Each of the kernel creates one feature map. The size and depth of the filter depend on the input images. For example, an RGB image will have filters with a depth of three, and a binary image will have a depth of one. The main goal of the convolution layer is to extract features such as –edges, color, gradient orientation from the input image. The beginning convolutional layers extract basic features and as the network close to the end, the later convolution layers learn specific features.

The pooling layer reduces the size of the convoluted features and reduces the dimension. The pooling layer extracts the dominant features, which is rational to the training data. There are two types of pooling layers- max pooling and average pooling. The max-pooling returns the maximum value of the part of the image convoluted by the kernel, and average pooling returns the average value. Nonetheless, a convolution layer and a pooling layer together represent a “layer” in CNN. Based on the complexity of the requirement, the number of layers can increase and decrease. However, the more layer will not guarantee the best predictions.

Each combination of convolution and pooling layers, generally followed by an activation function layer. There are many activation layers available now but the most researchers use Rectified Linear Unit (ReLU). The ReLU layers change any negative values of features map to zero. For this reason, all values passes to pooling layers are non-negative.

A fully connected layer (FC-Layer) is a feedforward neural network. The input of this layer is the output of the last pooling layer of the network. The output of the pooling layer is flattened (unrolling a three-dimensional matrix to a vector). The flattened vector is connected to fully connected layers and performs the same calculations. The final layer used a Softmax activation function (mostly ReLU) that provides the probabilities of any input value associated with the desired class.Some prominent CNN architectures are known as- Alexnet, Googlenet, Resnet, VGG-16 and VGG 19. All these CNN is already trained with over a million dataset with a capability of classifying 1000 classes as mentioned in Imagenet. Those architectures

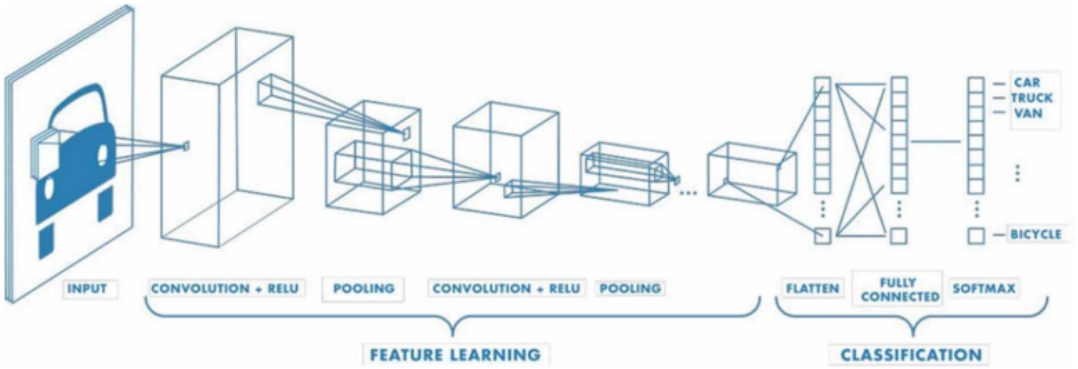


Figure 1.9: Structure of a Convolutional Neural Network (CNN)[7].

are carefully designed and tested for the optimal performances. Researchers can use them for their own dataset. This method is known as "Transfer learning". Transfer learning provides an easy way to classify images compared to build a CNN from scratch. Those prominent CNN models are open to modification for any specific classification task.[7]

1.4.6 Object detection

Object Detection is the task of classification and localization of objects in an image or video.Image classification labels the image as a whole. Finding the position of the object in addition to labeling the object is called object localization. Typically, the position of the object is defined by rectangular coordinates. Finding multiple objects in the image with rectangular coordinates is called detection[17].

Early object detection models were built as an ensemble of hand-crafted feature extractors such as Viola-Jones detector, Histogram of Oriented Gradients (HOG) etc. These models were slow, inaccurate and performed poorly on unfamiliar datasets. The re-introduction of convolutional neural network (CNNs) and deep learning for image classification changed the landscape of visual perception. Its use in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 challenge by AlexNet inspired further research of its application in computer vision. Today, object detection finds application from self-driving cars and identity detection to security and medical uses.[9]

1.4.6.1 Classification of object detection algorithms

In recent years, object detection algorithms have made great breakthroughs. The current popular object detection methods can be divided into two categories.
One is the R-CNN algorithm based on Region Proposal, such as R-CNN, Fast R-CNN, Faster R-CNN, etc. They are two-stage and require the First use of heuristic methods for example Selective search, or CNN network to generate Region Proposal and then perform classification and regression on Region Proposal. The other is one-stage algorithms such as Yolo and SSD, which only use a CNN network to directly predict the categories and positions of different targets.
The two-stage object detection algorithm needs to perform region extraction operations, first use the CNN backbone network to extract image features, then find possible candidate regions from the feature map, and finally perform sliding window operations on the candidate regions to further determine the target category and location information.

The one-stage object detection algorithm does not extract candidate regions through the intermediate layer, but performs feature extraction, target classification, and position regression in the entire convolutional network, and then obtains the target position and category. The recognition accuracy is slightly weaker than that of the two-stage object detection algorithm. Under the premise, the speed has been greatly improved. The development process of one-stage and two-stage algorithms is shown in figure 1.11 and figure 1.10, respectively.

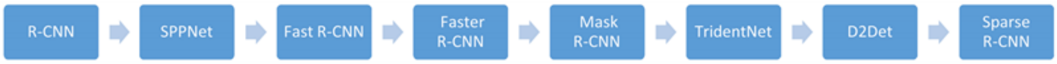


Figure 1.10: The development history of the two-stage object detection network framework[8].

1.4.6.2 Common object detection network model

R-CNN

In 2014, Ross Girshick proposed R-CNN , which uses a selective search algorithm

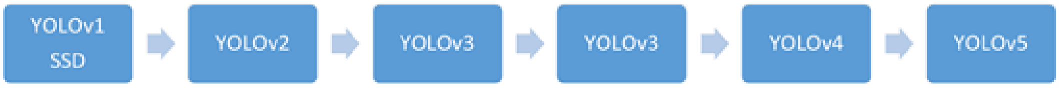


Figure 1.11: The development history of the one-stage object detection network framework[8].

to replace the sliding window, which solves the problem of window redundancy and reduces the time complexity of the algorithm. At the same time, the traditional hand-made feature extraction part is replaced with a convolutional neural network, which can more effectively extract the features of the image and improve the network's anti-interference ability.

RCNN first selects possible object frames from a set of object candidate frames through Selective Search algorithm, and then resizes the images in these selected object frames to a fixed size image, and feeds them to CNN The model (a CNN model trained on the ImageNet data set, such as AlexNet) extracts features, and finally sends the extracted features to the classifier to predict whether the image in the object frame has the target to be detected, and uses the regression to further predict Which category does the detection target belong to.

The performance of the R-CNN model has been greatly improved compared to traditional object detection algorithms, but there are also many problems. For example, R-CNN generates about 2000 candidate regions, and feature extraction takes too much time; convolutional neural networks require fixed-size input, and image cropping or stretching will cause loss of image information; training speed is slow, not only training image classification The network also needs to train the SVM classifier and regressor. The structure of R-CNN network is shown in Figure 1.12.

YOLO

YOLO was proposed in 2016 and published in CVPR, the computer vision conference.Unlike the R-CNN series that needs to find the candidate area first, and then identify the objects in the candidate area, YOLO's prediction is based on the entire picture, and it will output all detected target information at one time, including

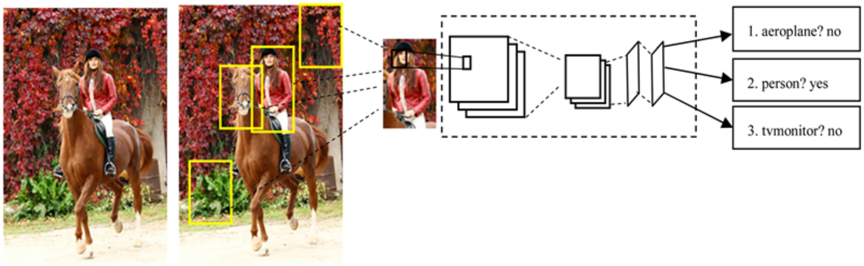


Figure 1.12: The architecture of the R-CNN framework[8].

category and location.The first step of YOLO is to divide the picture. It divides the picture into grids, and the size of each grid is equal. The core idea of YOLO is to turn object detection into a regression problem, using the entire image as the input of the network, and only going through a neural network to get the location of the bounding box and its category. Its detection speed is extremely fast, the generalization ability is strong, the speed is provided, and the accuracy is reduced. The disadvantage is that for small objects, overlapping objects cannot be detected. [8]

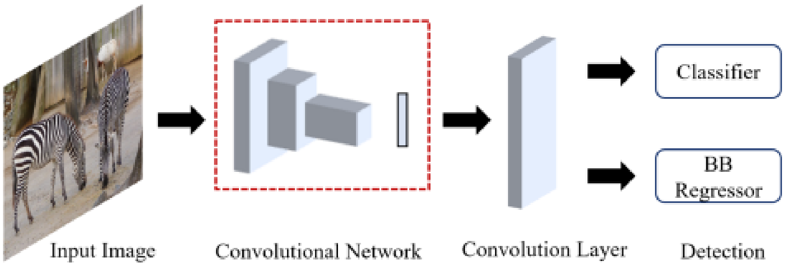


Figure 1.13: internal architecture of the yolo object detector[9]

1.4.6.3 Evaluation metrics for object detectors

Object detectors use multiple criteria to measure the performance of the detectors viz., frames per second (FPS), precision and recall. However, mean Average Precision (mAP) is the most common evaluation metric. Precision is derived from

Intersection over Union (IoU), which is the ratio of the area of overlap and the area of union between the ground truth and the predicted bounding box. A threshold is set to determine if the detection is correct. If the IoU is more than the threshold, it is classified as True Positive while an IoU below it is classified as False Positive. If the model fails to detect an object present in the ground truth, it is termed as False Negative. Precision measures the percentage of correct predictions while the recall measure the correct predictions with respect to the ground truth.

$$\begin{gathered} Precision = \frac{True\ \ Positive}{True\ \ Positive + False\ \ Positive}[9] \\ Precision = \frac{True\ \ Positive}{All\ \ Observations} \end{gathered} \tag{1.1}$$

$$\begin{gathered} Recall = \frac{True\ \ Positive}{True\ \ Positive + False\ \ Negative}[9] \\ Recall = \frac{True\ \ Positive}{Ground\ \ Truth} \end{gathered} \tag{1.2}$$

Based on the above equation, average precision is computed separately for each class. To compare performance between the detectors, the mean of average precision of all classes, called mean average precision (mAP) is used, which acts as a single metric for final evaluation.[9]

Chapter 2

Indoor positioning system

2.1 Introduction

In recent years, the landscape of technology and innovation has experienced a remarkable shift with the rapid increase in the usage of drones. Once largely limited to military applications, these unmanned aerial vehicles have now found diverse and rapidly growing commercial and domestic roles. Examples include Precision Agriculture, image collection, Logistics, monitoring, inspection, data collection, and even entertainment purposes. This mounting interest and usage surge have significantly boosted the demand for drone manufacturing, aiming to tailor these devices to the specific requirements of each application. An essential aspect of drone manufacturing lies in comprehensive testing, a process conducted primarily in controlled laboratory settings due to safety, security, and privacy considerations. However, a significant challenge arises during this phase: the absence or weak presence of GPS signals often hinders the accurate positioning of drones within the lab environment. This issue compromises testing validity, potentially leading to instability and inaccuracies in drone performance. Given that the fidelity of drone autopilots relies heavily on precise position corrections derived from GPS data – as Imus tend to exhibit divergence over the long term – resolving the lab-based GPS limitation becomes crucial for ensuring reliable and robust drone functionality

2.2 System architecture

To replace the reliance on GPS for drone position input, we developed an indoor positioning system utilizing a pair of cameras and computer vision techniques as object detection and depth estimation. This innovative system enables us to calculate the drone's position within the lab environment in real-time, referencing a customized frame of reference.

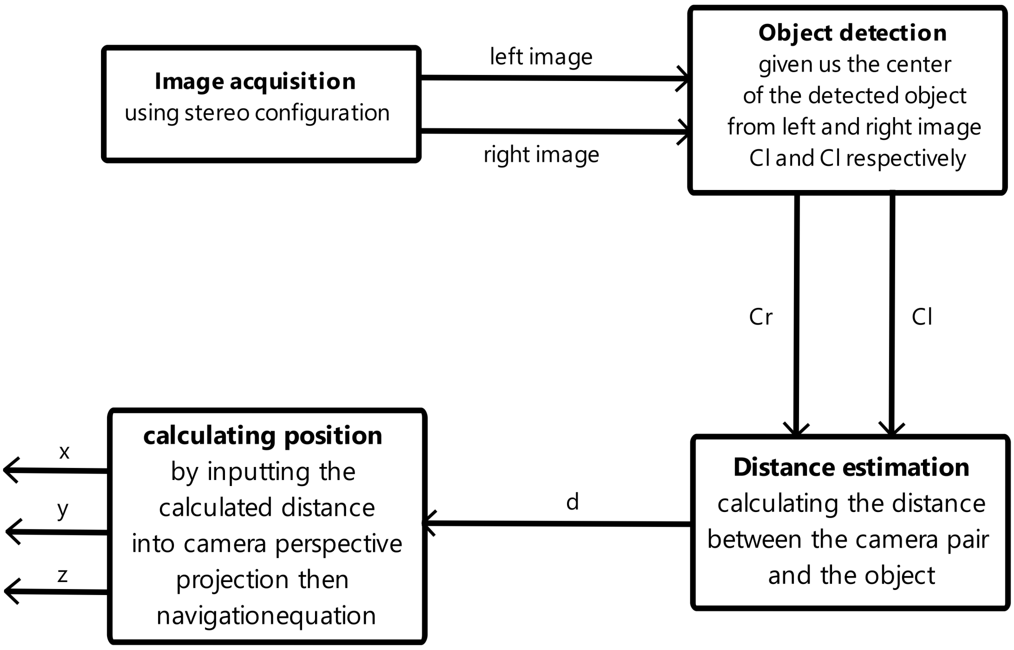


Figure 2.1: System operating diagram

The figure2.2 presents a comprehensive diagram illustrating the algorithm powering the IPS, highlighting various blocks and showcasing how they interact with one another.

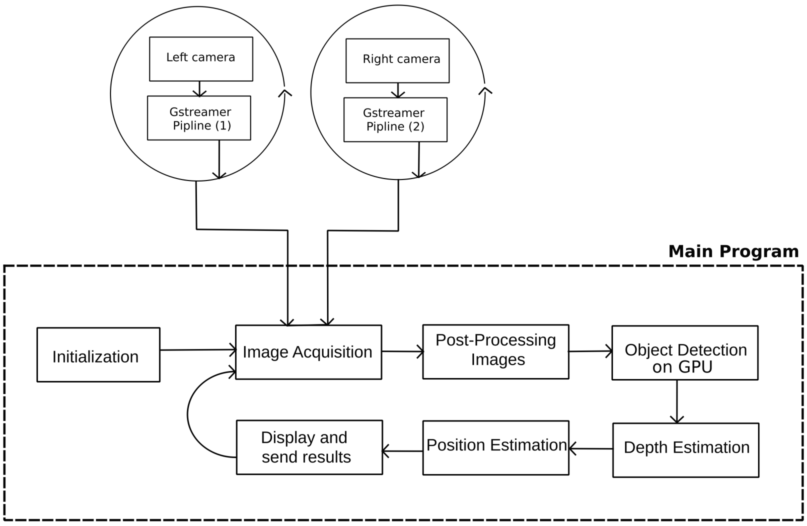


Figure 2.2: indoor positioning system algorithm diagram

2.3 Methodology

2.3.1 Configuration

The proposed system calculates the drone's position by utilizing visual inputs from both cameras simultaneously. This is accomplished by first defining the operational workspace and then aligning it with the overlapping field of view captured by both cameras. For an object to be properly detected, it must reside within this defined area, ensuring visibility to both the left and right cameras. The stereo setup involves positioning the cameras at a specific distance from each other, known as the baseline. This separation causes the views from the two cameras to be shifted either to the right or left, as illustrated in the accompanying figure 2.3.

Once the camera vision has been configured, it's essential to align the optical axes of the cameras to be both parallel and perpendicular to the baseline with utmost precision figure 2.4. This step holds significant importance as even the slightest deviation can greatly impact the overall precision. This meticulous alignment is necessary because the system relies on receiving two images with identical orientations.Any in-

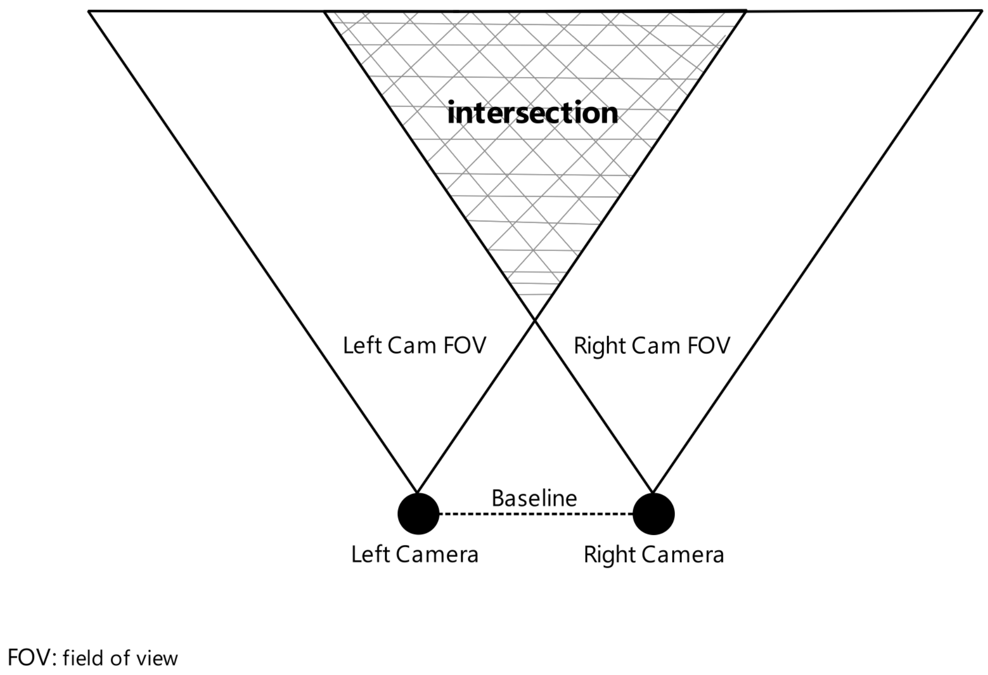


Figure 2.3: camera's vision

accuracies in the cameras' orientation would consequently lead to unmatched images, resulting in errors.

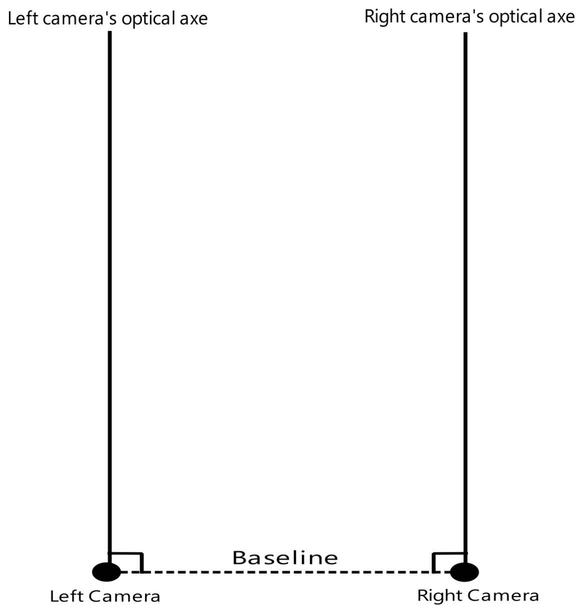


Figure 2.4: cameras alignment

Once the configuration is complete, we establish a connection between the cameras and a computer to acquire input images, enabling us to receive all the data in synchronized timing. On the software front, we ensure the utilization of multiprocessing by assigning each camera stream to a dedicated thread. This approach enables both data acquisition processes to occur concurrently Following this, we proceed to input each captured image into an object detection model. Specifically, we have chosen to utilize YOLOv5 due to its exceptional equilibrium between detection precision and speed. Through the detection generated by both the left and right images, we extract the pixel coordinates representing the center of each detected object. These pixel coordinates assume a crucial role in our forthcoming explanation of the depth estimation technique within this chapter, which serves as the fundamental pillar of our indoor positioning system.Once we have the distance between the detected object and the camera pair.

2.3.2 Position estimation

we proceed to calculate the object's coordinates by preforming backward camera projection as explained in [18],accomplished through the translation of object coordinates from the image frame to the world frame.

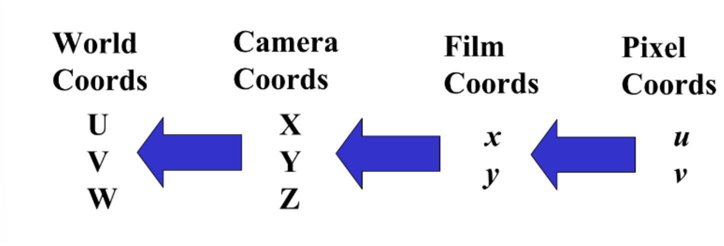


Figure 2.5: backward camera projection

2.3.2.1 From pixel to film reference

The pixel coordinates denote the position from the top-left corner of the camera's film to the desired pixel within it, depicted by the vectors (u, v). Conversely, the coordinates in film reference indicate the position of the desired pixel on the camera's film plane, originating from the intersection point of the optical axis, denoted by (x, u), as explained in the following figure. We use a resolution of 1920 x 1080 for this purpose.

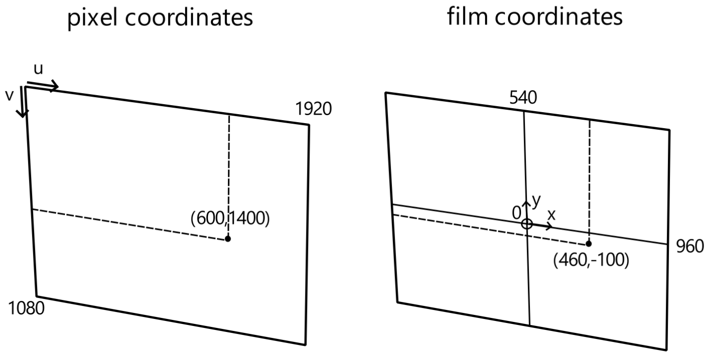


Figure 2.6: difference between film and pixel coordinates

We can calculate film coordinates using the flowing equations (2.1) (2.2).

$$x = u - (W/2) \tag{2.1}$$

$$y = (H/2) - v \tag{2.2}$$

2.3.2.2 From film to camera reference

we pass from film reference to camera reference which is the position of a given point in space taking the camera optical axe as a reference by applying basic perspective projection as illustrated in figure (2.7) using the following equations (2.3) (2.4) :

$$X = xZ/f \tag{2.3}$$

$$Y = yZ/f \tag{2.4}$$

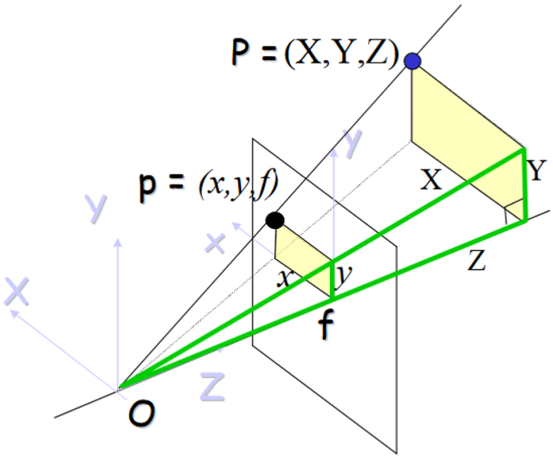


Figure 2.7: perspective projection

With Z is the calculated distance between the cameras pair and the detected object ,while O is the optical center , f is the focal length distance between the camera's film and the sensor , (x,y) film coordinates and X,Y,Z are camera coordinates.

2.3.2.3 From camera to world reference

In our situation, the room serves as our main reference point. To convert camera coordinates to real-world coordinates, we begin by measuring the exact world position of the chosen right camera Pwc(Uc,Vc,Wc), which acts as our reference for the conversion process. To simplify this, we align the right camera with the room's horizontal axis, setting Uc to zero. By measuring its height from the floor, we determine Vc, and by measuring its distance from the origin O, we determine Wc. Additionally, we calculate the angle between the camera pair and the floor θ, which we'll use later for rotations.

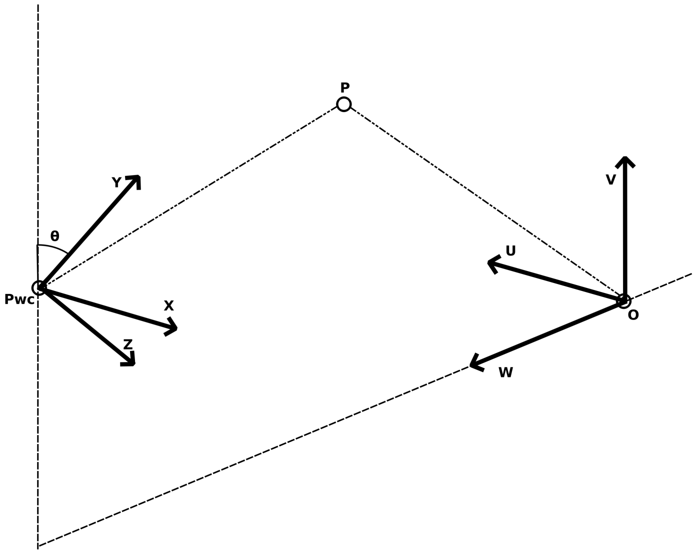


Figure 2.8: representation of both camera and world references

We then advance by aligning the camera frame axes with the world frame axes. This is achieved through the sequential application of two rotation matrices: first along the X-axis by an angle of θ (2.5), followed by a rotation along the Y-axis by an angle of 180° (2.6),Resulting in the final combined rotation matrix R (2.7).

$$R_X = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & \text{-}\sin(\theta)/ & \cos(\theta) \end{pmatrix} \tag{2.5}$$

$$R_Y = \begin{pmatrix} \cos(180^\circ\ ) & \sin(180^\circ\ ) & 0 \\ -\sin(180^\circ\ ) & \cos(180^\circ\ ) & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{2.6}$$

$$R = R_Y R_X \tag{2.7}$$

In the [18], it is explained that transitioning from the world coordinate system to the camera coordinate system is achieved using the given equation (2.8). Conversely, by inverting the equation (2.9), we can establish the transformation from the camera coordinate system back to the world reference, which is detailed in (2.10).

$$P_{world} = R(P_{camera} - Pwc) \tag{2.8}$$

$$P_{camera} = R^{-1} P_{camera} + Pwc \tag{2.9}$$

$$\begin{pmatrix} U \\ V \\ W \\ 1 \end{pmatrix} = \begin{pmatrix} R_{1,1} & R_{2,1} & R_{3,1} & 0 \\ R_{1,2} & R_{2,2} & R_{3,2} & 0 \\ R_{1,3} & R_{2,3} & R_{3,3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} + \begin{pmatrix} Uc \\ Vc \\ Wc \\ 0 \end{pmatrix} \tag{2.10}$$

2.4 Depth estimation

This computer vision technique forms the core of our indoor positioning system. Estimating the distance between the pair of cameras and the detected object marks the starting point for position estimation. Without this step, achieving accurate positioning is not possible. In our approach, we utilize a pair of cameras set up in a stereo configuration, which gives us a binocular vision. This binocular vision allows us to employ two different methods: disparity and triangulation. These methods will be explained further in this section. However, before applying any techniques, it's crucial to grasp our camera models and their functioning. In our case, we are using the pinhole camera model. Subsequently, we need to identify imperfections and errors in the input images that shape our perception of the real world. Correcting these issues becomes essential for achieving optimal results.

2.4.1 Generalities

2.4.1.1 The pinhole camera

The pinhole camera Or projective camera,This is a purely geometric model that describes the process whereby points in the world are projected into the image. Clearly, the position in the image depends on the position in the world, and the pinhole camera model captures this relationship.

In real life, a pinhole camera consists of a chamber1 with a small hole (the pinhole) in the front 2.9. Rays from an object in the world pass through this hole to form an inverted image on the back face of the box, or image plane. Our goal is to build a mathematical model of this process.

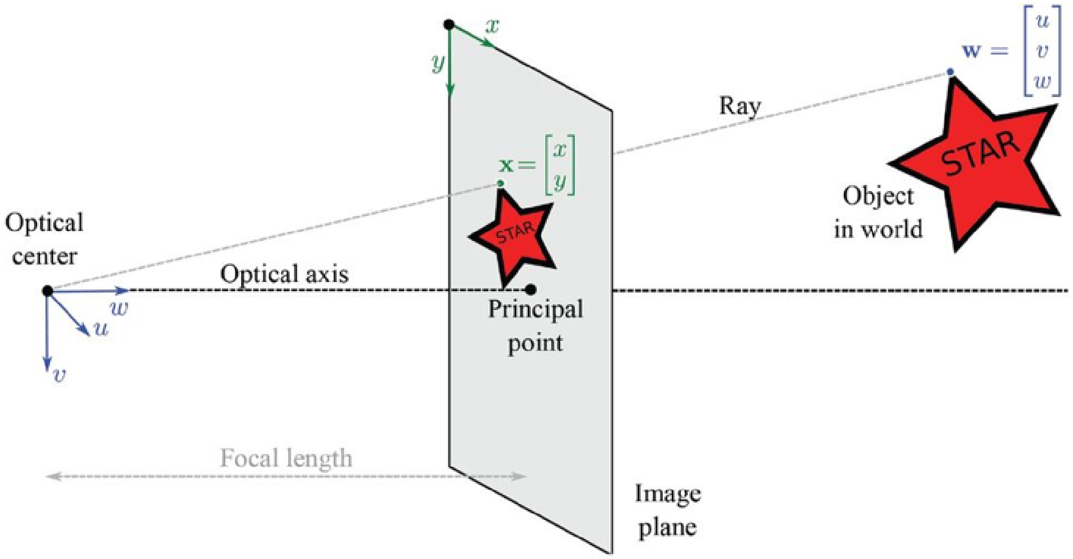

x
y
$\mathbf{w} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$
Ray
$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$
STAR
STAR
Object in world
Optical center
Optical axis
w
u
v
Principal point
Focal length
Image plane

Figure 2.9: The pinhole camera model[10]

It is slightly inconvenient that the image from the pinhole camera is upside-down. Hence, we instead consider the virtual image that would result from placing the image plane in front of the pinhole. Of course, it is not physically possible to build a camera this way, but it is mathematically equivalent to the true pinhole model (except that the image is the right way up) and it is easier to think about. From now on, we will always draw the image plane in front of the pinhole.Figure2.10 illustrates the pinhole camera model and defines some terminology. The pinhole itself (the point at which the rays converge) is called the optical center. We will assume for now that the optical center is at the origin of the 3D world coordinate system, in which points are represented as $w = [u, v, w]^T$. The virtual image is created on the image plane, which is displaced from the optical center along the w-axis or optical axis. The point where the optical axis strikes the image plane is known as the principal point. The distance between the principal point and the optical center (i.e., the distance between the image plane and the pinhole) is known as the focal length.

The pinhole camera model is a generative model that describes the likelihood Pr(x—w) of observing a feature at position $x = [x, y]^T$ in the image given that it is the projection of a 3D point $w = [u, v, w]^T$ in the world. Although light transport is essentially deterministic, we will nonetheless build a probability model; there is noise in the sensor, and unmodeled factors in the feature detection process can also affect the measured image position. However, for pedagogical reasons we will defer a discussion of this uncertainty until later, and temporarily treat the imaging process as if it were deterministic.

Our task then is to establish the position $x = [x, y]^T$ where the 3D point $w = [u, v, w]^T$ is imaged. Considering Figure 2.10 it is clear how to do this. We connect a ray between w and the optical center. The image position x can be found by observing where this ray strikes the image plane. This process is called perspective projection. In the next few sections, we will build a more precise mathematical model of this process. We will start with a very simple camera model (the normalized camera) and build up to a full camera parameterization.

2.4.1.2 The normalized camera

In the normalized camera, the focal length is one, and it is assumed that the origin of the 2D coordinate system (x,y) on the image plane is centered at the principal point. 2.11 shows a 2D slice of the geometry of this system (the u- and x-axes now point upward out of the page and cannot be seen). By similar triangles, it can easily be seen that the y-position in the image of the world point at $w = [u, v, w]^T$ is given by v/w. More generally, in the normalized camera, a 3D point $w = [u, v, w]^T$ is projected into the image at $x = [x, y]^T$ using the relations

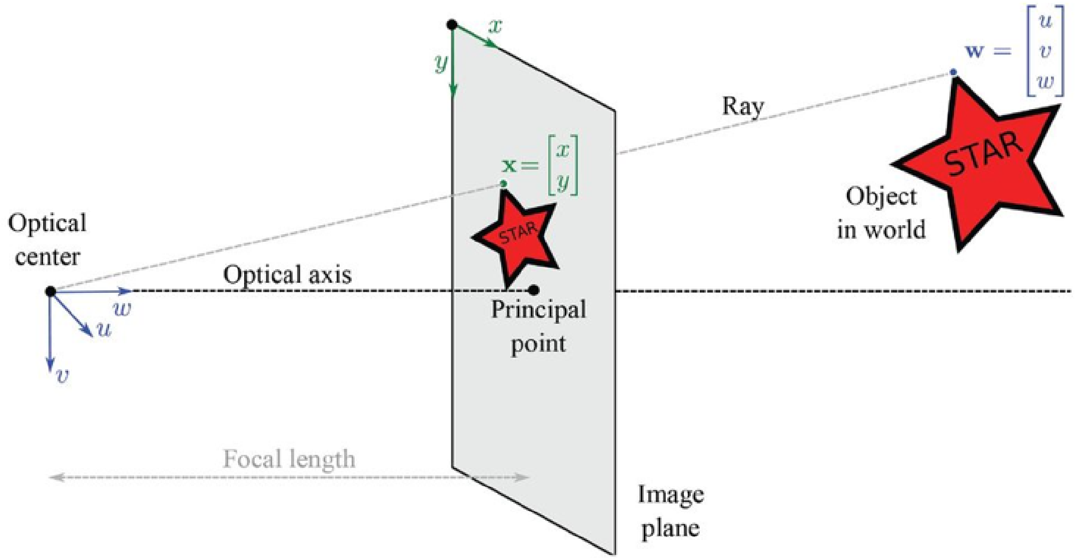

x
y
$\mathbf{w} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$
Ray
$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$
STAR
Object
in world
STAR
Optical
center
Optical axis
w
u
v
Principal
point
Focal length
Image
plane

Figure 2.10: Pinhole camera model terminology[10]

$$\begin{aligned} x &= \frac{u}{w} \\ y &= \frac{u}{w} \end{aligned} [10] \tag{2.11}$$

where x,y,u,v, and w are measured in the same real-world units (e.g., mm).

2.4.1.3 Focal length parameters

The normalized camera is unrealistic; for one thing, in a real camera, there is no particular reason why the focal length should be one. Moreover, the final position in the image is measured in pixels, not physical distance, and so the model must take into account the photoreceptor spacing. Both of these factors have the effect of changing the mapping between points $w = [u, v, w]^T$ in the 3D world and their 2D positions $x = [x, y]^T$ in the image plane by a constant scaling factorϕ 2.12so that

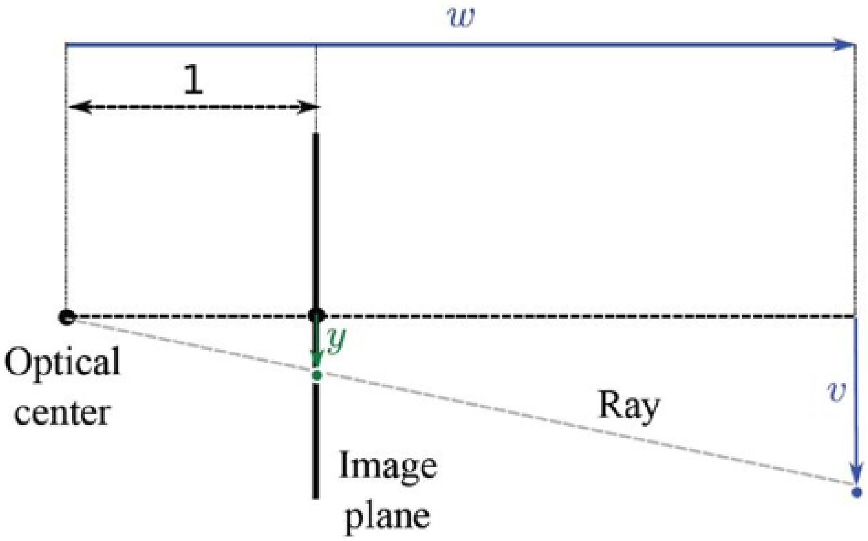


Figure 2.11: Normalized camera[10]

$$\begin{aligned} x &= \frac{\phi u}{w} \\ y &= \frac{\phi u}{w} \end{aligned} [10] \tag{2.12}$$

To add a further complication, the spacing of the photoreceptors may differ in the x- and y directions, so the scaling may be different in each direction, giving the relations :

$$\begin{aligned} x &= \frac{\phi_x u}{w} \\ y &= \frac{\phi_y u}{w} \end{aligned} [10] \tag{2.13}$$

where ϕ_x and ϕ_y are separate scaling factors for the x- and y- directions. These parameters are known as the focal length parameters in the x- and y-directions, but this name is somewhat misleading – they account for not just the distance between the optical center and the principal point (the true focal length) but also the photoreceptor spacing.

2.4.1.4 Offset and skew parameters

The model so far is still incomplete in that pixel position $x = [0, 0]^T$ is at the principal point (where the w-axis intersects the image plane). In most imaging systems, the pixel position $x = [0, 0]^T$ is at the top-left of the image rather than the center. To cope with this, we add offset parameters δ_x and δ_y so that

$$\begin{aligned} x &= \frac{\phi_x u}{w} + \delta_x \\ y &= \frac{\phi_y u}{w} + \delta_y \end{aligned} \quad [10] \tag{2.14}$$

where δ_x and δ_y are the offsets in pixels from the top-left corner of the image to the position where the w-axis strikes the image plane. Another way to think about this is that the vector $[\delta_x$,$\delta_y]$T is the position of the principal point in pixels. If the image plane is exactly centered on the w-axis, these offset parameters should be half the image size: for a 640 × 480 VGA image δ_x and δ_y would be 320 and 240, respectively. However, in practice it is difficult and superfluous to manufacture cameras with the imaging sensor perfectly centered, and so we treat the offset parameters as unknown quantities.

We also introduce a skew term γ that moderates the projected position x as a function of the height v in the world. This parameter has no clear physical interpretation but can help explain the projection of points into the image in practice. The resulting camera model is

$$\begin{aligned} x &= \frac{\phi_x u + \gamma u}{w} + \delta_x \\ y &= \frac{\phi_y u}{w} + \delta_y \end{aligned} \quad [10] \tag{2.15}$$

2.4.1.5 Position and orientation of camera

Finally, we must account for the fact that the camera is not always conveniently centered at the origin of the world coordinate system with the optical axis exactly aligned with the w-axis. In general, we may want to define an arbitrary world coordinate system that may be common to more than one camera. To this end, we express the world points w in the coordinate system of the camera before they are passed through the projection model, using the coordinate transformation:

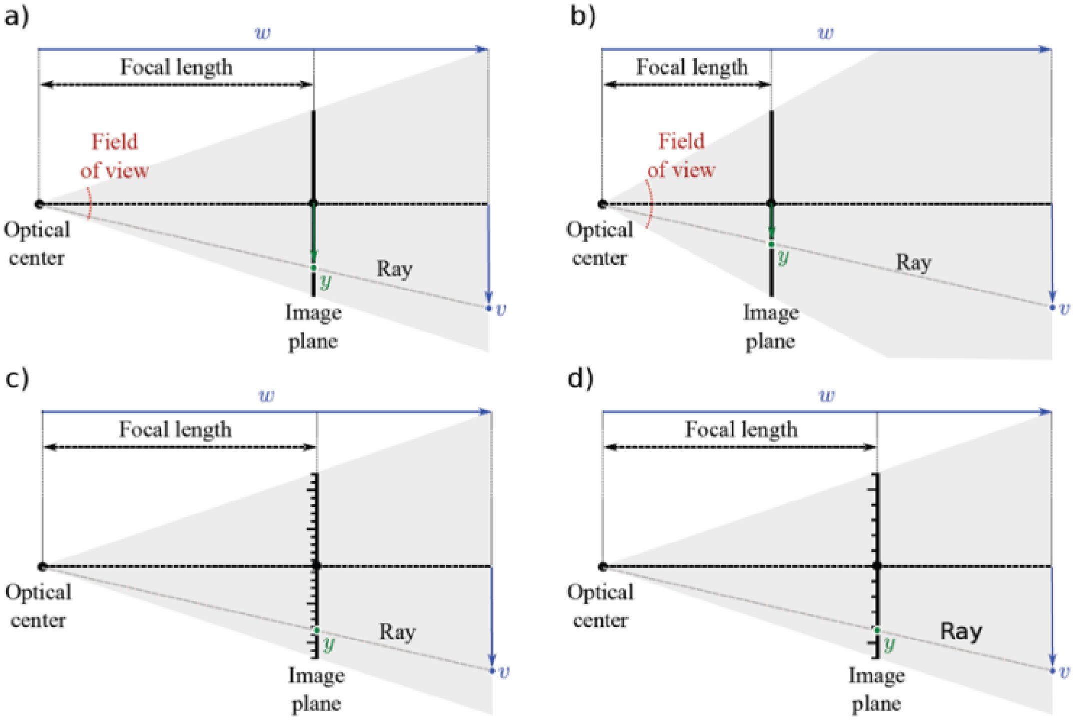


Figure 2.12: Focal length and photoreceptor spacing[10]

$$\begin{pmatrix} u' \\ v' \\ w' \end{pmatrix} = \begin{pmatrix} w_{1,1} & w_{2,1} & w_{3,1} \\ w_{1,2} & w_{2,2} & w_{3,2} \\ w_{1,3} & w_{2,3} & w_{3,3} \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \begin{pmatrix} \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} [10] \tag{2.16}$$

or

$$w' = \Omega w + \tau[10] \tag{2.17}$$

where w' is the transformed point, Ω is a 3 × 3 rotation matrix, and τ is a 3× 1 translation vector.

2.4.1.6 Full pinhole camera model

We are now in a position to describe the full camera model, by combining Equations 2.15 and 2.18.

A 3D point $w = [u, v, w]^T$ is projected to a 2D point $x = [x, y]^T$ by the relations :

$$\begin{aligned} x &= \frac{\phi_x(w_{11}u + w_{12}v + w_{13}w + \tau_y) + \gamma(w_{21}u + w_{22}v + w_{23}w + \tau_y)}{w_{31}u + w_{32}v + w_{33}w + \tau_s} + \delta_x \\ y &= \frac{\phi_y((w_{21}u + w_{22}v + w_{23}w + \tau_y))}{w_{31}u + w_{32}v + w_{33}w + \tau_s} + \delta_y \end{aligned} [10] \quad (2.18)$$

There are two sets of parameters in this model. The intrinsic or camera parameters ϕ_x and ϕ_y, γ, δ_x, δ_y describe the camera itself, and the extrinsic parameters Ω, τ describe the position and orientation of the camera in the world. we will store the intrinsic parameters in the intrinsic matrix Λ where

$$\Lambda = \begin{pmatrix} \phi_x & \gamma & \delta_x \\ 0 & \phi_y & \delta_y \\ 0 & 0 & 1 \end{pmatrix} [10] \quad (2.19)$$

We can now abbreviate the full projection model 2.18 by just writing

$$x = pinhole[w, \lambda, \Omega, \tau][10] \quad (2.20)$$

Finally, we must account for the fact that the estimated position of a feature in the image may differ from our predictions. There are a number of reasons for this, including noise in the sensor, sampling issues, and the fact that the detected position in the image may change at different viewpoints. We model these factors with additive noise that is normally distributed with a spherical covariance to give the final relation

$$Pr(x|w, \lambda, \Omega, \tau) = Norm_x[pinhole[w, \lambda, \Omega, \tau], \sigma^2 I][10] \quad (2.21)$$

where σ^2 is the variance of the noise.
Note that the pinhole camera is a generative model. We are describing the likelihood Pr(x—w,λ,Ω,τ) of observing a 2D image point x given the 3D world point w and the parameters λ,Ω,τ.

2.4.1.7 Radial distortion

In the previous section, we introduced the pinhole camera model. However, it has probably not escaped your attention that real-world cameras are rarely based on the pinhole: they have a lens (or possibly a system of several lenses) that collects light from a larger area and refocuses it on the image plane. In practice, this leads to a

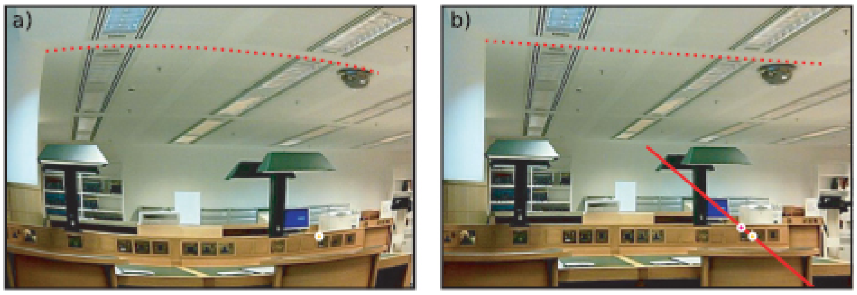


Figure 2.13: Radial distortion[10]

number of deviations from the pinhole model. For example, some parts of the image may be out of focus, which essentially means that the assumption that a point in the world w maps to a single point in the image x is no longer valid. There are more complex mathematical models for cameras that deal effectively with this situation, but they are not discussed here.

However, there is one deviation from the pinhole model that must be addressed. Radial distortion is a nonlinear warping of the image that depends on the distance from the center of the image. In practice, this occurs when the field of view of the lens system is large. It can easily be detected in an image because straight lines in the world no longer project to straight lines in the image 2.13 Radial distortion is commonly modeled as a polynomial function of the distance r from the center of the image. In the normalized camera, the final image positions (x',y') are expressed as functions of the original positions (x,y) by

$$\begin{matrix} x' = x(1+\beta_1 r^2 + \beta_2 r^4) \\ y' = y(1+\beta_1 r^2 + \beta_2 r^4), \end{matrix}[10] \tag{2.22}$$

where the parameters β_1 and β_2 control the degree of distortion. These relations describe a family of possible distortions that approximate the true distortion closely for most common lenses. This distortion is implemented after perspective projection (division by w) but before the effect of the intrinsic parameters (focal length, offset, etc.), so the warping is relative to the optical axis and not the origin of the pixel coordinate system. [10]

2.4.1.8 Camera calibration

Camera calibration is the estimation of the intrinsic and extrinsic parameters of a camera from the image positions of scene features such as points of lines, whose positions are known in some fixed world coordinate system 2.14. In this context, camera calibration can be modeled as an optimization process, where the discrepancy between the observed image features and their theoretical positions is minimized with respect to the camera's intrinsic and extrinsic parameters.[11]

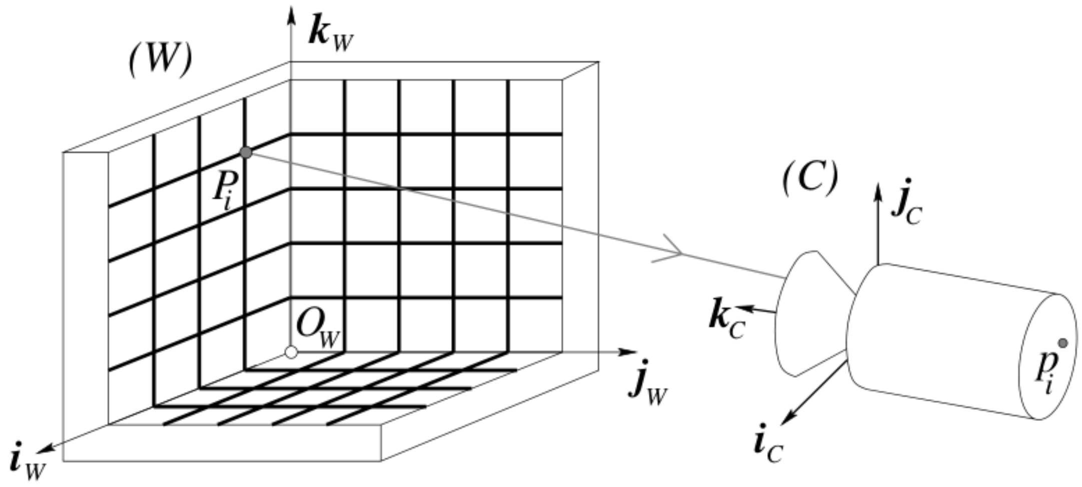


Figure 2.14: Camera calibration setup[11]

2.4.1.9 Calibration patterns

The use of a calibration pattern or set of markers is one of the more reliable ways to estimate a camera's intrinsic parameters. In photogrammetry, it is common to set up a camera in a large field looking at distant calibration targets whose exact location has been precomputed using surveying equipment. In this case, the translational component of the pose becomes irrelevant and only the camera rotation and intrinsic parameters need to be recovered.

2.4.1.10 Planar calibration patterns

When a finite workspace is being used and accurate machining and motion control platforms are available, a good way to perform calibration is to move a planar calibration target in a controlled fashion through the workspace volume. This approach is sometimes called the N- planes calibration approach and has the advantage that each camera pixel can be mapped to a unique 3D ray in space, which takes care of both linear effects modeled by the calibration matrix K and non-linear effects such as radial distortion . A less cumbersome but also less accurate calibration can be obtained by waving a planar calibration pattern in front of a camera. In this case, the pattern's pose has (in principle) to be recovered in conjunction with the intrinsic. In this technique, each input image is used to compute a separate homography H mapping the plane's calibration points (Xi, Yi, 0) into image coordinates (xi, yi)[19]

$$\begin{pmatrix} x_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} \simeq k \begin{pmatrix} r0 & r1 & t \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} \simeq Hp_i[19] \tag{2.23}$$

2.4.1.11 Stereo calibration

After the calibration of the monocular camera, the calibration result is calibrated for multi-eye stereoscopic calibration, that is, to solve the spatial position relationship of each camera in the camera system in the world coordinate system, that is, the rotation vector R and translation matrix T between multiple cameras need to be calculated[20].

2.4.1.12 Image rectification

The calculations associated with stereo algorithms are often considerably simplified when the images of interest have been rectified—that is, replaced by two equivalent pictures with a common image plane parallel to the baseline joining the two optical centers 2.15 . The rectification process can be implemented by projecting the original pictures onto the new image plane [11]

2.4.2 Method 1 with disparity

In an image, a detected 2D feature is the perspective projection of a 3D feature in the scene. A number of 3D points can project onto the same 2D point, which results in the loss of depth information. To recover this lost information, two images taken from different perspectives are required. The first step involved in recovering

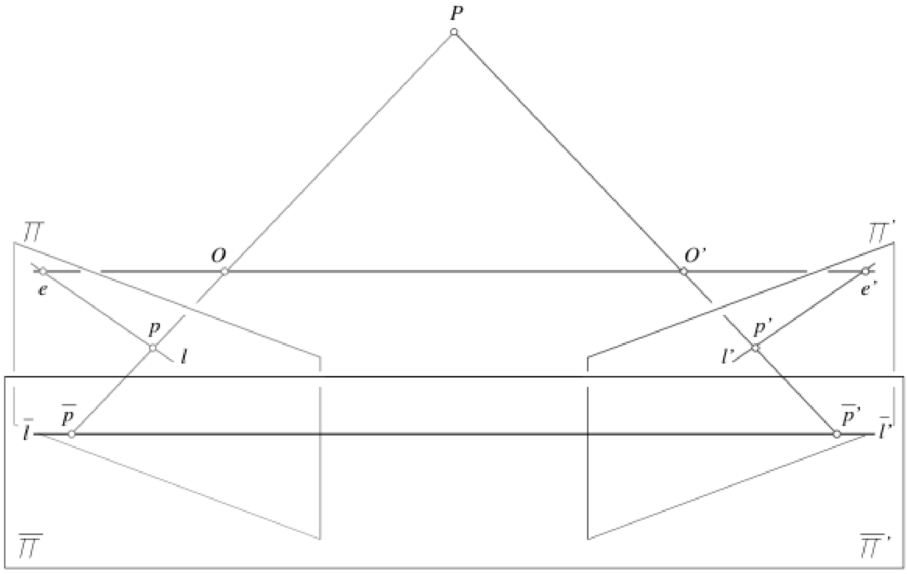


Figure 2.15: A rectified stereo pair[11]

3D spatial information is to establish the perspective transformation relationship between the scene and its projection onto the left and right images. As shown in 2.16, a point P defined by its coordinates (x, y, z) in the real world, will project as the corresponding 2D image coordinates (x_l, y_l) and (x_r, y_r) onto the left and right images respectively. The two cameras are separated by a fixed baseline distance D and have a known focal length f . Considering 0 to be the origin, which coincides with the image center in the left camera, then the perspective projections can be defined through simple algebra.

$$x = x_l.D/d[21] \tag{2.24}$$

$$y = y_l.D/d[21] \tag{2.25}$$

$$z = f.D/d[21] \tag{2.26}$$

where d is defined as the disparity between the two corresponding features in the left and right images:

$$d = |x_l - x_r|[21] \tag{2.27}$$

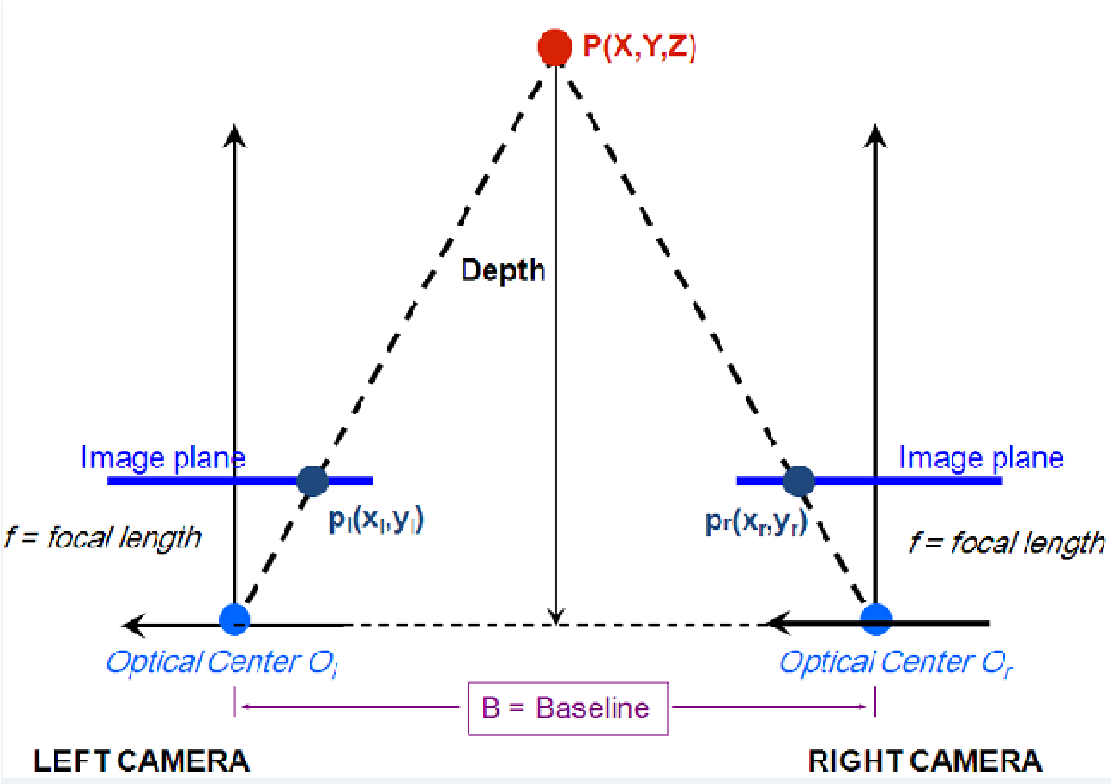


Figure 2.16: Stereo vision basics [12]

These equations provide the basis for deriving the 3D depth from stereo images [21].

2.4.3 Method 2 with triangulation

Triangulation stands as a proficient technique to determine the position of a point by gauging the angles to it from two established points situated at the extremities of a fixed baseline. This distinctive approach, distinct from the direct distance measurement in trilateration, leverages trigonometric principles to derive the point's coordinates. This point subsequently becomes the pivotal third vertex of a triangle, whereby one side and two angles are already ascertained. The inherent simplicity of this method has enabled its application, particularly in harnessing the potential of a stereo configuration to shape the requisite triangle. The angles necessary for computation are adeptly extracted through the utilization of computer vision, thereby underscoring the precision and efficacy of this methodology.

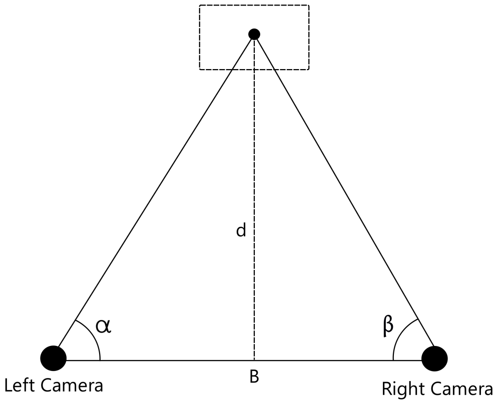


Figure 2.17: triangulation representation

this approach, known as triangulation, is widely employed by numerous [22] [23] due to its effective blend of affordability and dependability. However, its success depends significantly on the alignment and orientation of the cameras, as these factors directly influence angle estimation, subsequently impacting the entire calculation process. Hence, careful precision is vital during the setup of the configuration to guarantee precise results.

When it comes to determining the angle of the target object from each camera, we can employ the technique discussed in [24]. This method draws upon an understanding of the camera model, basic geometry, and the object's coordinates within the camera's film. By utilizing the triangle formed by the optical axis, the detected object, and the camera film as illustrated in figure 2.18 , we can express this relationship in the following equation (2.28),where f is the focal length and x is the object horizontal coordinate in film reference

$$\alpha = arctan(x/f) \tag{2.28}$$

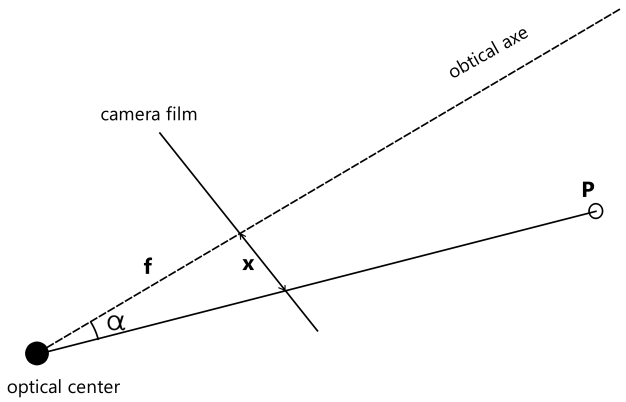


Figure 2.18: estimate angles with camera

Once we have acquired the angle of the object from each camera, the next step involves estimating the distance between the camera pair and the target object. This estimation can be achieved through the utilization of the triangulation equation(2.30)

$$l = \frac{d}{\tan(\alpha)} + \frac{d}{\tan(\beta)} = d(\frac{\cos(\alpha)}{\sin(\alpha)} + \frac{\cos(\beta)}{\sin(\beta)}) = d\frac{\sin(\alpha+\beta)}{\sin(\alpha)\sin(\beta)} \tag{2.29}$$

$$d = l\frac{\sin(\alpha)\sin(\beta)}{\sin(\alpha+\beta)} \tag{2.30}$$

2.5 Experimental setup

We initiated the construction of this system by procuring two identical WiFi IP cameras initially employed for surveillance purposes. Our selection was based on their favorable resolution (1920 x 1080) and FOV. Moreover, these cameras were chosen for their compatibility, given their utilization of the RTSP protocol for video output. Additionally, they exhibited an acceptable frame rate of 12 fps.

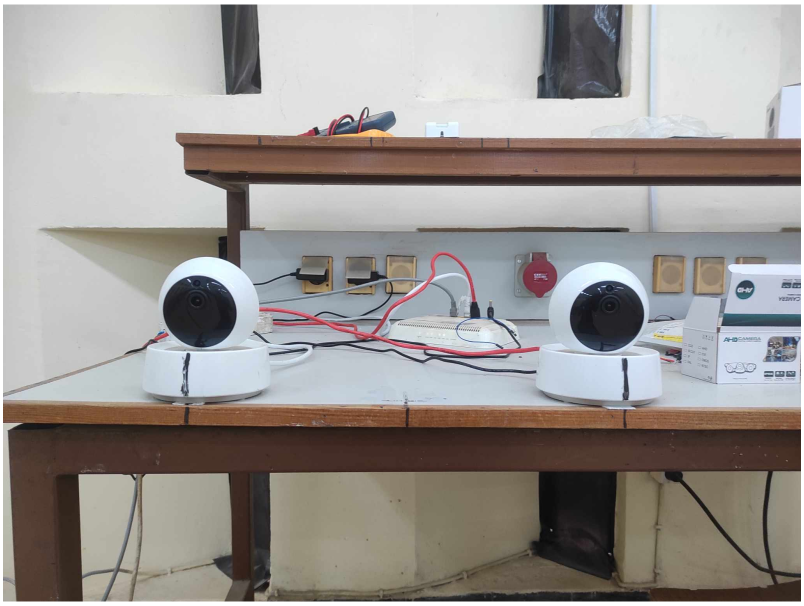


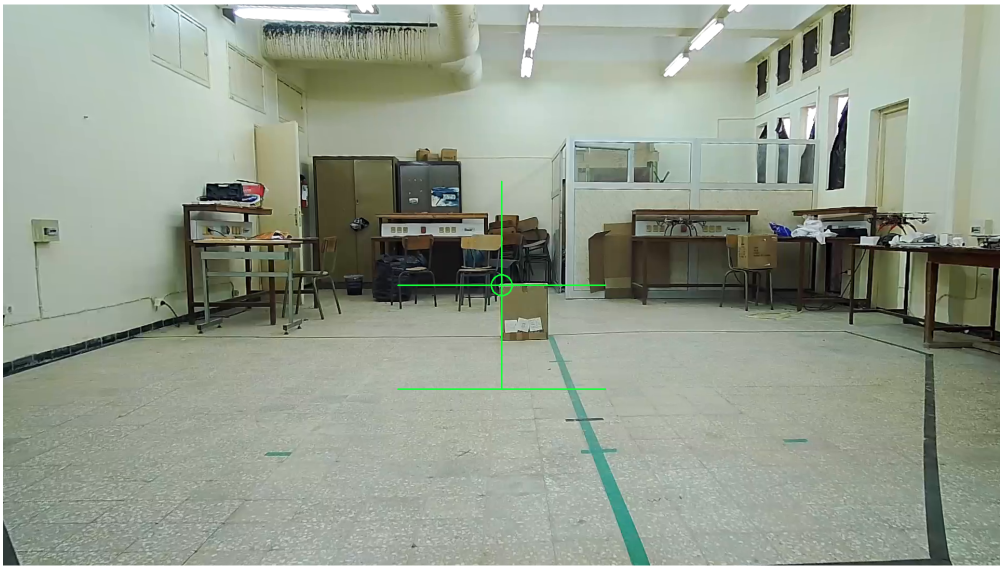
Figure 2.19: an image of the stereo camera pair

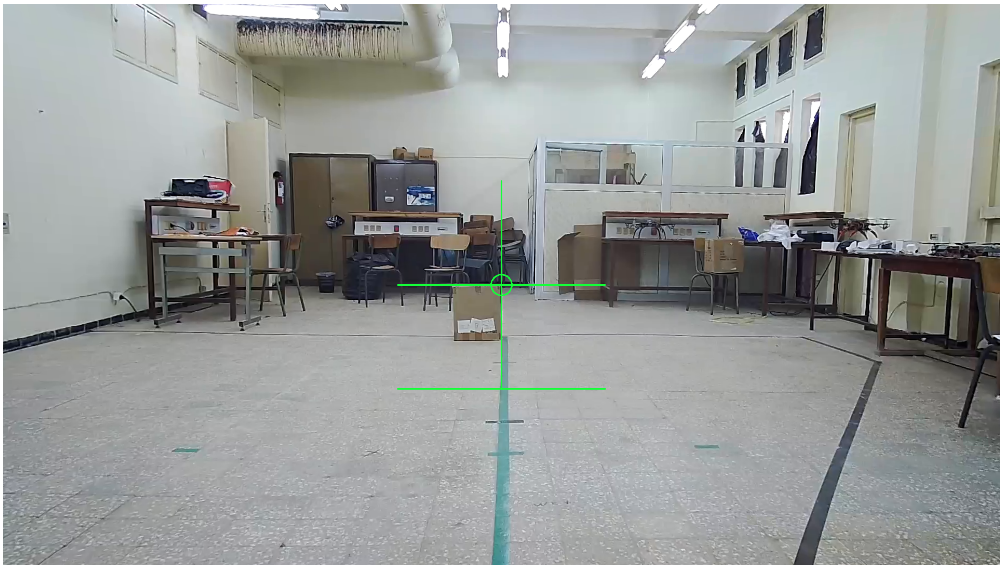
At the outset, we encountered an issue related to latency. The latency was approximately 2 seconds, which contradicted the intended application that demanded relatively real-time estimation., such outdated data would sabotage the drone rather than correcting its navigation systems, we measured this latency By placing the camera's view on the laptop screen beside a stopwatch we run on it, we captured the laptop screen to determine the actual stopwatch time and the corresponding image from the camera. This revealed a noticeable 2-second delay between the laptop's stopwatch and the video stream.

To mitigate that undesired delay, we undertook both hardware and software measures. On the hardware side, we established a connection with the cameras using an Ethernet cable and a hub. This setup reduced latency while also enhancing the stability of the connection by minimizing noise. On the software side, we integrated a specialized streaming library called Gstreamer. This library facilitated the creation of a pipeline for each camera, allowing them to run concurrently. Gstreamer optimized both the latency and quality of the video streams, requiring us only to specify the camera IP while the rest of the optimization occurred automatically. This reduction brought the latency down to nearly 100 ms, which was deemed suitable for our application. After achieving images from both cameras with acceptable latency and stability, we turned to the OpenCV library. OpenCV offers an array of computer vision tools and algorithms that greatly simplify image manipulation. Importantly, OpenCV is compatible with Gstreamer. Following this, we established the center of the lab as the origin and arranged the cameras in a stereo configuration, maintaining a baseline of 36 cm between them. These cameras were situated 3.5 meters apart from the origin, on a table with a height of 0.89 meters. Notably, the right camera was aligned with the origin axis.

Next, we needed to calibrate the cameras' orientations as detailed earlier. This involved placing a box with the same width as the baseline behind the origin, positioned 3 meters away, marking the extent of our operational space. This box was aligned with the origin axis, resulting in a straight line connecting the origin, the box right side, and the right camera.

Afterward, we opened both camera views on the laptop using OpenCV. We added crosshairs to the center of each camera's view since it represents the optical axe, which helped us get the right alignment. Our main goal was to make sure the bottom-right corner of the crosshair in the left image matched the left corner of the box. Similarly, we aimed to align the bottom-left corner of the crosshair in the right image with the right corner of the box , as you can see in figure 2.20. We were very careful with this alignment since it impacts the overall precision of the system.





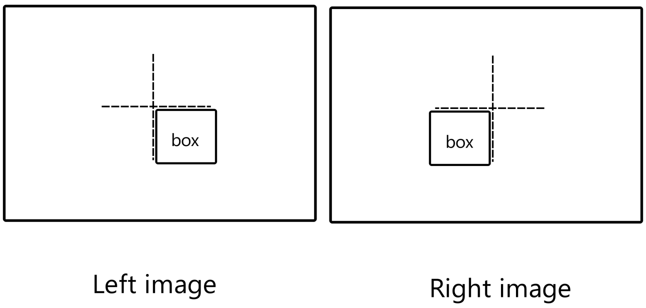


Figure 2.20: align cameras using tier video streams

Ensuring that the cameras' optical axes were perpendicular to the baseline was our next step. With the cameras' positions relative to the lab origin and their orientations fixed, we then focused on determining the angle between the cameras' optical axes and the lab's horizontal reference plane. This task was simplified by computing a vector orientation that incorporated the right camera. To do this, we selected the top-right corner of the box as a second reference point. As a result of this calculation, we arrived at an angle of 4.98°.

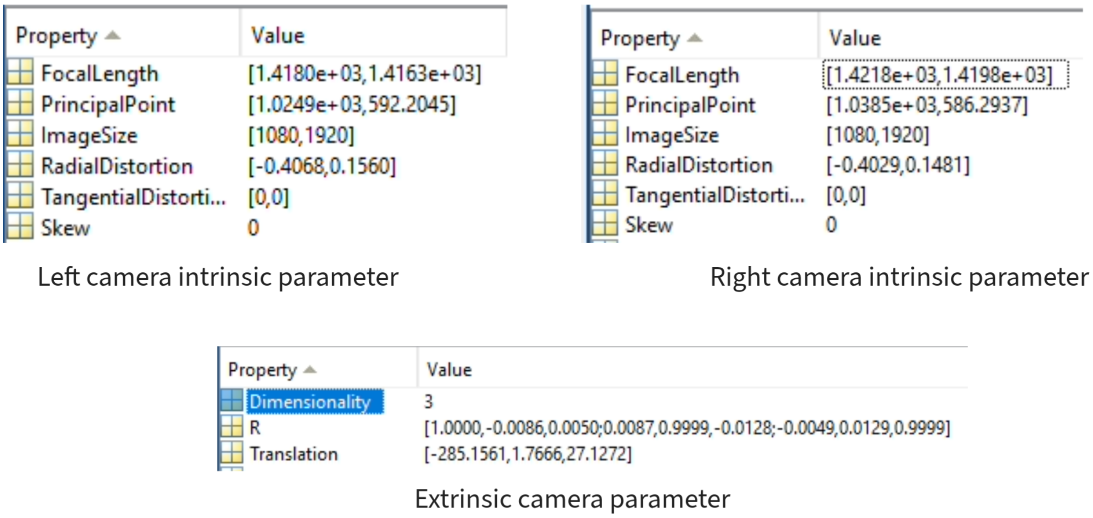
Next, we calibrated the cameras by capturing images of chessboard patterns printed on A3 papers with each camera. Applying the calibration methods we mentioned earlier, we calculated correction matrices to account for any imaging distortions.

Once our setup was ready and calibrated, we began developing our algorithm. This algorithm would play a crucial role in coordinating the cameras and the drone, processing visual data effectively, and driving our project forward. We utilized Python exclusively to code our program on a Linux system, as it's highly recommended for computer vision applications. Python, along with C++, is preferred due to its simplicity, performance, and compatibility with the required libraries. This aligns well with our project's goals. In terms of our algorithm's structure, it can be summarized as follows:

2.5.1 Initialization

We began by loading all the necessary libraries, initializing constant variables and camera parameters ,and establishing camera connections. This crucial step laid the foundation for our algorithm's functionality. Image Acquisition: Our next phase involved connecting each camera to the program using the RTSP protocol via Gstreamer. This configuration ensured the simultaneous reception of frames from both the left and right camera streams. To optimize resource utilization, each stream was placed in a separate processing thread. These threads operated independently from the main program loop, continuously providing frames available in the pipeline at any given moment.This design not only safeguarded the streams' latency against potential delays or errors in the main program but also prevented any such disruptions from affecting the streams' functionality or causing them to shut down. This approach bolstered the reliability and robustness of our image acquisition process.

2.5.2 Calibration process



Property ▲	Value
FocalLength	[1.4180e+03,1.4163e+03]
PrincipalPoint	[1.0249e+03,592.2045]
ImageSize	[1080,1920]
RadialDistortion	[-0.4068,0.1560]
TangentialDistorti...	[0,0]
Skew	0

Left camera intrinsic parameter

Property ▲	Value
FocalLength	[1.4218e+03,1.4198e+03]
PrincipalPoint	[1.0385e+03,586.2937]
ImageSize	[1080,1920]
RadialDistortion	[-0.4029,0.1481]
TangentialDistorti...	[0,0]
Skew	0

Right camera intrinsic parameter

Property ▲	Value
Dimensionality	3
R	[1.0000,-0.0086,0.0050;0.0087,0.9999,-0.0128;-0.0049,0.0129,0.9999]
Translation	[-285.1561,1.7666,27.1272]

Extrinsic camera parameter

Figure 2.21: extrinsic and intrinsic camera parameters

In most instances of camera calibration, the geometric and optical characteristics of the camera, along with its relative positioning in the World Coordinate System (WCS), are typically established through a combination of experimentation and computation. There are three primary methods available for calibrating camera parameters using a calibration target: manual calibration, calibration utilizing the Matlab toolbox, and calibration using OpenCV. Notably, the Matlab toolbox method offers superior accuracy and robustness in stereo calibration when contrasted with manual and OpenCV-based approaches. Consequently, the calibration employs the Bouguet algorithm from the Matlab calibration toolbox, with the resultant outcomes then imported into OpenCV for subsequent image correction.

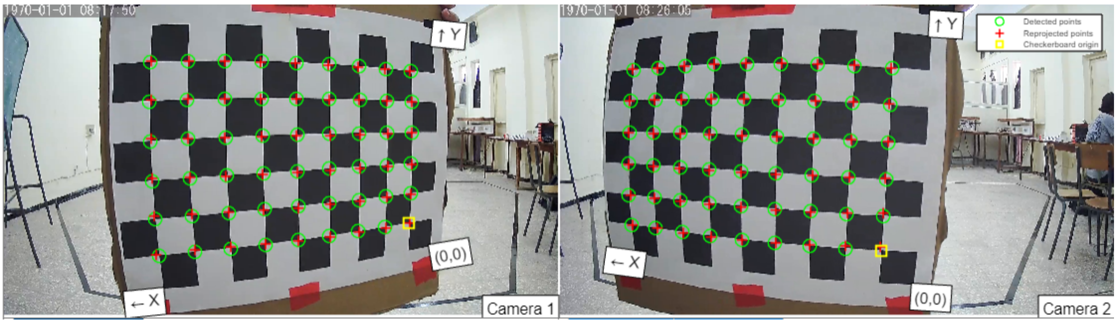


Figure 2.22: Matlab stereo camera calibrator

2.5.3 Binocular image correction

Within the context of binocular correction, the tasks of distortion elimination and line alignment are executed for both the left and right views. This process is guided by the monocular internal parameters, such as focal length, imaging origin, and distortion coefficients. Additionally, the relative positional relationships, encompassing rotation matrices and translation vectors derived from camera calibration. These adjustments aim to achieve several key outcomes: aligning the imaging origins coordinates for both views, ensuring parallelism of the optical axes of the two cameras, establishing coplanarity of the left and right imaging planes, and aligning epipolar lines. With the aforementioned calibration parameters as a foundation, the necessary correction parameters are acquired utilizing the cvStereoRectify function within the OpenCV framework. These acquired parameters are then employed to rectify the input images for both the left and right perspectives using the cvRemap function.

2.5.4 Post-processing images

Once we acquired the images, our focus shifted to correcting the distortion effect. This was achieved using the expression we derived during the discussion of distortion effects, in conjunction with the correction matrices we previously calculated. Subsequently, we resized the images to a resolution of 640x640. This resolution aligns with the specifications of the YOLOv5 model, which our system was built around. All of these adjustments were performed utilizing the OpenCV library. This stage of post-processing was instrumental in preparing the images for further analysis and detection using our chosen model.

2.5.5 Performing object detection

After processing the images as described, we proceed to feed them individually to the YOLOv5 model loaded through OpenCV. This model, which we previously trained, conducts object detection on each input image. This results in the identification of the drone within the image, alongside the generation of a bounding box encompassing the detected drone. Extracting the drone's center coordinates, height, and width from this process, we focus specifically on the horizontal coordinates.It's worth noting that object detection is computationally intensive and can significantly slow down the system, particularly during image processing and calculation phases. To mitigate this challenge, we utilized a laptop equipped with a powerful graphics card (RTX 3060). This graphics card, with its enhanced computational capabilities, coupled with the CUDA library, facilitated the execution of the object detection process on the graphics card rather than the CPU. This efficient approach effectively eliminated delays between data acquisition and object detection, preserving a consistent frame rate.

2.5.6 Measuring the distance with disparity

The task involves establishing the connection between the disparity value and the corresponding distance. This disparity is computed by taking the difference in horizontal coordinates of the bounding box center in the images from the right and left perspectives. To accomplish this, we conducted empirical measurements of disparity values at various points and utilized Excel software to derive a regression model.

2.5.7 Depth estimation

With the coordinates of the detected object centers in hand, we direct them to one of the algorithms we previously discussed. This algorithm performs the crucial task of estimating the distance between the object and the drone. It's essential to note that the calculated distance is variable, subject to fluctuations due to the inherent instability of detection, especially when one of the images or both, fails to detect the object. Even when maintaining the object's position, the detection outcome can still exhibit slight variations. These variations significantly impact the accuracy of depth estimation. To overcome this challenge, we introduced an alpha filter to the obtained centers. This filter employs an initial value derived from the object's position when it's located at the origin. A threshold of 0.09 is applied to the alpha filter .Mathematically, the process can be represented in (2.31) :

$$\begin{aligned} C_{L_n} &= 0.09 C_{L_n} + (1 - 0.09) C_{Ln-1} \\ C_{R_n} &= 0.09 C_{R_n} + (1 - 0.09) C_{Rn-1} \end{aligned} \tag{2.31}$$

where C_R and C_L are the horizontal center of the detected object from the right and the left image respectively.By implementing this approach, we enhance the stability of our depth estimation. This accounts for the inherent fluctuations in the detection process and contributes to more consistent and reliable depth calculations.

2.5.8 Position estimation

Having determined the distance, we advance to position estimation. This involves following the methodology steps we discussed earlier, incorporating the calculated distance to determine the precise position of the drone relative to the cameras.

2.5.9 Display and send results

Ultimately, we showcase the results on the right camera's stream, which we selected as our reference in this approach. This involves drawing a bounding box around the detected drone and supplementing it with the corresponding world coordinates. This visual representation offers a clear and immediate understanding of the drone's position within the environment.Finally, as the estimated position is ascertained, our system fulfills its primary purpose by wirelessly transmitting this vital information to the drone within the indoor environment. This step is pivotal, enabling the drone to navigate adeptly within the indoor space, guided by real-time feedback from our system.

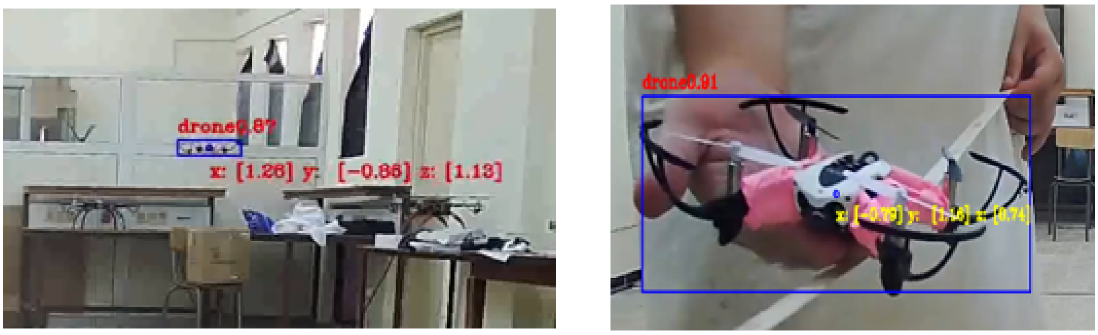

drone0.87
x: [1.26] y: [-0.86] z: [1.13]
drone0.91
x: [-0.79] y: [1.16] z: [0.74]

Figure 2.23: capture of the positioning system output

2.6 System evaluation

To comprehensively evaluate the effectiveness and performance of the developed indoor positioning system, a meticulous evaluation process was executed. This assessment primarily centers on core variables and estimation sources, particularly the coordinates derived from the cameras and the depth estimation methodology used to determine the distance between camera pairs and detected objects. The accuracy of the system's coordinates remains contingent upon the efficacy of the depth estimation method employed, as the camera pair configuration remains unchanged.

To establish quantifiable measures for comparison against the estimated values, we conducted measurements of distances between the center of the baseline and randomly selected points across the operational space. This measurement span ranged from 0.5 meters to 5 meters, accomplished using a tape measure Figure 2.24.

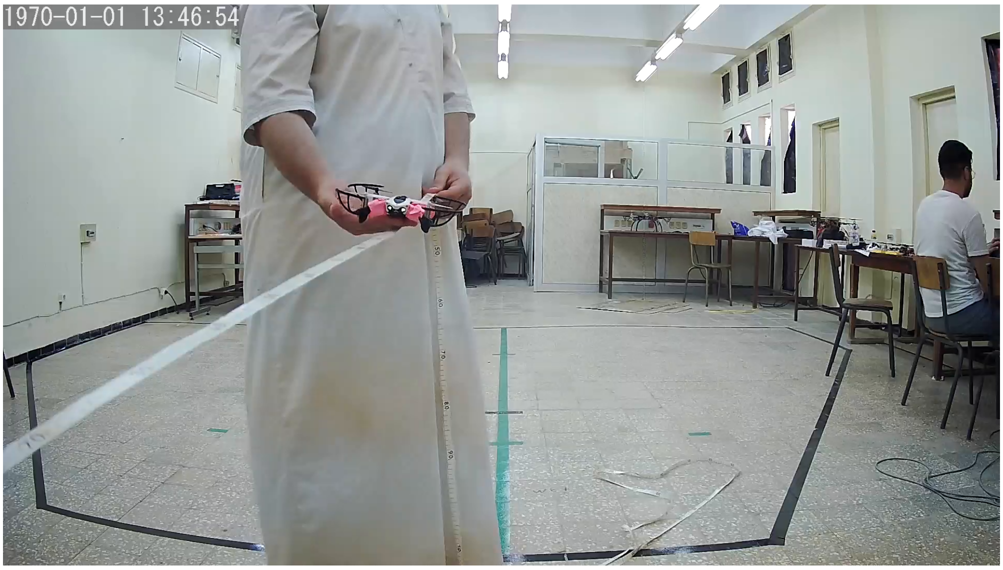


Figure 2.24: Taking measures to evaluate the system

With these measured distances in hand, we then proceeded to calculate the positions within the laboratory frame. Using the series of translation steps we have seen previously, allowing us to map the measured distances onto the lab's spatial coordinates. This process enabled a comprehensive evaluation of the system's accuracy and alignment with real-world distances across the designated operational range.giving us the following values in table 2.1

In order to facilitate a comprehensive understanding and to clearly differentiate between the two methods employed in our study, we have chosen to supplement our discussion with a series of figures 2.25 - 2.31.These visual aids are valuable tools that help illustrate the subtle differences and variations between the methods in a more easily understandable way.

distance (m)	Δe with triangulation (cm)	Δe with disparity (cm)
0.5	1	11
1	2	6
1.5	3	8
2	5	13
2.5	7	29
3	11	28
3.5	12	94
4	16	81
4.5	15	157
5	21	111

Table 2.1: average error in depth estimation

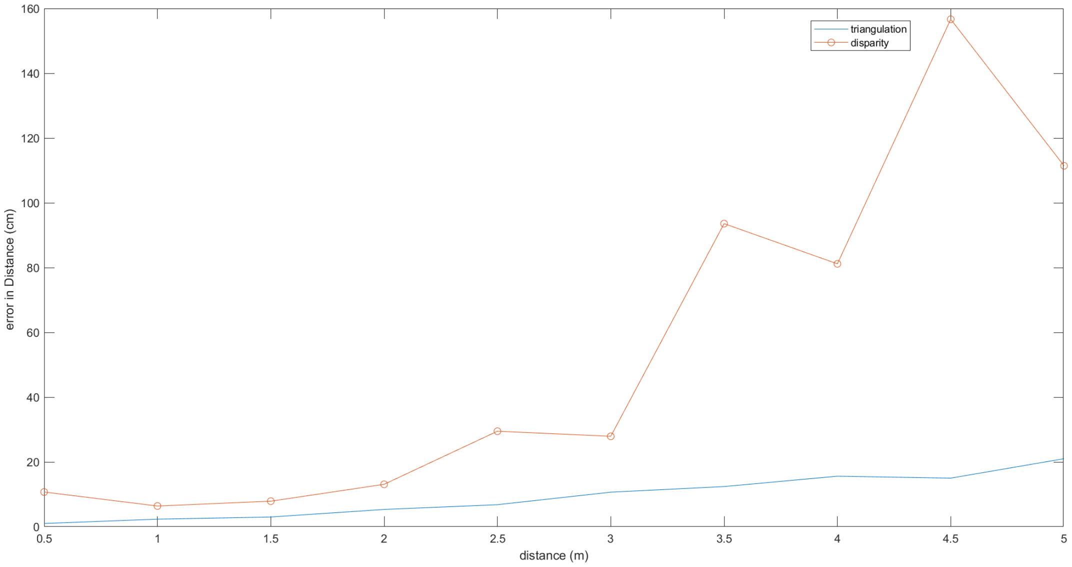


Figure 2.25: error in depth estimation

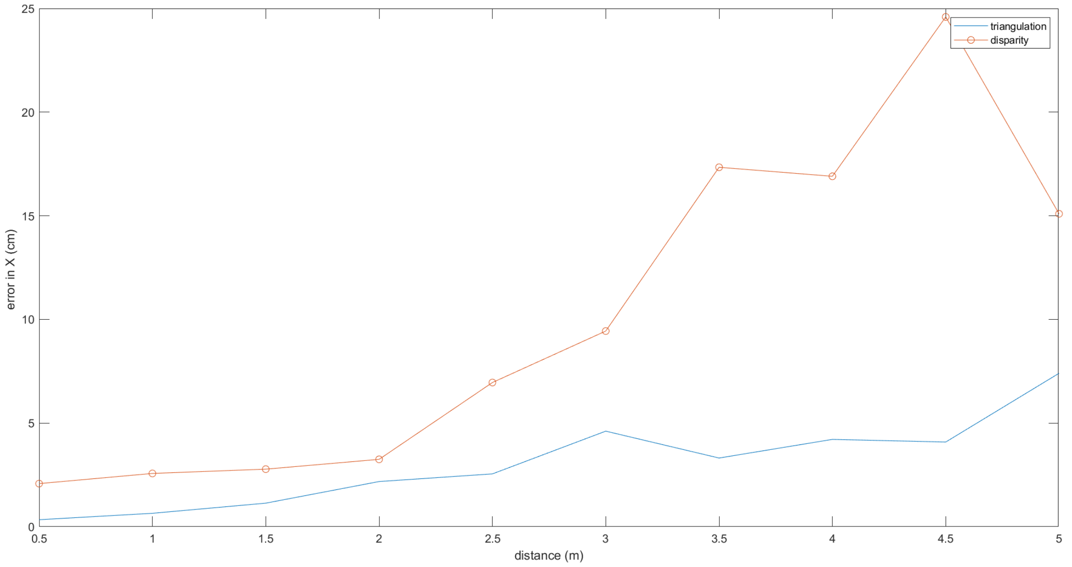


Figure 2.26: error in position estimation X

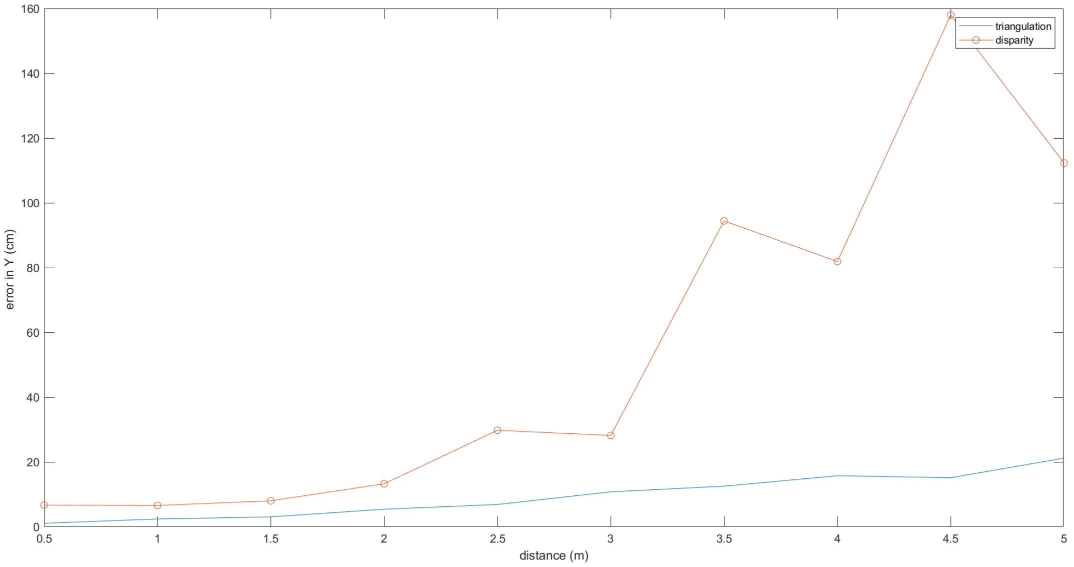


Figure 2.27: error in position estimation Y

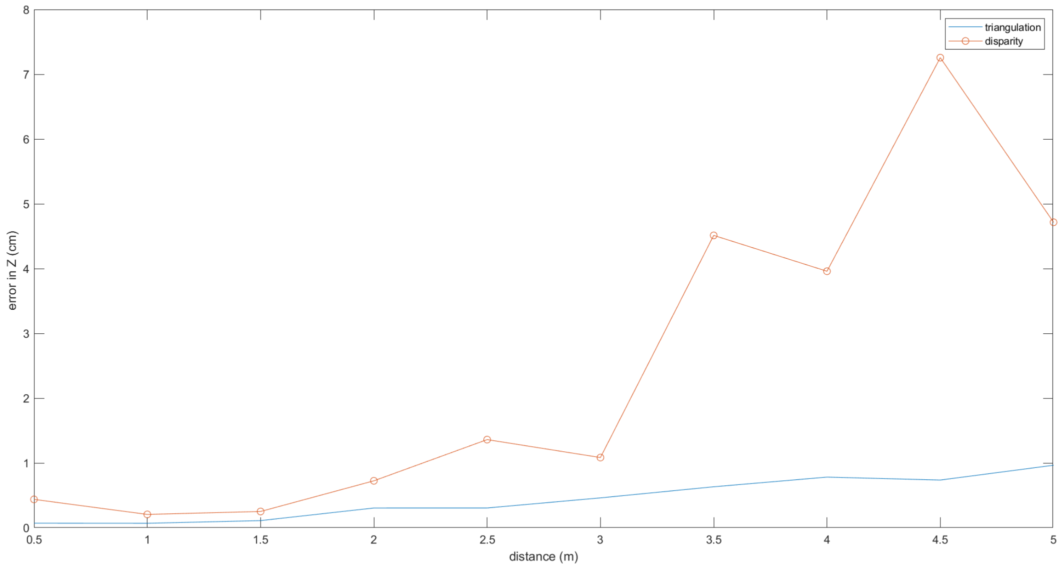


Figure 2.28: error in position estimation Z

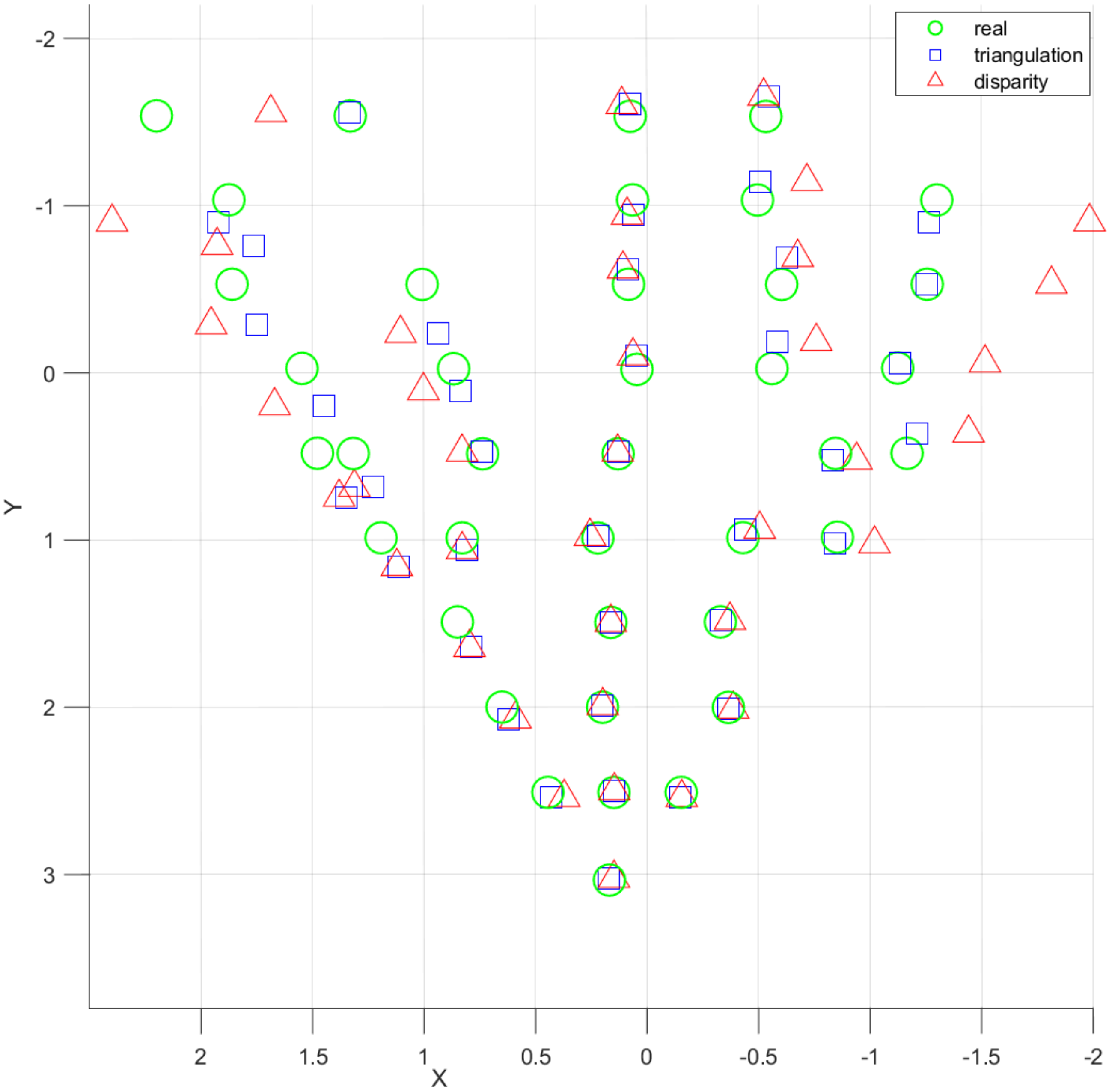


Figure 2.29: 3d points of position estimation on XY plane

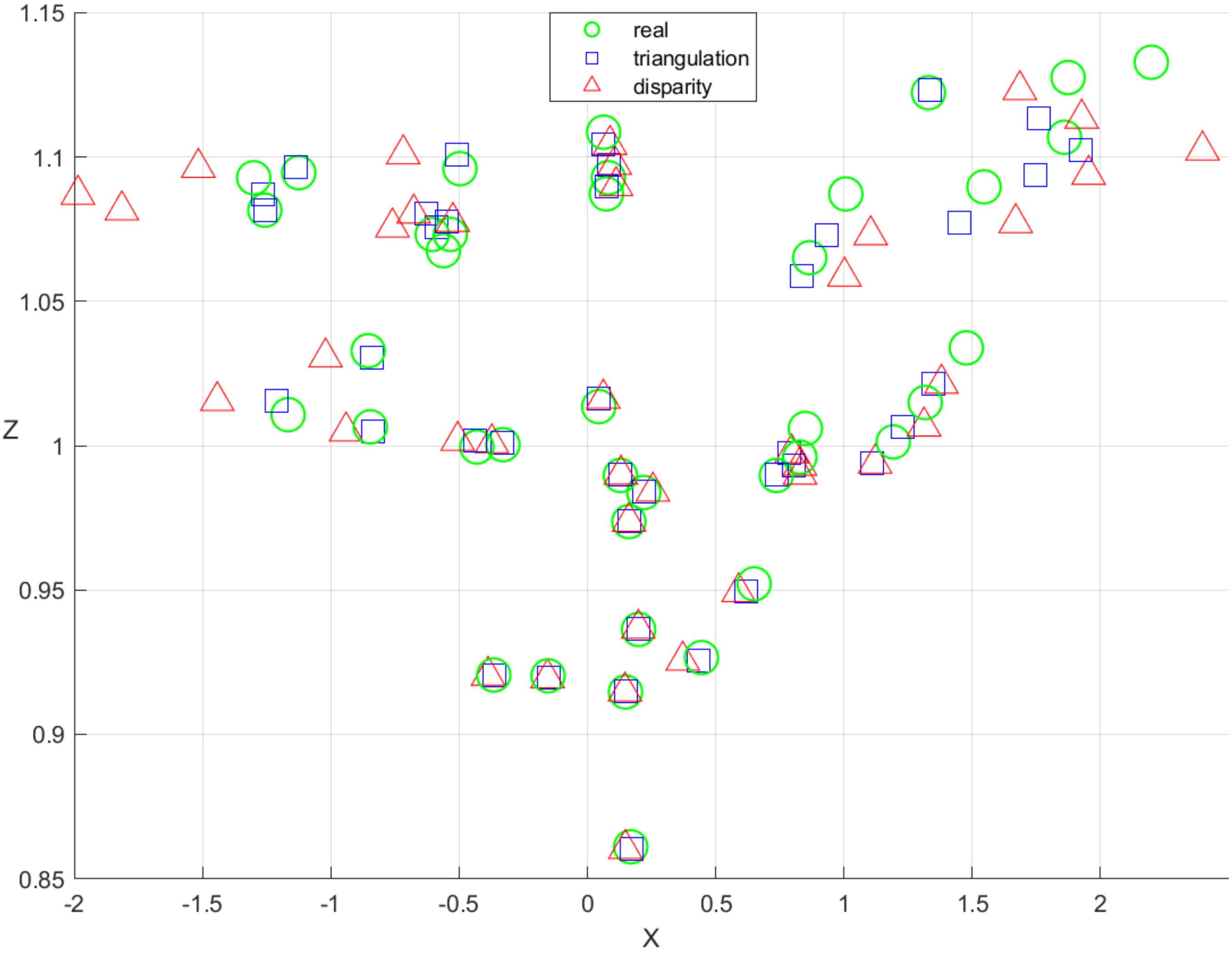


Figure 2.30: 3d points of position estimation on XZ plane

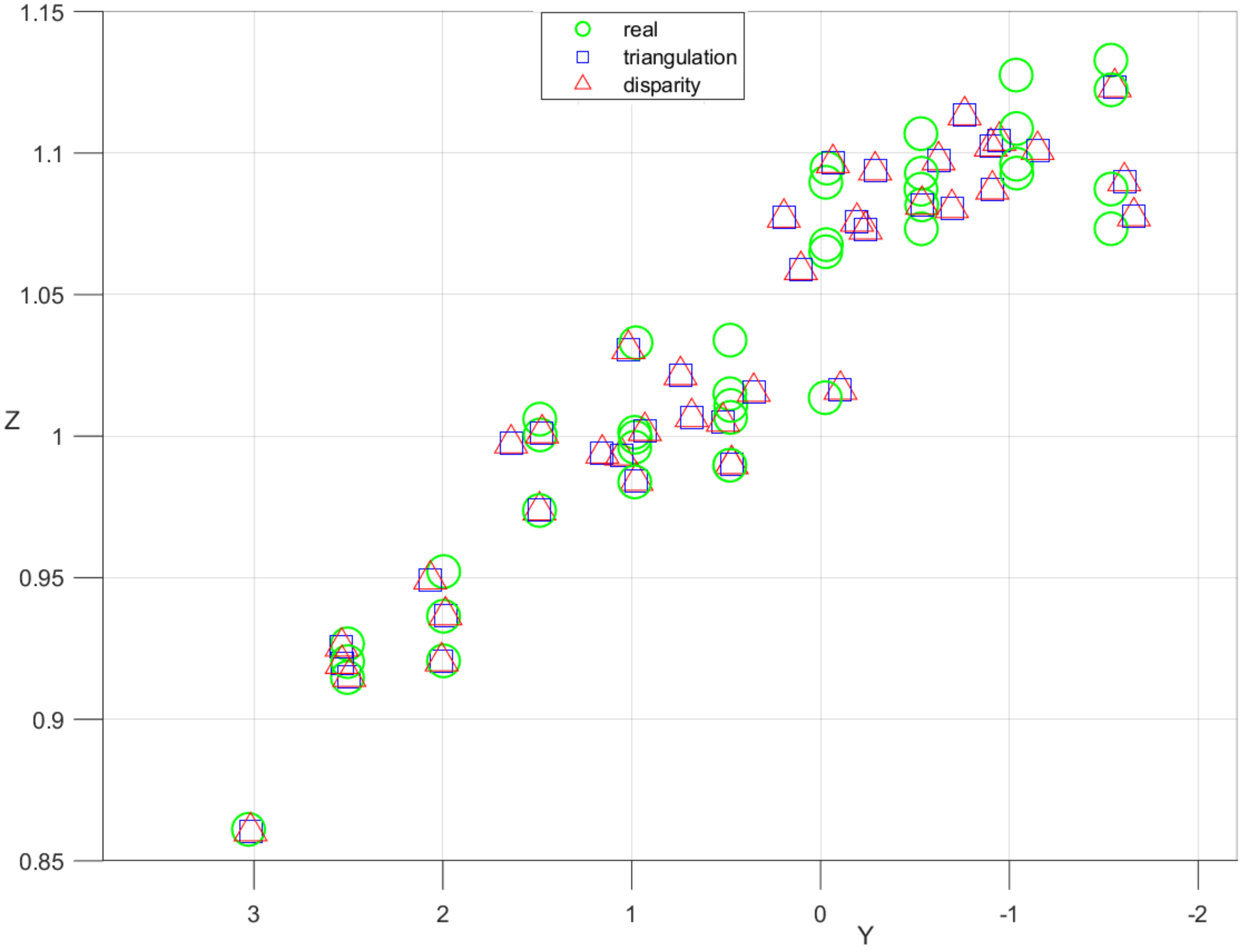


Figure 2.31: 3d points of position estimation on YZ plane

2.7 Discussion

The obtained results from our comprehensive evaluation offer promising insights into the potential of the system we have developed. This system, designed for indoor navigation and position estimation, demonstrates itself as a valuable and cost-effective solution. Notably, the results underscore the viability of the triangulation method over the disparity method. This distinction is attributed to the disparity method's heightened dependence on the stereo configuration and the general performance of the utilized cameras. Given that our system employs an emulation of a stereo camera setup, it becomes evident that adopting a prebuilt stereo camera with established calibration would greatly enhance the efficacy of the disparity approach. On the contrary, the triangulation method's efficacy rests upon the precise estimation of angles from each camera's viewpoint, coupled with the accuracy of object detection – factors that together contribute to mitigating errors in the positioning process.

Furthermore, our analysis highlights a common trend in both methods, wherein precision experiences a decline with increasing distance between the detected object and the camera pair. This phenomenon can be attributed to a fundamental principle inherent in stereo configurations [22], as an object moves farther away from the cameras, the corresponding disparity diminishes Figure 2.32. Consequently, this reduction in disparity can impede the system's ability to accurately detect changes in the object's position or orientation, particularly as the object recedes farther from the camera pair.

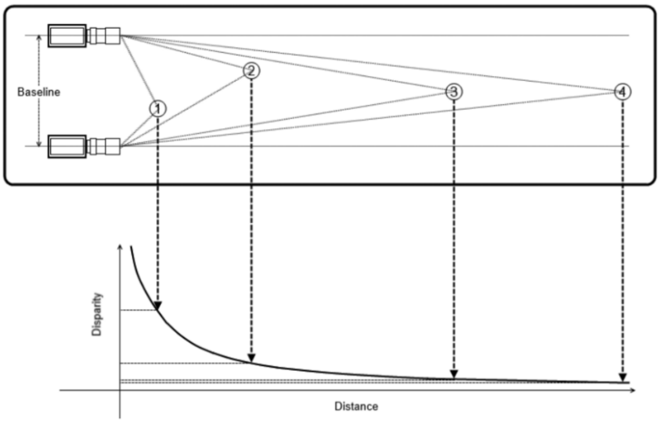


Figure 2.32: Depth and disparity are inversely related, so precise depth measurement is restricted to nearby objects

2.8 Conclusion

In conclusion, the development and evaluation of the indoor positioning system have yielded valuable insights into its performance and potential applications. While the present iteration of the system presents promising outcomes, avenues for refinement persist. Notably, there exists significant potential for substantial enhancements in both performance and precision. Achieving this leap forward hinges on optimizing hardware, introducing supplementary position estimation sources for heightened redundancy, and refining the algorithmic framework. The implications of this work extend beyond indoor positioning, potentially benefiting various domains such as robotics, navigation, and augmented reality applications. The successful combination of advanced measurement techniques, translation processes, and depth estimation methods underscores the potential impact of the system on addressing real-world positioning challenges. In essence, this endeavor marks a significant stride toward achieving reliable indoor positioning. As technology continues to evolve, the insights gained from this work provide a solid foundation for further advancements in spatial estimation and its diverse applications.

Chapter 3

Multi UAV formation control

3.1 Introduction

In recent years, the proliferation of unmanned aerial vehicles, commonly referred to as drones, has revolutionized various industries by enabling new perspectives and possibilities. Drones have transcended their initial role as mere recreational gadgets and have become powerful tools with transformative potential. As technology advances, the application of drones has evolved from singular and isolated operations to collaborative and orchestrated endeavors involving multiple units, giving rise to the concept of multidrone applications. The idea of employing multiple drones in coordinated activities has captivated the imagination of researchers, industries, and enthusiasts alike. This paradigm shift holds the promise of enhancing efficiency, scalability, and versatility across various domains. From precision agriculture to disaster response, from entertainment spectacles to autonomous delivery systems also Surveillance, multidrone applications have the potential to reshape the way we perceive and interact with our surroundings.This chapter aims to simplify the key concepts behind building a multi-drone system and provide a groundwork for future research. Additionally, it presents an initial approach to creating a two-drone system for detecting wildfires.

3.2 Related work

The field of multidrone applications has garnered significant interest in recent years, with researchers and industries exploring various aspects of coordinated drone operations. A review of related literature reveals the diverse range of applications, challenges, and solutions that have been investigated. Researchers have drawn inspiration from natural swarm behaviors observed in animals to develop coordination algorithms for drone swarms [25]. Studies by Hussein et al [26] explored algorithms for decision-making in drone swarms using machine learning, highlighting the importance of robust communication protocols and adaptive strategies. The agricultural sector has embraced multidrone systems for tasks such as crop monitoring, irrigation, and pesticide distribution, as shown in [27], where employing a swarm of drones has proven to enhance both efficiency and effectiveness, leading to reduced mission durations. Multidrone systems have shown potential in disaster management and surveillance scenarios, as reviewed in [28]. The entertainment industry has harnessed the visual impact of drone formations for captivating aerial displays, as explored by Dharna N et al [29], emphasizing the need for precise positioning and timing control. In the realm of autonomous deliveries, Balsam et al [30] introduced a groundbreaking swarm delivery service, exemplifying the substantial time reductions achieved through the implementation of their innovative system. Furthermore, in the context of environmental monitoring, Fabrice.S et al [31] proposed a framework for using drone swarms to detect wildfires in remote areas, integrating real-time image processing with thermal sensors to identify heat signatures indicative of fire outbreaks, and encompassing the containment and monitoring of identified wildfires. The aforementioned studies demonstrate the breadth and depth of research in multidrone applications, but there remains room for further exploration, particularly in addressing scalability, communication reliability, energy efficiency, and integration with existing infrastructure. The following sections of this chapter aim to build upon these foundations by presenting a systematic breakdown of constructing a multi-drone system and showcasing an initial approach to wildfire detection using a two-drone setup.

3.3 Multi-drone system architecture

Building a drone swarm starts with each drone as a basic unit. These drones are equipped with their own abilities. The exciting part happens when they work together, showing swarm behavior. While creating a drone swarm involves many aspects like communication, control, and patterns, it's important to remember that everything begins with designing capable individual drones. The way they act together is the result of careful research, and studying swarm behavior is just a part of the whole process [32].Once we ensure that we have stable drones suitable for operating within a swarm, several architectures come to the forefront As follows. An intricate communication network forms the backbone of the swarm's architecture that depends on the application and the mission seeked. This network serves as a conduit for disseminating commands, sharing data, and maintaining situational awareness among the drones.Xi Chen et al [33] classified the types of drone swarm network architecture into two categories as outlined below

3.3.1 Centralized communication architecture

The method of communication between drones has evolved over time. Initially, individual drones used a basic communication system. However, as groups of drones began collaborating, a new communication approach emerged. Think of a group of drones depicted in Figure 3.1. In this arrangement, there's a central point where all the drones connect, that acts as a ground control station . Each drone communicates with this GCS and receives its instructions from there. This setup is reliable and straightforward. It's well-suited for smaller drone groups operating in limited areas and performing uncomplicated tasks.

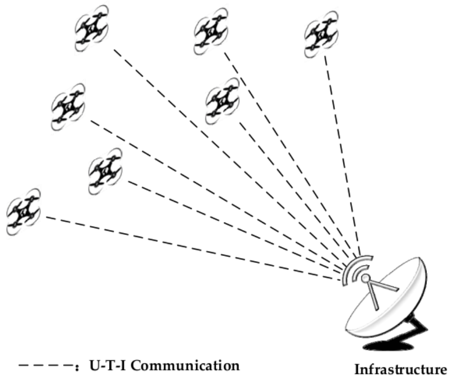


Figure 3.1: Schematic depicting the centralized communication architecture where (U-T-I) is communication between the UAV and the infrastructure

This communication approach is referred to as "centralized communication architecture." It's akin to having a leader who guides everyone's actions. This method is easier to manage, especially when the group isn't too large or the tasks aren't overly complex. An example of its use is in crowd surveillance, where drones collaborate to monitor large gatherings of people. nonetheless, a significant limitation emerges due to the infrastructure requirement for inter-UAV connections. The pronounced disparity between the distances from UAVs to the infrastructure and between UAVs themselves results in prolonged transmission delays. This challenge is further exacerbated by the dynamic mobility of UAVs and the extensive coverage mandates of swarm applications, ultimately compromising the architecture's stability.

Moreover, the architecture's reliance on infrastructure introduces a vulnerability factor. The operational state of the ground station or satellite becomes pivotal, as any disruption to these central nodes triggers a network-wide breakdown. This susceptibility translates into the architecture's SPOF drawback. Consequently, the centralized communication architecture's reliability is compromised, with its susceptibility to disruption classifying it as an unreliable communication model.

3.3.2 Decentralized communication architecture

In a "single-group swarm Ad hoc network" (illustrated in Figure 2), the internal communication within the drone swarm operates independently, separate from external infrastructure. Communication between the swarm and this external infrastructure occurs via a singular connection point, facilitated by a designated "gateway UAV." In this structure, other UAVs serve as relay nodes, forwarding data within the swarm. This method enables real-time sharing of situational information among swarm UAVs, optimizing collaborative control and operational efficiency. Similarly, the gateway UAV communicates with the infrastructure, allowing the upload and download of swarm information, including instructions. This requires the gateway UAV to possess distinct transceivers: one for close, low-power communication with other UAVs and another for long-range, high-power communication with the infrastructure.

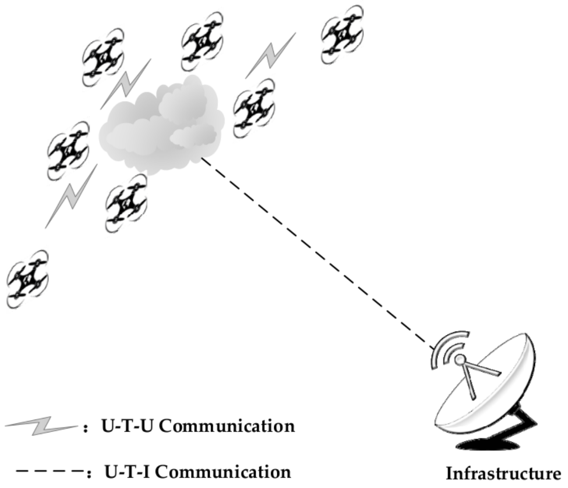


Figure 3.2: Schematic showing a single-group swarm Ad hoc network where (U-T-U) is communication between the UAV and other UAV

In the pursuit of mission-oriented optimization, the landscape of intra-swarm communication architecture has experienced iterative transformations. Notably, Figure 3.3 exemplify three prevalent models that have emerged.Figure 3.3a portrays the "ring architecture," wherein UAVs configure a closed loop of communication through bidirectional links. Any UAV within this arrangement can assume the pivotal role of a gateway node for the entire swarm. In case a direct link between adjacent UAVs falters, the affected UAV can re-establish connection via the communication loop. This ring architecture imparts a certain degree of stability. However, its scalability is compromised, indicating an intrinsic limitation.

Conversely, Figure 3.3b demonstrates the "star architecture," positioning the gateway UAV centrally. This gateway not only interfaces with the infrastructure but also acts as a linchpin of communication across the entire UAV swarm. Evidently, the star architecture is susceptible to a SPOF vulnerability. In the event of gateway node failure, the entire system succumbs to collapse.

A harmonious amalgamation of the ring and star architectures materializes as the "meshed architecture," represented in Figure 3.3c. This configuration encapsulates the merits of both systems. All UAV nodes within the swarm assume uniform capabilities, seamlessly blending terminal and routing functionalities. Data transmission between nodes adopts diverse pathways, with any UAV capable of assuming the role of a gateway node. Remarkably, the meshed architecture emerged as the de facto standard for intra-swarm communication systems, standing as a testament to its efficacy and versatility.

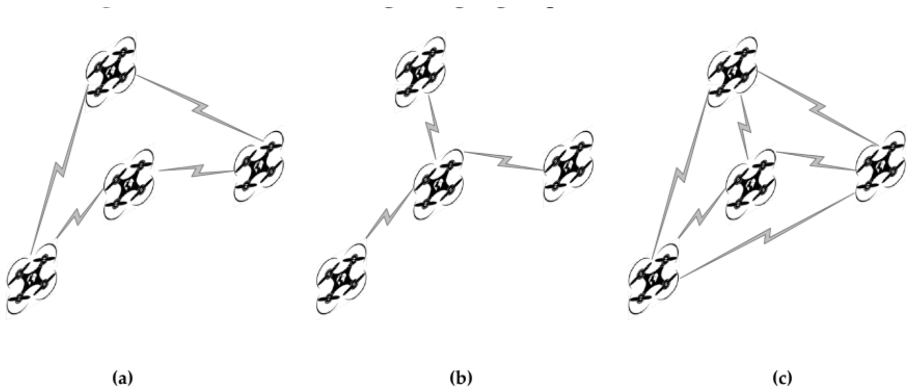


Figure 3.3: Intra-swarm communication architecture: (a) ring architecture, (b) star architecture,(c) meshed architecture

3.4 Initial approach proposed

3.4.1 Architecture

Recognizing the multiple benefits of utilizing drone swarms across various applications, we introduced an initial approach for our fire detection application. We employed a formation involving two drones, with a centralized leader-follower arrangement. This decision was based on the method's simplicity and the smaller number of drones needed. Our aim was to devise an algorithm where a couple of leading drone execute its tasks, while the follower drone maintain a constant distance. This coordination was achieved using computer vision and FPV cameras. It's important to note that this approach is well-suited for indoor environments, capitalizing on the indoor positioning system we introduced in Chapter 2.

A laptop will function as the GCS in this setup, playing a crucial role in coordinating the drone formation. It will take on several key responsibilities: first, it will calculate and estimate the leader drone's position using the positioning system, effectively replacing the need for traditional GPS and thereby enhancing stability.

The laptop will also serve as a mission planner, enabling the GCS to designate a specific target destination and way points for the leader drone to reach.

Furthermore, the GCS will establish communication with the leader drone through a radio link, utilizing the Pixhawk controller. This robust link will facilitate real-time data exchange, allowing the GCS to send vital information to the leader drone. Additionally, the GCS will receive a continuous video stream from the follower drone's perspective, leveraging object detection algorithms to analyze the video feed. Based on this analysis, the GCS will issue precise instructions to the follower drone, ensuring it maintains a consistent relative position with the leader drone.The follower drone, on the other hand, will establish its connection with the GCS via WiFi.

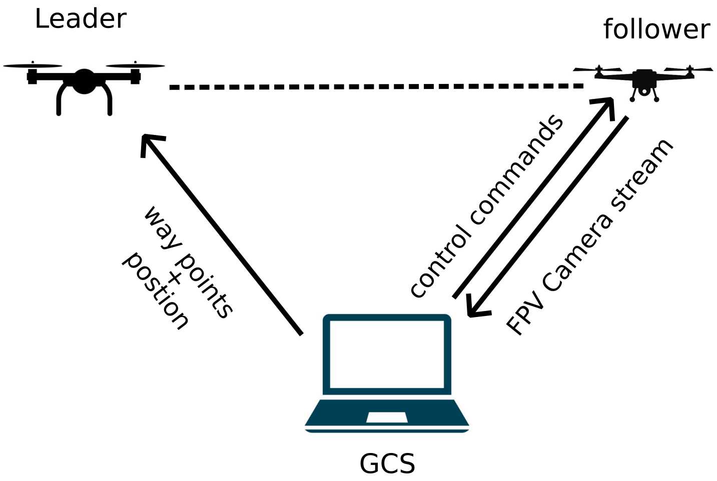


Figure 3.4: centralized leader-follower proposed architecture

3.4.2 Drones used

3.4.2.1 Pixhawk drone

We've successfully developed a quadcopter equipped with the PIX Hawk autopilot system, known for its reliability and precision, ensuring stable and autonomous flight capabilities. This advanced drone is easily controlled through a user-friendly interface accessible via a dedicated mobile app or ground control station. With intuitive controls and real-time telemetry data, users can effortlessly plan flight routes, set waypoints, and execute autonomous missions. The quadcopter offers both manual and autonomous flight modes, making it versatile for various applications, including aerial photography, surveillance, and research. Whether you're a beginner in need of a dependable drone or an experienced pilot seeking top-tier performance, our quadcopter with the PIX Hawk autopilot system has you covered.

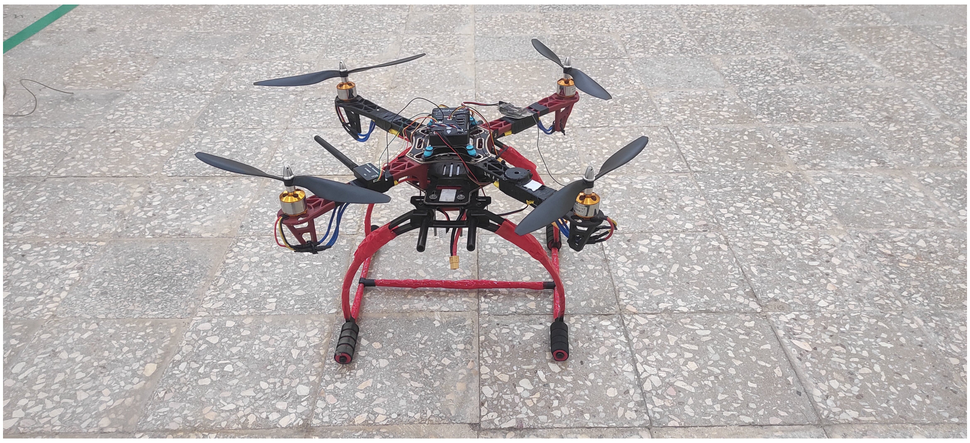


Figure 3.5: image of Parrot Pixhawk drone

3.4.2.2 Naza drone

We are thrilled to introduce our latest creation,a quadcopter featuring the renowned Naza M-Lite autopilot system. With the Naza M-Lite, we've achieved an exceptional level of stability and precision in our quadcopter's flight capabilities. The drone is effortlessly controlled via a user-friendly remote control unit, ensuring smooth maneuvering. The Naza M-Lite's intelligent flight modes and GPS-based functions provide an outstanding flying experience, including safety features like Return-to-Home. Whether you're a beginner seeking a stable and responsive drone or an experienced pilot looking for professional-grade performance, our quadcopter with the Naza M-Lite autopilot system is designed to meet your needs, delivering a seamless and enjoyable flying experience for all skill levels.

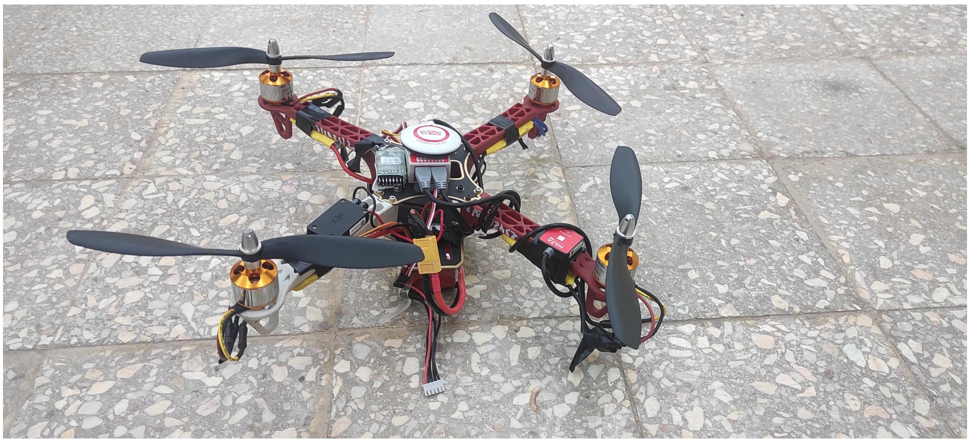


Figure 3.6: image of Parrot Naza drone

3.4.2.3 Parrot mambo

The Parrot Mambo Fly is an affordable mini quadcopter drone produced by Parrot, a French drone manufacturer. When equipped with the optional FPV camera, it is referred to as the Parrot Mambo FPV. This camera uses the RTSP protocol, simplifying image acquisition via Wi-Fi. The Mambo Fly is an excellent choice for development due to its automatic stabilization system and user-friendly controls. It offers connectivity options via both Wi-Fi and Bluetooth using the PyParrot library with Python, making it well-suited for our workflow.

Figure 3.7: image of Parrot Mambo mini drone

3.4.3 Drone detection

Among the numerous object detection models available, our choice of YOLOv5 stems from its exceptional balance between performance and precision. This framework strikes a remarkable equilibrium that aligns with our requirements. Notably, YOLOv5's seamless integration with OpenCV enhances its compatibility and usability within our workflow.One key driving factor behind our selection is YOLOv5's foundation on Ultralytics. This platform equips YOLOv5 with an intuitive and user-friendly interface for both training and evaluation. This ensures a clear and straightforward process in harnessing the model's capabilities.

After selecting an object detection model, our focus shifted to training it for the specific task of drone detection. Initially, we gathered images of the drone we intended to detect, capturing them from various angles and distances. These images were meticulously labeled with corresponding bounding boxes, accurately marking the drone's presence within each image.

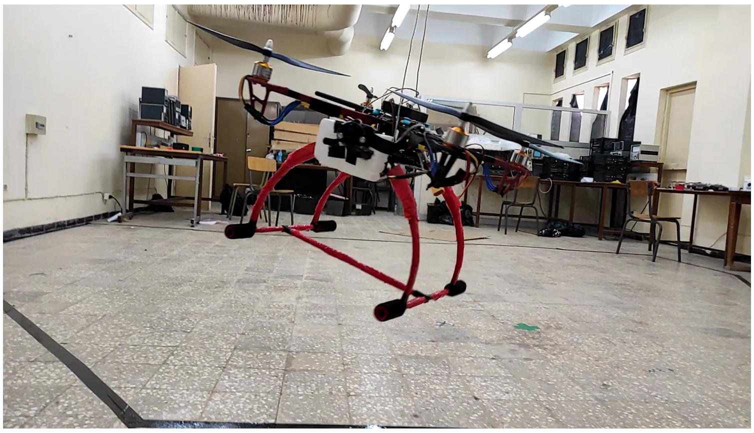


Figure 3.8: example of the images acquired for training

Subsequently, we employed data augmentation techniques to enhance the diversity and robustness of our dataset. Applying rotations and flips (both horizontal and vertical) to the collected images, we significantly increased our dataset size, amassing a total of 12,000 labeled images.

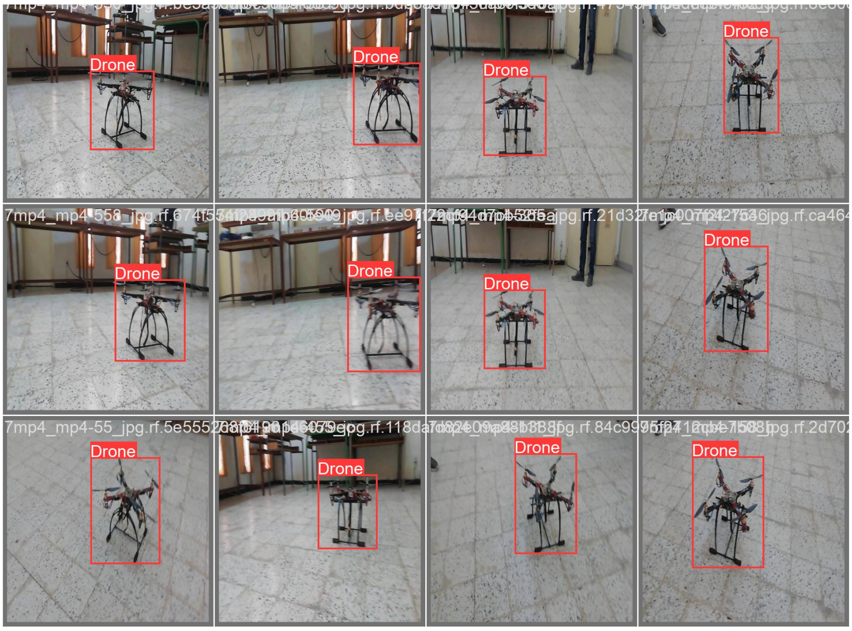


Figure 3.9: example of the labeled images

To prepare for the training process, we partitioned our dataset into three subsets: 85% for training, 10% for validation, and 5% for testing. This allocation, combined with the substantial number of images, proved adequate for our purposes. Given the single-drone scenario and the consistent environment, this setup and dataset size are deemed sufficient to achieve accurate detection results.

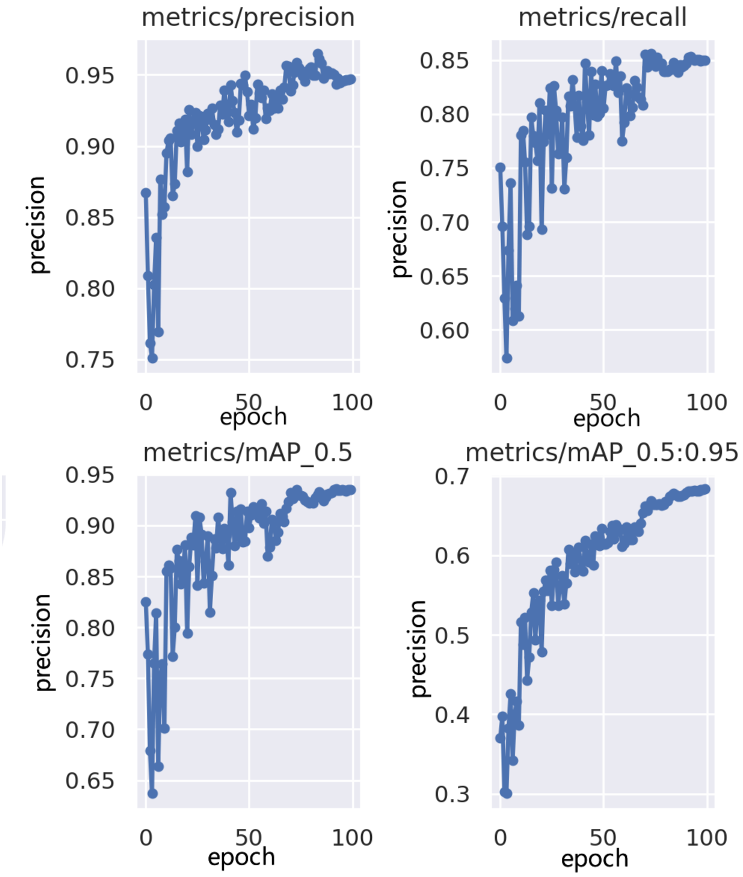


Figure 3.10: training results of the YOLOv5 model of our custom dataset

The figure's results demonstrate the success of our training process. Following 100 epochs of training, the model achieved a commendable precision of around 0.9. This level of precision surpasses our application's requirements, reaffirming the effectiveness of our model for the intended task.

3.4.4 Leader follower algorithm

In our two-drone swarm formation, we've established a well-coordinated system. The leader drone, for which we've trained our object detection model, communicates with the Ground Control Station (GCS) via a radio link. It receives real-time updates about its position in the indoor environment and the navigation waypoints it needs to follow. Conversely, the follower drone communicates over WiFi, transmitting a continuous video stream captured by its FPV camera to the GCS. Here, the GCS plays a pivotal role by performing object detection on this incoming video stream. From each frame, we extract essential parameters of the detected leader drone, primarily focusing on its height. These parameters form the foundation of our tracking algorithm. This algorithm is equipped with six distinct commands: left, right, up, down, backwards, and forwards. Each command has two speeds, which will eventually translate into specific drone actions.

To facilitate tracking, we transform the pixel coordinates of the leader drone's center (Cd) into film coordinates, with the image center serving as the origin point. Around this origin, we establish a circular region, the size of which is determined by our desired tracking sensitivity.Additionally, we define four crucial reference lines: two vertical and two horizontal. These lines are positioned between the circular region and the image edges as lustreated in Figure 3.11.

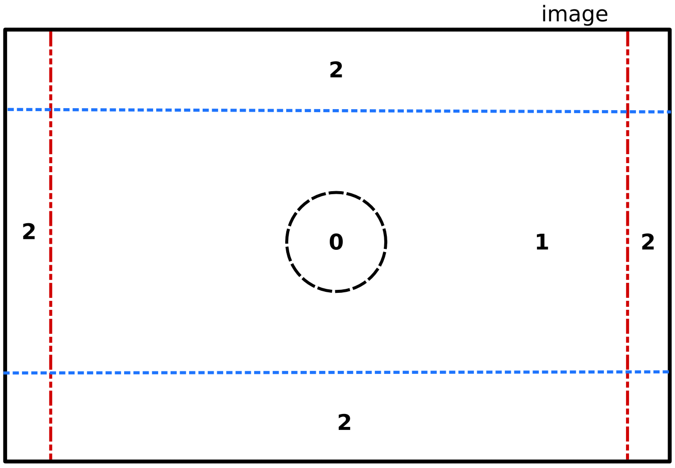


Figure 3.11: illustration of the decision zones relative to the image received from follower drone

We define three zones: Zone (0) requires no action, Zone (1) mandates adjustments at Speed 1 for minor corrections, and Zone (2) necessitates rapid corrections at Speed 2.In our tracking algorithm, if the Cd's y-coordinate falls within the circular region, no action is necessary. However, if it falls between the circular region and the horizontal line, the algorithm commands the drone to adjust its altitude (up or down) by decreasing or increasing the thrust to reposition Cd within the circle with Speed 1 for subtle adjustments. If the Cd's y-coordinate falls between the horizontal line and the image edge, similar commands apply but using Speed 2 for larger corrections.Similarly, with the Cd's x-coordinate relative to circular region and the vertical lines, the algorithm commands the drone to execute roll movements (left or right) to reposition Cd within the desired zone. In this case, Speed 2 is employed as a yaw movement for quicker adjustments.

Finally, our decision regarding forward/backward actions is determined by comparing the height of the bounding box of the detected leader drone with a reference height. This reference height is calculated based on the desired distance we want the follower drone to maintain from the leader. We calculate this reference height using the perspective projection equation (2.4), which relates the distance we want to operate with Z, where Y is the height of the leader drone and y is the reference height. The equation is as follows in (3.1):

$$y = Yf/Z \tag{3.1}$$

the same logic will be applied in forward/backward actions as illustrated in Figure and will be translated to pitch movement

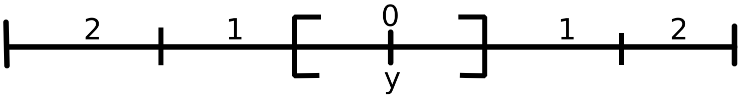


Figure 3.12: illustration of the decision zones relative to reference height y

The assignment of zones (0), (1), and (2) will depend on the desired tracking sensitivity. These zones are dynamically determined based on the precision required for the tracking process.The Figure3.13 provides a clear overview of the functioning of our tracking algorithm.

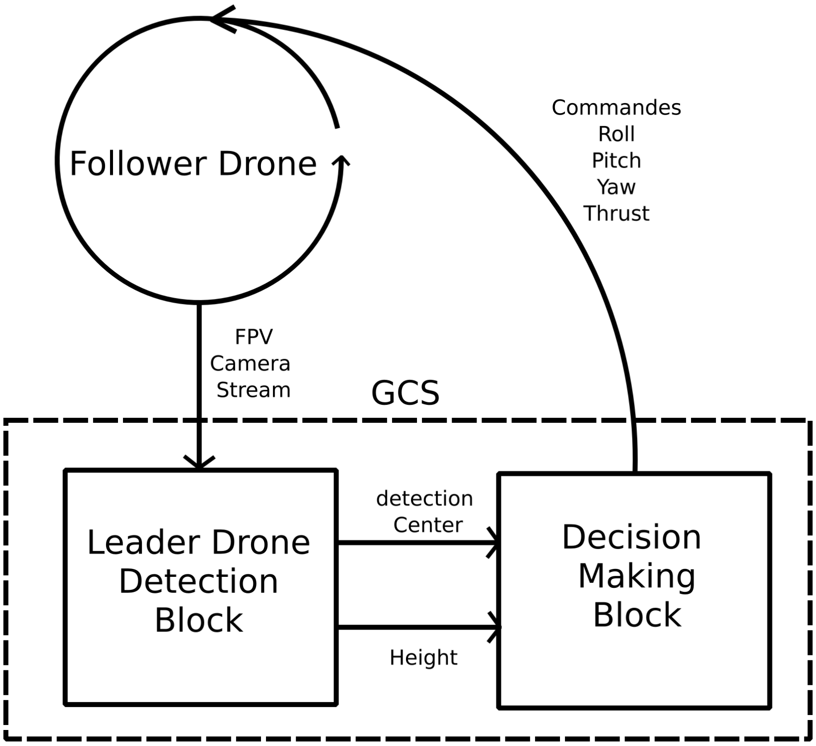


Figure 3.13: Diagram of the follower tracking algorithm

3.5 Conclusion

Throughout this chapter, we have delved into the immense potential of utilizing drone swarms, particularly in the context of wildfire management. Drone swarms offer enhanced flexibility, reliability, efficiency, and reduced mission time, all achieved through the collaborative efforts of the drones. These benefits can be harnessed in various mission formats and formations, depending on the specific needs of the task at hand.As a foundational and initial attempt to construct a swarm formation for wildfire detection, we proposed a straightforward approach powered solely by a computer vision-based and the previous positioning system. Unfortunately, due to constraints such as limited testing opportunities and time constraints, we were only able to explore the object detection aspect of this approach.This chapter highlights the exciting possibilities that lie ahead in the realm of drone swarm applications, particularly in addressing critical challenges like wildfire detection and management. Further research and development in this area hold the promise of more advanced and effective swarm formations that can significantly contribute to addressing real-world challenges.

Chapter 4

Fire detection and localization

4.1 Introduction

Fire detection is vital for safety, preventing devastating consequences in various settings. Fires in residential, industrial, or natural areas threaten lives, property, and the environment. Swift and accurate detection is crucial for timely responses that can minimize harm, save lives, and protect resources. Forest fires are a growing concern in countries like Algeria, the USA, and Canada in 2023 due to abundant combustible materials and favorable climates for ignition and rapid spread.

Recent research has explored the use of Unmanned Aerial Vehicles (UAVs) for fire monitoring and detection [34]. In the context of forest fire detection, UAVs serve several roles. Initially, they were used for navigating forested areas, capturing video footage, and then analyzing it to identify fires. With advancements in UAV technology, affordable commercial UAVs are now accessible for various research purposes. These UAVs can enter high-risk zones, provide a bird's-eye view of challenging terrain, and conduct nighttime missions without risking human lives. The integration of UAVs into these operations offers numerous advantages, as discussed in [35].

This chapter focuses on enhancing fire detection through drone technology and computer vision. It addresses challenges in complex environments, emphasizes the need for reliable detection, and explores innovative solutions. By combining cutting-edge tech and data-driven approaches, we aim to improve accuracy, speed, and adaptability in fire detection, contributing to more effective prevention and response in evolving fire landscapes.

4.2 Related work

4.2.1 UAVs for forest fire detection and monitoring

One common task in forest fire remote sensing is fire mapping, which generates maps of fire locations within a specific timeframe using georeferenced aerial images. These maps can also help determine ongoing fire perimeters and estimate positions in unobserved areas. Continuous mapping to provide regular updates is referred to as monitoring. Drones, equipped with high-resolution cameras, play a crucial role in generating accurate fire data, allowing for the characterization of fire geometry. Remote 3D reconstruction of forest fires provides valuable information for firefighters to assess fire severity safely at specific locations [36].

Drone applications in firefighting, primarily focused on forest fire remote sensing, are discussed in [37]. A proposed autonomous monitoring system in [38] tracks hot spots using a realistic simulation of fire progression. Drones are considered for tasks like aerial prescribed fire lighting, but implementation faces challenges [39]. [40] presents a cost-effective framework using mixed learning techniques for precise fire detection over burned areas. Additionally, [41] introduces a real-time forest fire monitoring system utilizing a UAV equipped with sensors and a camera for onboard data processing.

4.2.2 Stereo vision-based fire localization system

A stereo vision-based fire location system automatically detects fire flames in camera-captured images, then uses calibrated stereo vision to determine the fire's 3D coordinates relative to the camera's perspective. This precise information is vital for rapid firefighting responses and accurate water injection [42]. [43] applied stereo vision for 3D fire location in an industrial setting, while [44] combined stereo infrared cameras and laser radar for fire positioning in smoky environments, with an operational range of under 10 meters. [45] faced limitations for short-distance fire identification.

[46] developed a stereo vision system with a 100mm base distance,successfully localizing fires within 20 meters in outdoor experiments. However, challenges remain, as discussed by [47] and [48], including calibration accuracy affecting precision and adaptability to varying light positioning distances. Ongoing efforts explore solutions for improvement.

4.2.3 Vision based automatic forest fire detection techniques

Vision-based techniques are crucial in forest fire monitoring and detection, offering real-time data capture, wide detection range, and easy verification. In the past decade, vision-based UAV systems have played a significant role in this field.[49] introduces YOLO-v8 for precise fire detection in smart city settings.[50] provides an overview of object detection techniques, with a focus on YOLO's evolution.

Leveraging multiple information sources is crucial for complex forest fires, often involving color, motion, and geometry analysis. [51] introduces a novel framework combining these features with machine learning for fire detection. [52] presents a multi-UAV system for forest fire detection using color and motion analysis. [53] proposes a robust forest fire detection system, curating a diverse dataset and employing transfer learning. [54] offers a deep learning-based model for identifying fires in satellite images, achieving high accuracy.

4.3 Methodology

4.3.1 System description

we propose a fire detection and geo-localization system (see Figure 4.1) using UAV imagery. Here's how it works:

- Data Collection: UAVs with high-res cameras capture detailed images of the target area.
- Object Detection: We use YOLO models to identify fire, no fire, and smoke based on labeled images from data collection.
- Precise Fire Localization: Advanced stereo vision techniques involve camera calibration, depth estimation, and position calculation to pinpoint the fire's 3D coordinates as shown in chapter 2 while utilizing the drones ordination with their GPS estimation in the navigation translation to acquire the fire's global coordinates.

This integrated system provides accurate fire detection and localization, enabling swift responses in urban and wildland settings, enhancing safety for lives, property, and the environment.

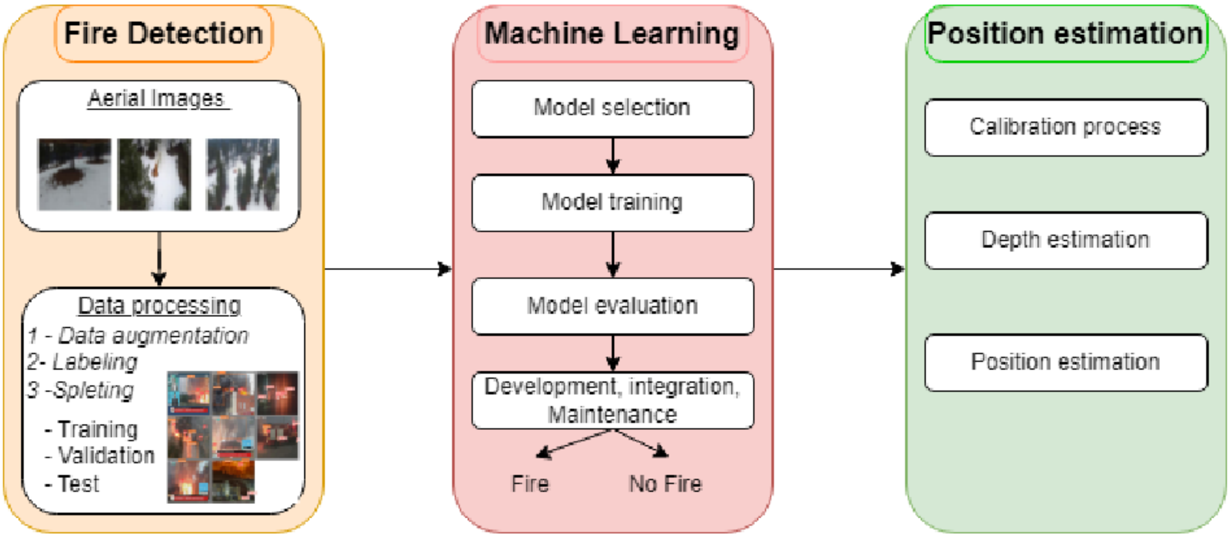


Figure 4.1: Fire detection and geo-localization scheme

4.3.2 Data collection

The effectiveness of CNN-based forest fire detection heavily relies on the quality and quantity of the datasets used. A high-quality dataset enables deep learning models to capture a wider range of characteristics and improve their generalization capabilities. To achieve this, we compiled a dataset exceeding 12,000 images from various publicly available fire datasets, including BowFire, FiSmo, Flame, and custom images from our laboratory.

This dataset consists of aerial images of fires, captured by UAVs in diverse scenarios and with different equipment configurations. A selection of representative samples from this dataset is shown in Figure 4.3.Before use, we conducted preprocessing to ensure the relevance and availability of aerial images for the study area. As a result, we excluded images without fires, those where fires were not discernible, and considered hardware limitations. Ultimately, we curated a dataset containing

12,530 images, each meticulously labeled into one of three primary categories: Fire, No-fire, or Smoke.

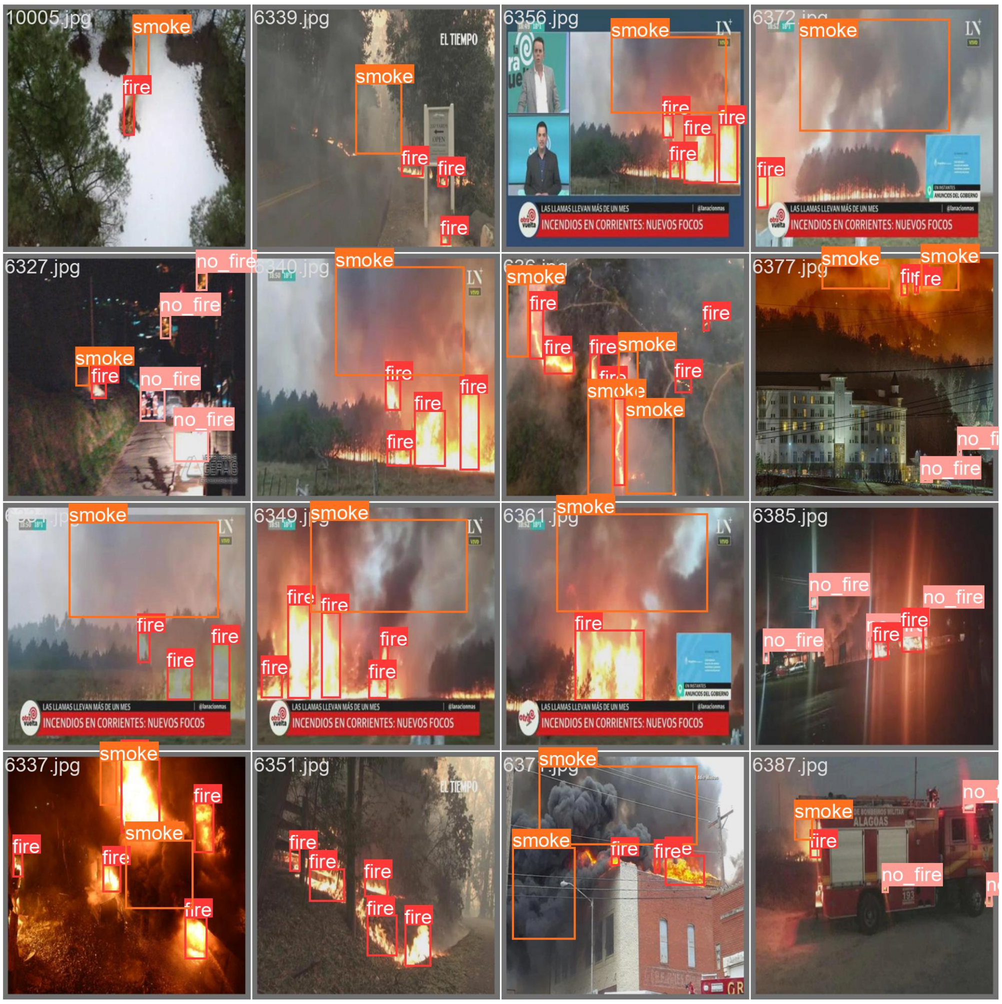


Figure 4.2: Image labeling session

4.3.3 Yolo architecture

Joseph Redmon et al [55]. introduced YOLO (You Only Look Once) in their publication at CVPR 2016, marking a significant milestone in the field of object detection. YOLO revolutionized the approach to real-time object detection by accomplishing the task with a single pass of the network. This was a departure from earlier methods that relied on sliding windows followed by a classifier, which had to be executed hundreds or even thousands of times per image. Additionally, YOLO distinguished itself from more advanced methods that divided the detection process into two steps: first identifying possible regions or proposals containing objects and then running a classifier on these proposals. YOLO's innovative approach streamlined the process and delivered more straightforward output, contributing to its popularity and impact in the computer vision community.

4.3.3.1 Yolov4

YOLOv4 leverages an array of innovative features, including Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross mini-Batch Normalization (CmBN), Self-adversarial-training (SAT), Mish-activation, Mosaic data augmentation, DropBlock regularization, and CIoU loss, all of which collaborate synergistically to optimize its performance, enabling it to achieve state-of-the-art results in object detection. The typical structure of an object detector encompasses the input, the backbone, the neck, and the head. YOLOv4's backbone, pretrained on ImageNet, plays a central role in class prediction and object bounding box estimation, with options to use backbones like VGG, ResNet, ResNeXt, or DenseNet. The neck component collects feature maps from various stages, facilitating multiscale feature extraction. Ultimately, the head processes these features to produce the final object detections and classifications, culminating in YOLOv4's outstanding object detection capabilities within the computer vision domain.

4.3.3.2 Yolov5

A month after YOLOv4's release, Glenn Jocher and his team introduced YOLOv5 [56], a new iteration. YOLOv4 was originally created by Alexey Bochkovsky using the Darknet framework, while YOLOv5 transformed previous versions into Python-based PyTorch, a simpler choice for IoT device integration. YOLOv5's naming generated debate due to its perceived lack of significant advancements over YOLOv4. Glenn didn't publish papers about YOLOv5, raising questions. While YOLOv5 and YOLOv4 have similar architectures, comparing their performance is challenging due to language and framework differences. Over time, YOLOv5 demonstrated superior performance in specific conditions, gaining traction in the computer vision community.

4.3.3.3 Yolov8

In January 2023, Ultralytics [57], the developer behind YOLOv5, introduced YOLOv8, an extension that broadened its capabilities to encompass various vision tasks such as object detection, segmentation, pose estimation, tracking, and classification. YOLOv8 retains the core structure of YOLOv5 but refines the CSPLayer, now referred to as the C2f module, which combines high-level features with contextual information for enhanced detection accuracy. It adopts an anchor-free model with a disentangled head, enabling independent processing of objectness, classification, and regression tasks, ultimately improving overall model accuracy. The output layer employs the sigmoid function for objectness scores and softmax for class probabilities, while leveraging CIoU and DFL loss functions for bounding box loss and binary cross-entropy for classification loss, particularly benefiting smaller object detection. YOLOv8 also introduces YOLOv8-Seg, a semantic segmentation counterpart with a CSPDarknet53 feature extractor and C2f module, alongside two segmentation heads for predicting segmentation masks, following a similar detection structure as YOLOv8.

4.3.3.4 Yolov-NAS

In May 2023, Deci introduced YOLO-NAS [58], a specialized deep learning model aimed at improving small object detection, localization accuracy, and performance-computation ratio, making it suitable for edge devices. This open-source framework utilizes AutoNAC, an adaptable system that tailors models to specific tasks, data, and performance goals. During the Neural Architecture Search (NAS) process, RepVGG blocks were integrated for compatibility with Post-Training Quantization (PTQ), resulting in three YOLO-NAS models: YOLO-NASS, YOLO-NASM, and YOLO-NASL (small, medium, and large).

4.3.4 Model training

The forest fire detection system was trained using various YOLO models on a robust platform featuring an AMD Ryzen 5600H processor with 6 cores and 12 threads, along with an Nvidia RTX 3060 GPU with 6GB of RAM.Training optimization utilized the stochastic gradient method (SGD), with key hyperparameters set as follows: initial learning rate (0.003), batch size (16), and number of epochs (200). Specific training strategies were employed to expedite convergence and enhance model accuracy, supported by empirical results.

Figure 6 visually presents the detection outcomes across different test dataset scenarios. In most cases, the system effectively identifies and classifies fire, no-fire, and smoke regions with high confidence. However, it's important to note that some small fire objects may remain undetected due to limitations in discernibility caused by factors like distance and image resolution.

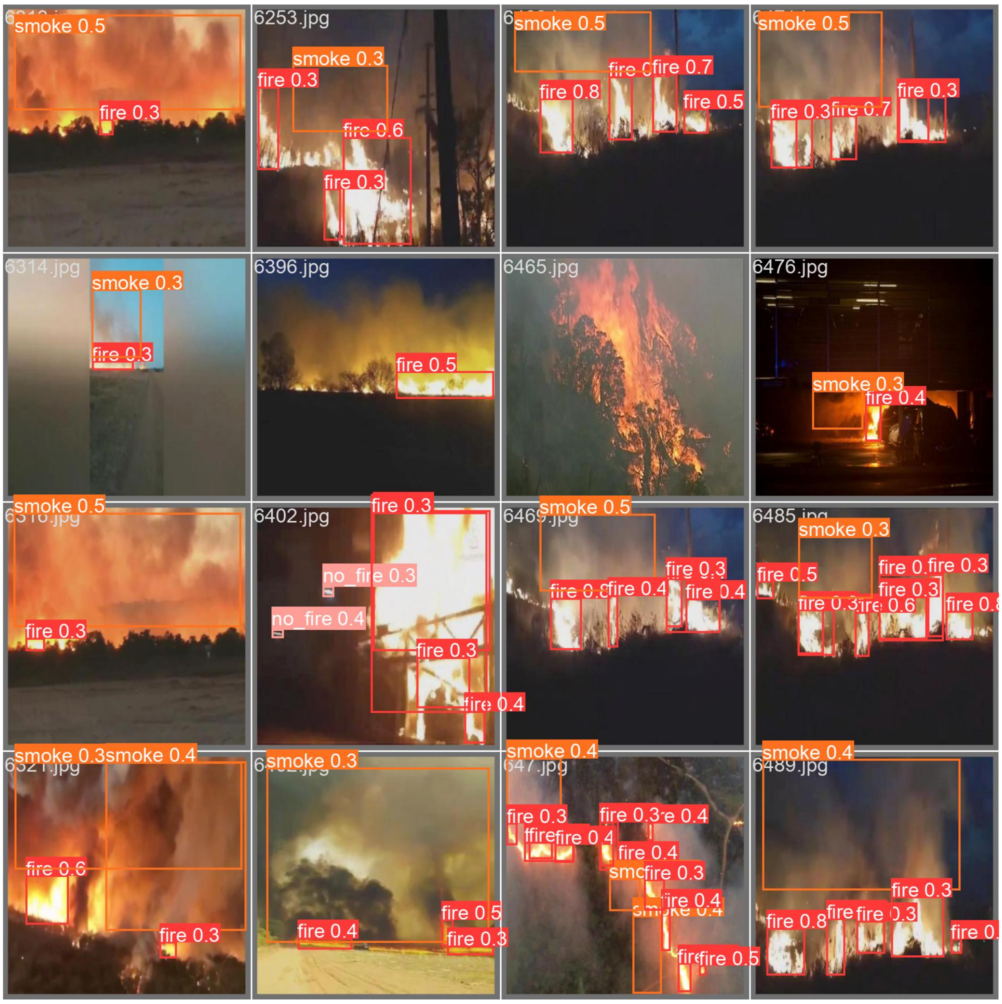


Figure 4.3: Example of confidence score diversity

4.3.5 Comprehensive comparison of yolo models

Table 4.1 compares the performance metrics of different fire detection models: Yolov4, Yolov5, Yolov8, and YoloNas. The metrics include Precision (P), Recall (R), and mean Average Precision at IoU 0.5 (mAP50) for three classes: Fire, No Fire, and Smoke, showcasing variations in their abilities.

- For the Fire class:
  - Yolov4: Precision 0.58, Recall 0.61
  - Yolov5: Precision 0.62, Recall 0.66
  - Yolov8: Precision 0.64, Recall 0.67
  - YoloNas: Precision 0.67, Recall 0.71
- For the No Fire class:
  - Yolov8: Precision 0.76, Recall 0.76
  - Others have slightly lower values.
- For the Smoke class:
  - Yolov8: Precision 0.60, Recall 0.43
  - Yolov4: Precision 0.50, Recall 0.50
  - YoloNas: Balanced mAP50 of 0.53
- For the overall performance (mAP50):
  - YoloNas leads with 0.87, indicating superior overall performance.
  - Yolov4, Yolov5, and Yolov8 also have competitive scores (0.66, 0.68, and 0.70, respectively).

These results reveal trade-offs between Precision and Recall, with YoloNas demonstrating balanced performance across classes, making it a notable choice for comprehensive fire detection.

Table 4.1: Yolo Models evaluation results

Model	Class	P	R	mAP50
Yolov4	Fire	0.58	0.61	0.66
	No Fire	0.72	0.71	0.72
	Smoke	0.54	0.48	0.51
Yolov5	Fire	0.62	0.66	0.68
	No Fire	0.73	0.72	0.75
	Smoke	0.55	0.47	0.50
Yolov8	Fire	0.64	0.67	0.70
	No Fire	0.76	0.76	0.83
	Smoke	0.60	0.50	0.54
YoloNas	Fire	0.67	0.71	0.73
	No Fire	0.81	0.78	0.87
	Smoke	0.63	0.48	0.53

4.4 Position estimation

This part plays a pivotal role in achieving the system's objectives, especially concerning the swift and effective response to wildfire propagation. Precisely localizing the fire is paramount, and this localization approach draws from IPS discussed in Chapter 2, albeit with significant enhancements. Notably, the stereoscopic camera, previously static, is now dynamic,based on the drone's heading and orientation and the integration of GPS postion estimation. It enables the precise determination of the fire's location relative to the drone's frame, with subsequent translation into the global frame. This process provides accurate geo coordinates for the fire's location, facilitating rapid and effective responses to wildfire spread based on precise information.

4.5 Conclusion

In conclusion, the integration of advanced technology, including Unmanned Aerial Vehicles, YOLO-based object detection models, and dynamic stereoscopic cameras, has significantly enhanced the capabilities of forest fire detection and localization systems. These systems rely on high-quality datasets, powerful hardware, and meticulous training processes to achieve impressive results. While there are trade-offs between precision and recall, YoloNas emerges as a notable choice for comprehensive fire detection with its balanced performance across fire-related classes. Precise fire localization, using GPS position estimation and dynamic camera adjustments, is pivotal for the rapid and effective response to wildfires. This holistic approach holds the promise of not only safeguarding lives, property, and the environment but also mitigating the devastating consequences of wildfires in both urban and wildland settings. The continuous advancement of technology and the refinement of these systems are poised to further improve their capabilities and bolster fire safety and prevention efforts.

General Conclusion

In the course of this work, we embarked on a journey into the dynamic and transformative intersection of computer vision, machine learning, and unmanned aerial vehicles. Our exploration began by delving into the realms of computer vision and machine learning, demonstrating their pivotal role in enhancing the capabilities of drones for a multitude of applications. With these fundamental concepts as our foundation, we ventured into the development of an indoor positioning system, a crucial innovation intended to overcome the limitations imposed by the absence of GPS signal within indoor environments.

The creation of this indoor positioning system marked a significant milestone in our research journey, providing a solution to the challenge of precise drone localization in environments where traditional GPS-based positioning falls short. The successful implementation of the IPS not only paved the way for our subsequent work but also offered promising results and a great potential that underscored the feasibility of deploying drones for various indoor applications.

As we progressed, our focus shifted towards the realization of swarm-based drone technologies. Our research aimed to push the boundaries further, capitalizing on the potential of drone swarms to tackle complex tasks. Within this context, we proposed a two-drones swarm initial approach, built upon a leader-follower centralized formation. This approach represents a significant stride towards the development of intelligent and coordinated swarm of drones capable of addressing a wide range of applications,and help to evolve Wilde fire detection UAV based systems.

The journey we undertook throughout this work has illuminated the remarkable possibilities that lie at the confluence of computer vision, machine learning, and UAV technology. Our work contributes not only to the advancement of drone capabilities but also to the broader landscape of technology-driven solutions in indoor and outdoor environments. The successful implementation of the IPS and the initiation of the two-drones swarm approach signify not just individual achievements but also a collective step forward in harnessing the potential of these cutting-edge technologies.

In closing, this work underscores the significance of continued research and innovation in the realm of drones and their applications. As we conclude this chapter, we recognize that our findings open doors to new horizons and invite future explorations into the limitless possibilities of this dynamic field. Our work stands as a testament to the power of interdisciplinary research and collaboration in shaping the future of technology, making it an exciting time to be part of this ever-evolving journey.

Further work

The work conducted in this ongoing thesis research suggests potential areas for future research, including the following propositions :

- Enhance the IPS by incorporating additional cameras at strategic locations to mitigate precision degradation. Implement a data fusion approach that accumulate data from multiple positioning sources, giving preference to the source with the highest probability of precision in determining the position.
- Explore the foundational principles of the positioning system and consider future integration with stereo vision technology on the drones. This integration can enable the creation of a depth map, providing valuable insights into the surrounding environment, including the proximity of objects. Essentially, this depth map can function as a proximity sensor or vision-based radar system, aiding in obstacle avoidance and enhancing overall operational safety.
- Dive deeper into the realm of swarm formation by expanding the swarm with the inclusion of more drones. Transition towards a decentralized and autonomous swarm framework,with a monitoring feature. Implement advanced machine learning algorithms, specifically leveraging Reinforcement Learning techniques, to the coordination and behavior of the swarm.

=0mu plus 1mu

[hyphens]url hyperref

Bibliography

[1] S. Thapa, V. S. Chitale, S. Pradhan, B. Shakya, S. Sharma, S. Regmi, S. Bajracharya, S. Adhikari, and G. S. Dangol, *Forest Fire Detection and Monitoring*, pp. 147–167. Cham: Springer International Publishing, 2021.

[2] P. G. Fahlstrom, T. J. Gleason, and M. H. Sadraey, *Introduction to UAV systems*. John Wiley & Sons, 2022.

[3] C. Yuan, Y. Zhang, and Z. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," *Canadian journal of forest research*, vol. 45, no. 7, pp. 783–792, 2015.

[4] M. Sadi, "Uav-based forest fire detection and localization using visual and thermal cameras," December 2020. Available at https://spectrum.library.concordia.ca/id/eprint/987884/1/Sadi$_{M}ASc_{S}2021.pdf$.

[5] S. Bhattacharyya, V. Snasel, A. Hassanien, S. Saha, and B. Tripathy, *Deep Learning: Research and Applications*. De Gruyter Frontiers in Computational Intelligence, De Gruyter, 2020.

[6] T. W. Edgar and D. O. Manz, "Chapter 6 - machine learning," in *Research Methods for Cyber Security* (T. W. Edgar and D. O. Manz, eds.), pp. 153–173, Syngress, 2017.

[7] A. B. M. R. Islam, "Machine Learning in Computer Vision:," in *Advances in Educational Technologies and Instructional Design* (S. Khadimally, ed.), pp. 48–72, IGI Global, Feb. 2022.

[8] J. Ren and Y. Wang, "Overview of object detection algorithms using convolutional neural networks," *Journal of Computer and Communications*, vol. 10, no. 1, pp. 115–132, 2022.

[9] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, “A survey of modern deep learning based object detection models,” *Digital Signal Processing*, vol. 126, p. 103514, 2022.

[10] S. J. Prince, *Computer vision: models, learning, and inference.* Cambridge University Press, 2012.

[11] D. Forsyth and J. Ponce, *Computer vision: a modern approach.* Boston: Pearson, 2nd ed ed., 2012.

[12] A. Fahmy, “Stereo vision based depth estimation algorithm in uncalibrated rectification,” 2013.

[13] S. Živanović, D. Zigar, and J. Čipev, “Forest roads as the key to forest protection against fire,” *Safety Engineering*, vol. 11, no. 2, pp. 59–64, 2021.

[14] C. Yuan, Z. Liu, and Y. Zhang, “Vision-based forest fire detection in aerial images for firefighting using uavs,” in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1200–1205, 2016.

[15] C. Yuan, *Automatic Fire Detection Using Computer Vision Techniques for UAV-based Forest Fire Surveillance.* Phd thesis, University of Concordia, Montréal, Québec, Canada, May 2017. Available at https://core.ac.uk/download/pdf/211519923.pdf.

[16] A. I. Khan and S. Al-Habsi, “Machine learning in computer vision,” *Procedia Computer Science*, vol. 167, pp. 1444–1451, 2020. International Conference on Computational Intelligence and Data Science.

[17] R. Shanmugamani and S. Moore, *Deep Learning for Computer Vision: Expert Techniques to Train Advanced Neural Networks Using TensorFlow and Keras.* Packt Publishing, 2018.

[18] R. Collins, “Camera projection,” *CSE486, Penn State.*

[19] R. Szeliski, *Computer vision: algorithms and applications.* Texts in computer science, London ; New York: Springer, 2011. OCLC: ocn462920910.

[20] X. Liu, J. Tian, H. Kuang, and X. Ma, “A Stereo Calibration Method of Multi-Camera Based on Circular Calibration Board,” *Electronics*, vol. 11, p. 627, Feb. 2022.

[21] S. Shah and J. Aggarwal, “Depth estimation using stereo fish-eye lenses,” in *Proceedings of 1st International Conference on Image Processing*, vol. 2, (Austin, TX, USA), pp. 740–744, IEEE Comput. Soc. Press, 1994.

[22] N. Sombekke, “Triangulation for depth estimation,” *university of amsterdam*, 2020.

[23] W. A. O. I. A. A. H. Abdelmoghit Zaarane *, Ibtissam Slimani, “Distance measurement system for autonomous vehicles using stereo camera,” *LTI Lab, Department of Physics, Faculty of Sciences Ben M’Sik, University Hassan II Of Casablanca, Morocco*, 2020.

[24] C. B. M. . C. S. Andersen, “Optimal landmark selection for triangulation of robot position,” *Laboratory of Image Analysis, Aalborg University*, 1998.

[25] E. Teague and R. K. Jr, *Swarming Unmanned Aircraft Systems.* no, 2008.

[26] A. Hussein, S. Elsawah, E. Petraki, and H. A. Abbass, “A machine education approach to swarm decision-making in best-of-n problems,” *Swarm Intell.*, Mar. 2022.

[27] P. Skobelev, D. Budaev, N. Gusev, and G. Voschuk, “Designing multi-agent swarm of UAV for precise agriculture,” in *Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*, Communications in computer and information science, pp. 47–59, Cham: Springer International Publishing, 2018.

[28] C. H. H. H. Mingyang Lyu, Yibo Zhao, “Unmanned aerial vehicles for search and rescue: A survey,” *Remote Sensing*, 2023.

[29] R. K. Dharna Nar, “Optimal waypoint assignment for designing drone light show formations,” *Results in Control and Optimization*, 2022.

[30] B. Alkouz, A. Bouguettaya, and S. Mistry, “Swarm-based drone-as-a-service (SDaaS) for delivery,” in *2020 IEEE International Conference on Web Services (ICWS)*, IEEE, Oct. 2020.

[31] H. K. T. L. abrice Saffre, Hanno Hildmann, “Monitoring and cordoning wildfires with an autonomous swarm of unmanned aerial vehicles,” *MDPI*, 2022.

[32] E. Zaitseva, V. Levashenko, R. Mukhamediev, N. Brinzei, A. Kovalenko, and A. Symagulov, “Review of reliability assessment methods of drone swarm (fleet)

and a new importance evaluation based method of drone swarm structure analysis," *Mathematics*, vol. 11, p. 2551, June 2023.

[33] X. Chen, J. Tang, and S. Lao, "Review of unmanned aerial vehicle swarm communication architectures and routing protocols," *Applied Sciences*, vol. 10, p. 3661, May 2020.

[34] F. H. J. Partheepan, S.; Sanati, "Autonomous unmanned aerial vehicles in bushfire management: Challenges and opportunities," *Drones*, 2023.

[35] M. T. M. A. S. N. Q. Ahmed, H.; Bakr, "nmanned aerial vehicles (uavs) and artificial intelligence (ai) in fire related disaster recovery: analytical survey study.," *IEEE*, 2022.

[36] H. T. A. K. A. Bouguettaya, A.; Zarzour, "review on early wildfire detection from unmanned aerial vehicles using deep learning-based computer vision algorithms.," *Signal Processing*, 2022.

[37] C. D. C. H. J. L. C. E. S. S. Twidwell, D.; Allen, "unmanned aerial systems for fire management.," *Frontiers in Ecology and the Environment*, p. 333–339., 2016.

[38] G. Skeele, R.C.; Hollinger, "Aerial vehicle path planning for monitoring wildfire frontiers. in proceedings of the field and service robotics.," *Springer*, 2016.

[39] C. E. S. D. B. H. C. T. D. C. eachly, E.; Detweiler, "Fire-aware planning of aerial trajectories and ignitions.," *IEEE*, 2018.

[40] D. A. K. R. C. P. T. N. S. R. asyap, V.L.; Sumathi, "Early detection of forest fire using mixed learning techniques and uav.," *Computational intelligence and neuroscience*, 2022.

[41] R. K. P. K. N. D. B. eslya, U.J.; Chaitanyab, "A detailed investigation on forest monitoring system for 539 wildfire using iot. i," *IOS Press,*, 2023.

[42] A. Kustu, T.; Taskin, "Deep learning and stereo vision based detection of post-earthquake fire geolocation for smart cities within the scope of disaster management: ˙istanbul case.," *International Journal of Disaster Risk Reduction*, 2023.

[43] B. Z. M. D. L. Song, T.; Tang, "An accurate 3-d fire location method based on sub-pixel edge detection and non-parametric stereo matching.," *Measurement*, 2014.

[44] C. Y. S. U. Tsai, P.F.; Liao, "Using deep learning with thermal imaging for human detection in heavy smoke scenarios.," *Sensors*, 2022.

[45] W. L. D. Z. G. Zhu, J.; Li, "study on water jet trajectory model of fire monitor based on simulation and experiment.," *Fire Technology*, 2019.

[46] L. A. M. P. A. M. X. A. Toulouse, T.; Rossi, "A multimodal 3d framework for fire characteristics estimation.," *Measurement Science and Technology*, 2018.

[47] B. R. McNeil, J.G.; Lattimer, "Robotic fire suppression through autonomous feedback control.," *Fire technology*, 2017.

[48] F. X. T. Wu, B.; Zhang, "Monocular-vision-based method for online measurement of pose parameters of weld stud.," *Measurement*, 2015.

[49] H. A. Talaat, F.M.; ZainEldin, "An improved fire detection approach based on yolo-v8 for smart cities.," *Neural Computing and Applications*, 2023.

[50] G. T. J. Diwan, T.; Anirudh, "Object detection using yolo: Challenges, architectural successors, datasets and applications.," *multimedia Tools and Applications*, 2023.

[51] A. A. F. K. G. J. K. R. Harjoko, A.; Dharmawan, "Real-time forest fire detection framework based on artificial intelligence using color probability model and motion feature analysis.," *Fire*, 2022.

[52] V. K. C. P. V. R. L. S. V. Sudhakar, S.; Vijayakumar, "Unmanned aerial vehicle (uav) based 569 forest fire detection and monitoring for reducing false alarms in forest-fires.," *Computer Communications*, 2020.

[53] B. K. S. A. R. A. A. Khan, A.; Hassan, "Deepfire: A novel dataset and deep transfer learning benchmark for forest fire detection.," *Mobile Information Systems*, 2022.

[54] A. B. S. Chopde, A.; Magon, "Forest fire detection and prediction from image processing using rcnn.,"

[55] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015.

[56] G. Jocher, A. Stoken, J. Borovec, NanoCode012, Ayush Chaurasia, TaoXie, L. Changyu, Abhiram V, Laughing, Tkianai, YxNONG, A. Hogan, Lorenzomammana, AlexWang1900, J. Hajek, L. Diaconu, , Marc, Yonghye Kwon, ,

Oleg, Wanghaoyang0106, Y. Defretin, A. Lohia, Ml5ah, B. Milanko, B. Fineran, D. Khromov, D. Yiwei, , Doug, Durgesh, and F. Ingham, “ultralytics/yolov5: v5.0 - yolov5-p6 1280 models, aws, supervise.ly and youtube integrations,” 2021.

[57] D. Reis, J. Kupec, J. Hong, and A. Daoudi, “Real-time flying object detection with yolov8,” 2023.

[58] Ultralytics, “Yolo-nas neural architecture search.” Ultralytics YOLOv8 Docs,https://docs.ultralytics.com/models/yolo-nas/citations-and-acknowledgements.

Business Model Canvas

Business Model Canvas - *BMC*

Porteurs de projet :	*Promoteurs :*	*Code de projet :*
1-Chegrani Akram 2- Yahiaoui Mohamed	P- Dr. Choutri Kheireddine CO-P- Prof. Lagha Mohand	

Project Startup : Swarm of drones for forest fire detection

Key Partners :	Key Activities:	Value Propositions	Customer Relationships:	Customers:
• Suppliers of equipment • Research centers.	• drone production and development of software • Maintenance.	• forest monitoring prone to fire • detect fire at its beginnings • help journalists to report the events by offering a unique perspective of the event and landscapes	• personal help • social networks	• firefighters • journalist • general directorate of forests • farmers
	Key Resources:		Channels:	
	• engineers • Equipment		• Direct sales. • Online with Delivery	

Costs:	Revenues:
• Manufacturing, assembly and operating costs to produce the products. • Product marketing and promotion costs • Salaries, Workplaces cost	• Drones sale • Provide maintenance services • Software’s sale

Annex

The startup project technical sheet 1275

البطاقة التقنية للمشروع

The project technical sheet

Information card	
شقراني أكرم / يحياوي محمد Chegrani Akram\ Yahiaoui Mohamed	الاسم و اللقب **Votre prénom et nom**
سرب من الطائرات بدون طيار للكشف عن حرائق الغابات Swarm of drones for forest fire detection	الاسم التجاري للمشروع **Title of your Project**
\	الصفة القانونية للمشروع **Your legal status**
0672016275 0669538224	رقم الهاتف **Your phone number**
akramchegrani11@gmail.com **yh.med.2000@gmail.com**	البريد الالكتروني **Your email address**
Blida- Soumaâ	مقر مزاولة النشاط (الولاية- البلدية) **Your city or municipality of activity**

طبيعة المشروع (طبيعة الابتكار)

The nature of the project

المنتوج ذو طابع إنتاجي أو خدماتي

Sale of goods or services

- **Key Partners :**

In the Business Model Canvas, key partners refer to the individuals, companies, or organizations with whom we collaborate to create, distribute, or deliver value to our customers, our example:

Suppliers of equipment are essential to UAV production, providing critical components like propulsion systems, sensors, and avionics. Their role is indispensable, directly influencing UAV performance and capabilities. This partnership between UAV manufacturers and equipment suppliers drives technological advancements and supports a wide range of applications, from surveillance to agriculture and beyond.

Research Centers: This partnership enables us to remain at the cutting edge of technological advancements and harness the latest research findings to enhance our project.

- **Key Activities:**

The key activities in the Business Model Canvas refer to the essential tasks and actions undertaken by our company to create, deliver, and sustain our value proposition. In our example:

In the dynamic landscape of our drone startup, two key activities take center stage. Firstly, the core of our operations lies in the production of cutting-edge drones and the continuous development of innovative software solutions. This dual approach ensures that our UAVs remain at the forefront of technology, equipped with the latest capabilities and features. Equally vital is our commitment to maintenance services, where we provide essential support to our customers, ensuring the longevity and optimal performance of their drones. These key activities define our mission to deliver excellence in both hardware and software while maintaining a strong and enduring connection with our valued clientele.

- **Key Resources:**

These resources form the backbone of a startup, providing the essential tools, assets, and knowledge required to create, deliver, and maintain a unique value proposition, our example:
Engineers and equipment’s: in the realm of drone manufacturing and software development, the role of engineers and cutting-edge equipment is indispensable. Engineers bring their expertise to the table, lending their innovative problem-solving skills to design and build drones that are not only technologically advanced but also safe and efficient. Their knowledge of aerodynamics, electronics, and software programming is paramount in creating drones that can fulfill a diverse range of functions, from aerial photography to industrial inspections.

Moreover, the importance of high-quality equipment cannot be overstated. State-of-the-art machinery and tools are instrumental in the precise manufacturing of drone components, ensuring reliability and performance. From the intricate circuitry of flight controllers to the precision engineering of propulsion systems, the use of advanced equipment significantly contributes to the quality and functionality of the final product.

- **Value Propositions** :

Value Propositions in the Business Model Canvas refer to the unique benefits, advantages, and solutions that we, as a company, offer to meet our customers' needs and differentiate ourselves from our competitors. In our example, here are several value propositions:

In the domain of forest monitoring, particularly in areas susceptible to wildfires, our system excels on multiple fronts. Firstly, it is finely tuned to detect the inception of fires, ensuring rapid response and mitigation efforts. This early detection capability is vital in safeguarding the environment and reducing the potential for widespread destruction.

Furthermore, our system extends its reach beyond fire prevention. It aids journalists in their reporting endeavors by providing a distinctive perspective of events and landscapes. This unique vantage point not only enhances the accuracy and comprehensiveness of news coverage but also contributes to a deeper understanding of the situation on the ground. Together, these elements underscore the versatility and significance of our solution in forest monitoring and disaster reporting.

- **Customer Relationships**:

In the Business Model Canvas, "Customer Relationships" represents the various ways a company interacts with and serves its customers.

Establishing effective customer relationships is pivotal, as it directly impacts customer satisfaction and loyalty, ultimately driving the success of the business, in our example:

Personal assistance and social networks are invaluable tools. Personal help signifies our commitment to providing tailored support, ensuring that our customers receive individualized guidance and solutions. On the other hand, social networks open up a world of connectivity, enabling customers to engage with our brand and each other, fostering a sense of community and collaboration. These dual approaches enhance customer satisfaction, ensuring that our clientele receives the assistance they need while also fostering a vibrant network of shared experiences and insights.

- **Channels:**

In the Business Model Canvas, "Channels" represent the means through which products or services are distributed and the methods of communication used to engage with customers. By strategically managing channels, businesses can enhance their reach, accessibility, and overall success in delivering their value proposition, in our example:

In the realm of channels within the Business Model Canvas, "Direct Sales" and "Online with Delivery" serve as two fundamental approaches. Direct sales entail a hands-on, personal interaction with customers, often through a dedicated sales force or physical retail locations. On the other hand, "Online with Delivery" leverages the convenience of e-commerce platforms to reach a wider audience, with products and services delivered directly to customers' doorsteps. These channels cater to distinct customer preferences, offering flexibility and accessibility while ensuring that our value proposition seamlessly reaches our diverse clientele.

- **Customers:**

In the Business Model Canvas (BMC), "Customers" represent the heart of any business endeavor. They encompass the diverse individuals, groups, or organizations that a company aims to serve and create value for. Understanding customer segments, their needs, preferences, and behaviors, is essential for tailoring the company's value proposition, channels, and overall strategy, in our example:

Our customer base is remarkably diverse, encompassing a wide range of individuals and organizations with distinct needs for drones. Primarily, our drones find value among essential groups such as firefighters, who rely on our technology to combat wildfires more effectively. Journalists also benefit from our drones, using them to capture unique perspectives and cover events comprehensively. Furthermore, government agencies, like the General Directorate of Forests, leverage our drones for critical tasks such as forest monitoring and management. Additionally, our drones serve the agricultural sector, aiding farmers in optimizing crop yields and resource management. Our commitment to addressing the specific needs of these diverse customer segments underscores our dedication to delivering drone solutions that make a meaningful impact across various industries and applications.

- **Costs :**

In the Business Model Canvas, "Costs" encompass the financial investments that a business undertakes to deliver its value proposition and operate effectively. Cost management is pivotal, as it directly impacts the company's profitability, pricing strategies, and resource allocation, in our example:

In the realm of startups, prudent cost management is pivotal for success. This trio of costs includes the essential manufacturing, assembly, and operating expenses required to bring products to life. Concurrently, investing in effective product marketing and promotion is crucial to garner visibility and customer interest. Moreover, salaries and workplace costs play a pivotal role in assembling and retaining a skilled workforce. Striking the right balance in managing these costs is vital for startups as they navigate their early stages, ensuring they can efficiently deliver their value proposition, establish a market presence, and build a capable team while maintaining financial sustainability.

- **Revenues:**

In the Business Model Canvas, "Revenues" represent the lifeblood of a company, encompassing the various income streams that sustain its operations. Effectively managing and diversifying revenue streams is essential for ensuring financial stability and the long-term success of the business.in our example:

In our startup, income streams diversify through multiple avenues. Firstly, there is revenue from drone sales, where our cutting-edge UAVs find homes with various clientele. Additionally, providing maintenance services ensures ongoing customer satisfaction and a steady income stream, as clients rely on our expertise to keep
their drones operational. Furthermore, software sales, such as our fire detection software as a service, add another dimension to our income portfolio, offering customers invaluable solutions while bolstering our financial stability. These combined income streams underscore our commitment to innovation and excellence in the drone industry, facilitating sustainable growth and value delivery to our diverse customer base.

- **Our product :**

Equipped with state-of-the-art sensors and sophisticated algorithms, our drone can swiftly identify objects in real-time, delivering unparalleled efficiency and precision. This cutting-edge drone seamlessly combines the agility and stability of a quadrotor with the precision and intelligence of the Pixhawk autopilot system, resulting in a powerful platform for object detection tasks. Allow us to introduce our groundbreaking product, a quadrotor drone equipped with the renowned Pixhawk autopilot system, specially engineered for advanced object detection capabilities.