

**MATRIX BARCODE BASED DETECTION AND
TRACKING FOR AUTONOMOUS UAV LANDING
A PROJECT SUMMARY**

submitted by

VELAYUDHAN A	- AM.EN.U4EEE17101
NANDAKISHORE R NAIR	- AM.EN.U4EEE17123
NEERAJ S	- AM.EN.U4EEE17124
VISHNU S	- AM.EN.U4EEE17147

*in partial fulfilment for the award of the degree
of*
**BACHELOR OF TECHNOLOGY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING**



AMRITA SCHOOL OF ENGINEERING, AMRITAPURI

AMRITA VISHWA VIDYAPEETHAM

AMRITAPURI 690 525

APRIL 2021

AMRITA VISHWA VIDYAPEETHAM
AMRITA SCHOOL OF ENGINEERING, AMRITAPURI, 690525



BONAFIDE CERTIFICATE

This is to certify that the seminar report entitled "**MATRIX BARCODE BASED DETECTION AND TRACKING FOR AUTONOMOUS UAV LANDING**" submitted by

“VELAYUDHAN A - AM.EN.U4EEE17101
NANDAKISHORE R NAIR - AM.EN.U4EEE17123
NEERAJ S - AM.EN.U4EEE17124
VISHNU S - AM.EN.U4EEE17147

in partial fulfilment of the requirements for the award of the Degree, **Bachelor of Technology** in "**ELECTRICAL AND ELECTRONICS ENGINEERING**" is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Engineering, Amritapuri.

Vivek A
Assistant professor
Dept. Electrical and Electronics Engineering

Manjula G Nair
Professor & Chairperson
Dept. of Electrical and Electronics Engineering

ACKNOWLEDGEMENT

Presentation inspiration and motivation have always played a key role in the success of any venture.

We pay our deep sense of gratitude to Dr Manjula G Nair (HOD) Electrical and electronics department Amrita School of Engineering Amritapuri to encourage us to the highest peak and to provide us with the opportunity to prepare the project. We are immensely obliged to our friends for their elevating inspiration, encouraging guidance and kind supervision in the completion of our report.

We feel to acknowledge our indebtedness and a deep sense of gratitude to our guide Mr Vivek Asst Prof Department of Electrical and Electronics Engineering, valuable guidance and kind supervision is given to us throughout the course which shaped the present work as it shows.

Last but not least our parents are also an important inspiration for us. So with due regards, we express our gratitude to them.

AMRITA VISHWA VIDYAPEETHAM
AMRITA SCHOOL OF ENGINEERING, AMRITAPURI
DEPARTMENT OF ELECTRICAL AND ELECTRONICS
ENGINEERING

DECLARATION

We, hereby declare that this project report, entitled "**MATRIX BARCODE BASED DETECTION AND TRACKING FOR AUTONOMOUS UAV LANDING**" is a record of the original work done by us under the guidance of **MR VIVEK A, ASSISTANT PROFESSOR**, Department of Electrical and Electronics Engineering, Amrita School of Engineering, Amritapuri and that this work has not formed the basis for the award of any degree/diploma/associateship/fellowship or a similar award, to any candidate in any University, to best of my knowledge.

Place: Amritapuri

Name and Signature of the students

Date: 25-04-2021

Velayudhan A

Neeraj S

Nandakishore R Nair

Vishnu S

COUNTERSIGNED

Name of the guide: Mr Vivek A
Assistant Professor, Dept. of Electrical &
Electronics Engineering

ABSTRACT

The project MATRIX BARCODE BASED DETECTION AND TRACKING FOR AUTONOMOUS UAV LANDING is focused on the autonomous landing of UAVs and MAVs which can pave the way for a plethora of applications of UAVs. Although UAVs are becoming a pervasive sight in today's world many of the application require specialised human control or supervision. Autonomous landing on moving platforms is essential for the self-deployment and recovery of MAVs, but it remains a challenging task for both autonomous and piloted vehicles.

Creating and deploying a drone having a fully automated operational feature along with the focus of reduction in economic expenditure, attaining more accuracy over the landing zone with the effect of external disturbances like wind, lighting etc is the focus of the research.

The system employs an effective and simple method by which the image data taken is analysed based on the best possible image processing algorithm for the effective architecture of detection tracking and landing of the drone.

The said system opens up new innovative technology for easy and efficient drone capabilities by expanding over the existing ones. The modelling is simply based on a closed feedback loop, and with the onboard sensors like camera, inertial measurement unit and much facilitates in the deployment of autonomous drone landing using a fiducial marker, in this case, an April Tag.

TABLE OF CONTENTS

Acknowledgement.....	3
Abstract.....	5
List of Tables.....	7
List of Figures.....	8
1.Introduction.....	9
1.1 Overview.....	9
1.2 Literature reviews.....	9
2.Quadcopter Model.....	15
2.1 Reference frame.....	15
2.2 Actuation Variables.....	16
3.Point feature extraction.....	18
3.1 Overview.....	18
3.2 Analogy of point feature extraction techniques.....	19
4. Software architecture.....	33
4.1 Drone Composition.....	33
4.2 Image Processing.....	35
4.3 Controlling Logic.....	36
5. Experimental Results and Analysis.....	37
6.Conclusion.....	41

List of Tables

- Table 3.1.....Time of feature extraction vs algorithm
Table 3.2.....Tilt-Time of feature extraction method vs time
Table 3.3.....Tilt-Feature Extraction Method vs matched points
Table 3.4.....Noise-Feature Extraction Method and time taken
Table 3.5.....Noise-Feature Extraction Method vs matched points
Table 3.6.....Scale-Feature Extraction Method and time taken
Table 3.7.....Scale-feature extraction method vs matched point

List of Figures

- Fig 2.1.....Coordinate frames
Fig 2.2.....Basic architecture of a Quadcopter
Fig 2.3.....Quadcopter actuators and output
Fig 3.1.....Reference Image
Fig 3.2.....Time of Feature Extraction Vs Algorithm
Fig 3.3.....Reference Image for BRISK
Fig 3.4.....Reference Image for MSER
Fig 3.5.....TILT-Feature Extraction method Vs Time
Fig 3.6.....Brisk -90 deg rotation and cross-matching
Fig 3.7.....Fast-90 deg rotation and cross
Fig 3.8.....Surf -90 deg rotation and cross
Fig 3.9.....TILT - Feature Extraction Method vs Matched Points
Fig 3.10.....NOISE - Feature Extraction Method VS Time
Fig 3.11.....NOISE - Feature Extraction Method vs matched points
Fig 3.12.....Kaze Gaussian Noise Cross-match Image
Fig 3.13.....ORB Median Noise Cross-match Image
Fig 3.14.....SCALE - Feature Extraction Method and time taken
Fig 3.15.....SCALE - feature extraction method vs matched point
Fig 3.16.....Surf 1.9x scaled Cross-match Image
Fig 4.1.....Drone Composition
Fig 4.2.....Image Processing Block
Fig 4.3.....Controlling algorithm using Matlab Stateflow
Fig 4.4.....Inside movement block
Fig 5.1.....Block Diagram in Simulink MatLab
Fig 5.2.....Depiction of drone in the environment
Fig 5.3.....Depiction of drone in the environment
Fig 5.4.....Depiction of drone in the environment
Fig 5.5.....Depiction of drone in the environment
Fig 5.6.....Graphical representation of Drone states

1. INTRODUCTION

1.1 OVERVIEW

The development in the fields of sensing and computation in the past few years have made the concept of Unmanned Aerial Vehicle an actualized aspiration. The shift in development from fixed-wing aerial vehicles to the multirotor aerial vehicle has helped expand the application of UAV to fields like military and meteorological purposes due to their simplicity of design, stability of hovering, and exceptional agility. Multicopters also contribute to the extension of the scope of application of existing ground-based robots making it a promising solution to many tasks that are dangerous, expensive or impossible to be carried out by humans. With such vast application in crucial sectors as well as expeditious growth in aerial vehicle technology autonomous control of these UAVs have become an imperative vision.

Quadcopters are one of the simplest variants from the multi-copter series that processes significant advantages in comparison to traditional aerial vehicles. Quadcopters are omnidirectional aerial vehicle propelled by four rotors and have an edge over the fixed-wing aerial vehicle in terms of manoeuvrability constraints. Although this has aided quadcopters to find application in inspection, agriculture, search-and-rescue missions, wildlife monitoring, and consumer goods delivery most of the task performed involve specialised human interaction or supervision. Though these scenarios pave way for autonomous control of the UAV, a relevant and challenging task to increase the level of autonomy is the ability for the UAVs to land on both statics and moving platforms.

The ability to land on moving platforms can contribute to the application of the UAVs as the UAV can autonomously land on charging pads and base stations contributing to overcome constraints like limited battery capacity. Autonomous landing requires effective detection and tracking of the landing pad as well as efficient guidance and navigation and control. The limitation in the amount of payload resulting in limited computational resources also contributes to the complexity of the task.

1.2 LITERATURE REVIEWS

The [Watsamon Hogpracha](#) and [Sartid Vongpradhip \[1\]](#) proposed a system that

reads QR code from a moving object by recording a video file. The moving object was subjected to different speeds and analysing video files. QR Code abbreviation for Quick Response Code was first developed by Denso Wave corporation. It can hold a significant amount of data in a 2D matrix format using function pattern and encoding region, along with contrast limited adaptive histogram equalization. In simple words, histogram equalisation is to stretch the histogram of an image to either side, as the brighter image have all images confined to high values.

Haotian Zhang, Gaoang Wang, Zhichao Lei and Jenq-Neng Hwang proposed a [2]multi-object tracking and 3D localization scheme is implemented based on the deep learning-based on object detection. A multi-object tracking method called TrackletNet Tracker (TNT) which utilizes temporal and appearance information to track detected objects located on the ground for UAV is applied can be used for the project. The following novelties such as Accurate object detection, Multi-object tracking, Visual odometry and ground plane estimation, 3D object localization can be implemented in our project.

Takashi Anezaki* ,**Koki Eimon*** ,**Suriyon Tansuriyavong***, and **Yasushi Yagi** [3] researched human-tracking robots. They implemented a tracking system that differentiates individuals using QR codes and tracks subjects using shape-based pattern matching. They also proposed a rediscovery process using an IR camera to handle cases when the QR code strays completely out of the camera field of vision.

There are three main requirements for realizing
a human-following robot:

- (a) Continuous identification of the person being followed.
- (b) Detection of the location of the person being followed.
- (c) Control of the robot based on the detected location data.

Since case (a) is difficult with image processing alone, QR-tag
is used for tracking.

Saha, Debjoy & Udayagiri, Balaji & Agarwal, Parakh & Ghosh, Biswajit &

Kumar, Somesh focused on developing the computer vision tools[4] for efficient inventory management of packages of a warehouse using QR code. An industrial camera with 120 fps is used to detect the QR code. The z-bar library and google text recognition tesseract are used for QR code recognition and text detection respectively.

In this paper, the process of QR code recognition is split into 3 stages. In the first stage, the acquired image is rescaled cropped and the distortions are removed by blurring. The image is cropped to convert the QR code part of the video feed into a jpg file. In the next stage, the image obtained is passed to the z bar library and the QR code is detected and decoded. In the next stage, the image is preprocessed and passed to tesseract for text recognition. In this project, this decoded information is used to arrange the warehouse goods in their respective places. This paper provides an overview of the two main methods used for specific object detection that is QR code detection and text detection and explains the necessary algorithms required for carrying out this process.

The paper provides an overview of the necessary tools for image recognition, QR code decoding and text recognition. The paper also briefs about the algorithm of preprocessing the acquired image which includes rescaling, distortion removal, cropping. The tools and information required for image recognition and decoding like z-bar and tesseract which is necessary for our project was obtained from the paper.

H. Zhang, C. Zhang, W. Yang and C. Chen focused on the navigation of autonomous robots in environments[5] without GPS support is the area least explored. An approach for localization and navigation for a mobile robot in an indoor environment using QR code recognition is proposed in this paper. In this paper, the possibility of QR codes as a landmark for autonomous robots is explored and explained.

In this paper QR codes are used as a landmark that provides relative location and directions to the autonomous robots or UAVs. Exposure adjustable cameras are used to provide the input feed to the robot.

Unlike the usual QR code detection scenario a different configuration of the z-bar library is used in this process. The video feed from the camera is acquired by the

open-cv for image processing and QR code detection when a QR code is detected an image is captured by the camera and this image is sent to the z-bar library after preprocessing. This method improves the accuracy by 20% because the z-bar is more precise while processing jpg images. The papers also state the possibility of using both the techniques simultaneously for higher accuracy and processing speed.

The paper explains a new and accurate method of QR code recognition and decoding. This paper provides a new dimension and application to QR code detection using UAV in terms of localization and navigation in routine paths as an alternative for the GPS module.

Y. Feng, C. Zhang, S. Baek, S. Rawashdeh, and A. Mohammadi presents a new autonomous landing method [6] that can be implemented on micro UAVs that require high-bandwidth feedback control loops for safe landing under various uncertainties and wind disturbances. In their system architecture, including dynamic modelling of the UAV with a gimbaled camera, implementation of a Kalman filter for optimal localization of the mobile platform, and development of model predictive control (MPC), for the guidance of UAVs. They demonstrate autonomous landing with an error of less than 37 cm from the centre of a mobile platform travelling at a speed of up to 12 m/s under the condition of noisy measurements and wind disturbances.

E. Olson describes a new visual fiducial system[7] that uses a 2D bar code style "tag", allowing full 6 DOF localization of features from a single image. Our system improves upon previous systems, incorporating a fast and robust line detection system, a stronger digital coding system, and greater robustness to occlusion, warping, and lens distortion.

Roland Brockers, Sara Susca, David Zhu and Larry Matthies present a micro air vehicle that uses vision feedback[8] from a single down looking camera to navigate autonomously and detect an elevated landing platform as a surrogate for a rooftop. Their method requires no special preparation (labels or markers) of the landing location. Rather, leveraging the planar character of urban structure, the landing platform detection system uses a planar homography decomposition to detect landing targets and produce approach waypoints for autonomous landing.

The vehicle control algorithm uses a Kalman filter-based approach for pose estimation to fuse visual SLAM (PTAM) position estimates with IMU data to correct for high latency SLAM inputs and to increase the position estimate update rate to improve control stability. Scale recovery is achieved using inputs from a sonar altimeter. In experimental runs, we demonstrate a real-time implementation running on-board a micro aerial vehicle that is fully self-contained and independent from any external sensor information. With this method, the vehicle can search autonomously for a landing location and perform precision landing manoeuvres on the detected targets.

Yang, S., Scherer and S.A. & Zell present an onboard monocular vision system[9] for autonomous takeoff, hovering and landing of a Micro Aerial Vehicle (MAV). Since pose information with metric scale is critical for autonomous flight of a MAV, a novel solution to six degrees of freedom (DOF) poses estimation is developed, based on a single image of a typical landing pad which consists of the letter “H” surrounded by a circle. A vision algorithm for robust and real-time landing pad recognition is implemented. Then the 5 DOF pose is estimated from the elliptic projection of the circle by using projective geometry. The remaining geometric ambiguity is resolved by incorporating the gravity vector estimated by the inertial measurement unit (IMU). The last degree of freedom pose, yaw angle of the MAV, is estimated from the ellipse fitted from the letter “H”. The efficiency of the presented vision system is demonstrated comprehensively by comparing it to ground truth data provided by a tracking system and by using its pose estimates as control inputs to autonomous flights of a quadrotor.

P. Serra, R. Cunha, T. Hamel, D. Cabecinhas and C. Silvestre addresses the landing problem of a vertical take-off and landing vehicle[10], exemplified by a quadrotor, on a moving platform using image-based visual servo control. Observable features on a flat and textured target plane are exploited to derive a suitable control law. The target plane would be moving with bounded linear acceleration in any direction. For control purposes, the image of the centroid for a collection of landmarks is used as position measurement, whereas the translational optical flow is used as velocity measurement. The proposed control law guarantees convergence to the desired landing spot on the target plane, without

estimating any parameter related to the unknown height, which is also guaranteed to remain strictly positive. Moreover, convergence is guaranteed even in the presence of bounded and possibly time-varying disturbances, resulting, for example, from the motion of the target plane, measurement errors, or wind-induced force disturbances. To improve performance, an estimator for unknown constant force disturbances is also included in the control law.

T. Templeton, D. H. Shim, C. Geyer and S. S. Sastry present a vision-based terrain mapping and analysis system[11], and a model predictive control (MPC)-based flight control system, for autonomous landing of a helicopter-based unmanned aerial vehicle (UAV) in unknown terrain. The vision system is centred around Geyer et al.'s recursive multi-frame planar parallax algorithm (2006), which accurately estimates 3D structure using geo-referenced images from a single camera, as well as a modular and efficient mapping and terrain analysis module. The vision system determines the best trajectory to cover large areas of terrain or to perform a closer inspection of potential landing sites, and the flight control system guides the vehicle through the requested flight pattern by tracking the reference trajectory as computed by a real-time MPC-based optimization. This trajectory layer, which uses a constrained system model, provides an abstraction between the vision system and the vehicle. Both vision and flight control results are given from flight tests with an electric UAV.

D. Lee, T. Ryan and H. J. Kim they describe a vision-based algorithm to control a vertical-takeoff-and-landing unmanned aerial vehicle while tracking and landing on a moving platform[12]. They use image-based visual servoing (IBVS) to track the platform in two-dimensional image space and generate a velocity reference command used as the input to an adaptive sliding mode controller. Compared with other vision-based control algorithms that reconstruct a full three-dimensional representation of the target, which requires precise depth estimation, IBVS is computationally cheaper since it is less sensitive to the depth estimation allowing for a faster method to obtain this estimate. To enhance velocity tracking of the sliding mode controller, an adaptive rule is described to account for the ground effect experienced during the manoeuvre. Finally, the

IBVS algorithm integrated with the adaptive sliding mode controller for tracking and landing is validated in an experimental setup using a quadrotor.

Mr Vivek A, Velayudhan A, Neeraj S, Nandakishore R Nair and Vishnu S, we are comparing the different available point feature extraction techniques currently in existence based on the scale changes, noise, rotation etc[13]. The evaluation is based on the number of feature points from the reference image as well as matched points in the wild frame and elapsed time. The data and information from the study open up new research in the field of image processing based on point feature extraction.

2. QUADCOPTER MODEL

2.1 REFERENCE FRAME

We are using two reference frames namely world frame W and quadcopter's body frame D which can be used to represent the rotational matrices that map the coordinates from the D frame to the W frame. The W frame can be defined as a fixed referential in which W_x points towards north, Waypoints towards the west and W_z points upwards. D frame is represented as a fixed referential attached to the centre of mass of the quadcopter in which D_x points towards preferred forward direction and D_z points vertically upwards during an ideal hovering.

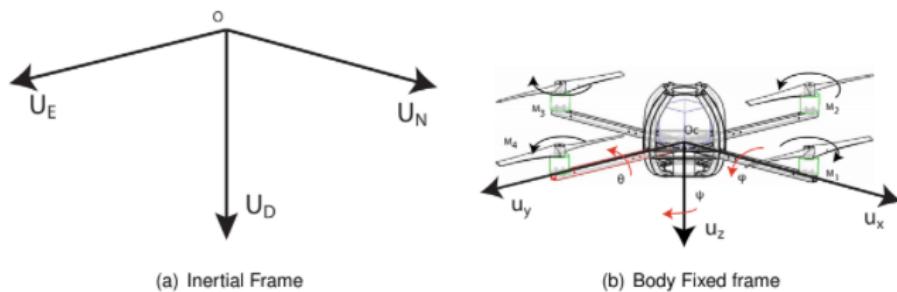


Fig 2.1: Coordinate frames a) Inertial frame b) Body Fixed frame

The matrix that represents the coordinate mapping between the W frame and the D frame can be represented as

$$\text{Rotational matrix} = R(y) \cdot R(p) \cdot R(r)$$

where y p and r are angles of yaw, pitch and roll respectively

$$\begin{aligned}
 &= \begin{bmatrix} Cy & -Sy & 0 \\ Sy & Cy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Cp & 0 & Sp \\ 0 & 1 & 0 \\ -Sp & 0 & Cp \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & Cr & -Sr \\ 0 & Sr & Cr \end{bmatrix} \\
 \text{Rotational matrix} &= \begin{bmatrix} CyCp & CySpSr-SyCr & SySr+CySpCr \\ SyCp & CyCr+SySpSr & SySpCr-CySr \\ -Sp & CpSr & CpCr \end{bmatrix}
 \end{aligned}$$

Where Cx represent Cos(x) and Sx represents Sin(x)

2.2 ACTUATION VARIABLES

As quadcopters are systems with six degrees of freedom that are controlled with four motors we need to redefine the actuation variables to compensate for this situation. This is done by four different actions namely throttle, yaw, pitch and roll.

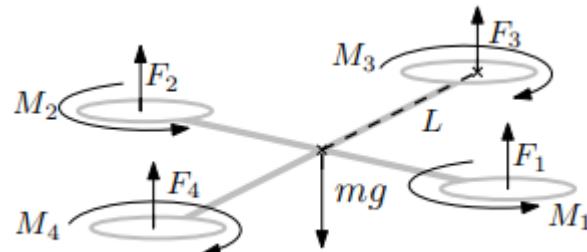


Fig 2.2: Basic architecture of a Quadcopter

Both quadrotors and helicopters are rotor-craft vehicles able to hover. The advantages of the quadrotors over the helicopters are not obvious. In this section, the concept of quadrotors is further analyzed and its manoeuvring capabilities are described. Standard helicopters have two rotors, the main one, located over the vehicle, produces the lift. The

The second rotor is located on the tail and cancels the torque produced by the main one. This allows the helicopter to yaw by simply changing the velocity of the tail rotor. To pitch or roll, the helicopter is equipped with a complex system that changes the angle of attack of the blades of the main rotor. Quadrotors have four identical rotors and the propellers have a fixed angle of attack. The blades are paired and each pair rotates in a different direction. Motors M1 and M3 have a

clockwise rotation when looked at above whilst motors M2 and M4 have a counter-clockwise rotation. To obtain the manoeuvres depicted in Figure 2.1(b) the speed of each motor is adjusted. The angular speeds of the motors are written $\omega = [\omega_1, \omega_2, \omega_3, \omega_4]$

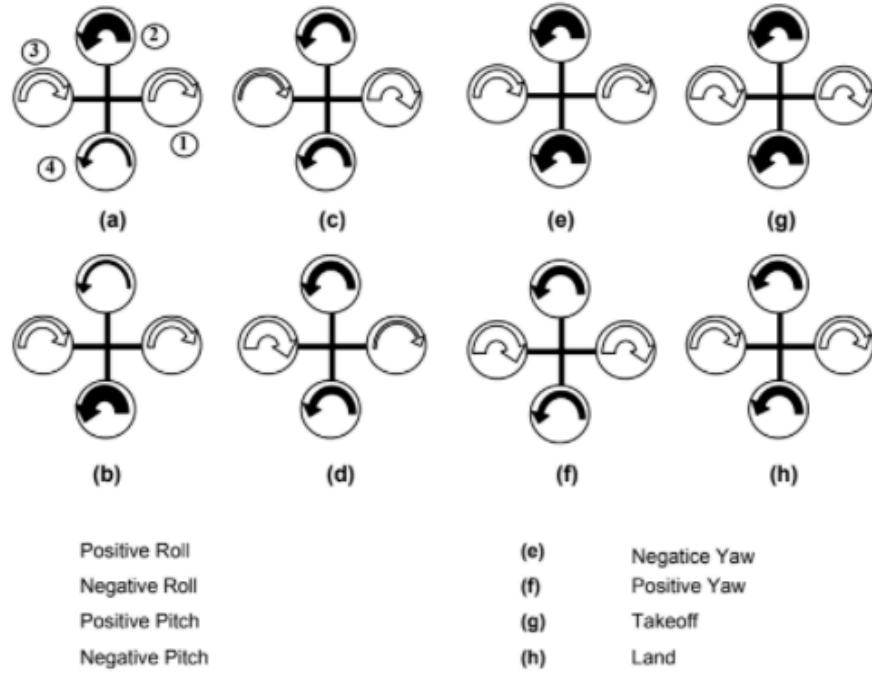


Fig 2.3: Quadcopter actuators and output

Throttle

Throttle can be defined as the force that is applied parallel to the Dz direction of the D frame that provides the necessary thrust to keep the UAV airborne. Throttle can be expressed as the sum of thrust vectors produced by the motors in the Dz direction and if all the forces F_i are balanced then the thrust will result in vertical acceleration.

Pitch

Pitch can be defined as the forward or backward tilt of the quadcopter in the Dx axis of the D frame. This movement occurs due to a force difference between the motors M1 and M2 resulting in the forward or backward motion of the quadcopter.

Roll

The sideways tilt of the quadcopter in the Dy axis of the D frame resulting in the motion of the quadcopter in the Dx Dy plane is called Roll. The roll motion is caused by a difference in the thrust of the motors M3 and M4 resulting in sideways tilted motion of the quadcopter.

Yaw

Every motor in the quadcopter induces a moment on the quadcopter in a direction opposite to the direction of rotation of the motor. Thus a difference in the combined clockwise moment(M3, M4) and anticlockwise(M1, M2) moment will cause the quadcopter to rotate in the Dz axis. This motion of the quadcopter is called yaw.

3.POINT FEATURE EXTRACTION

3.1 OVERVIEW

The human brain is capable of extracting, processing and manipulating data efficiently as information based on the neural pathways residing over our nervous system with the influence of neurons. However, compared to the machine point of view this concept of identification is relatively harder as they are trained to perform over an algorithm for fast computation. For example, computers can perform large amounts of calculation in milliseconds which would be a lifetime process for human brains. In contrast to that, humans can identify, differentiate, categorise the world around them which is harder from the machine's perspective. The artificial intelligence concept where the computers think like humans is the near future, but for the computer vision to attain the capability to identify, differentiate, categorise the world, it has to be trained under supervised or unsupervised learning. Image processing with feature extraction is an inevitable part of the machine training process and it is essential to find the most efficient feature extraction algorithm for every possible scenario. The concept of feature extraction works significantly in hand with image processing.

Image processing is a significant section of robotic and machine learning and plays an inevitable role in the autonomous landing of UAVs. The field of image processing is significant in terms of usage in the advanced world ranging from security, facial recognition to remote autonomous terrain mapping. The salient problem in image processing is how to detect, identify and recognise the feature points in an image. However, there are numerous feature point extraction techniques and it is essential to explore these methods to find the optimum method for each scenario to attain the best results.

3.2 ANALOGY OF POINT FEATURE EXTRACTION TECHNIQUES

For finding the best feature extraction method for autonomous landing of UAV a point feature extraction experiment was conducted with eight prominent feature point extraction techniques namely BRISK, FAST, HARRIS, KAZE, MINEIGEN, MSER, ORB, SURF and the key values like time of feature extraction, several the the match points, time is taken for cross-comparison. The implementation was done on Intel® core(TM) i7 10th gen processor with 16GB RAM and speed of 2.6-5GHz. The code was written in Matlab R2020b on Windows 10 64 bits. The experiment consists of various tests by introducing effects like rotation, scale change and noise. The sample picture considered for all the tests is shown below having a size of 2.2Mb.



Fig 3.1: Reference image used for image processing comparison

The first section tabulates the time required for feature extraction of each point feature extraction methods namely BRISK, FAST, HARRIS, KAZE, MINEIGEN, MSER, ORB, SURF.

ALGORITHM	TIME OF FEATURE EXTRACTION(sec)
BRISK	0.313587
FAST	0.074147
HARRIS	0.0853465
KAZE	4.763558
MINEIGEN	0.91857
MSER	1.010411
ORB	0.148321
SURF	0.334866

Table 3.1: Time of feature extraction vs algorithm

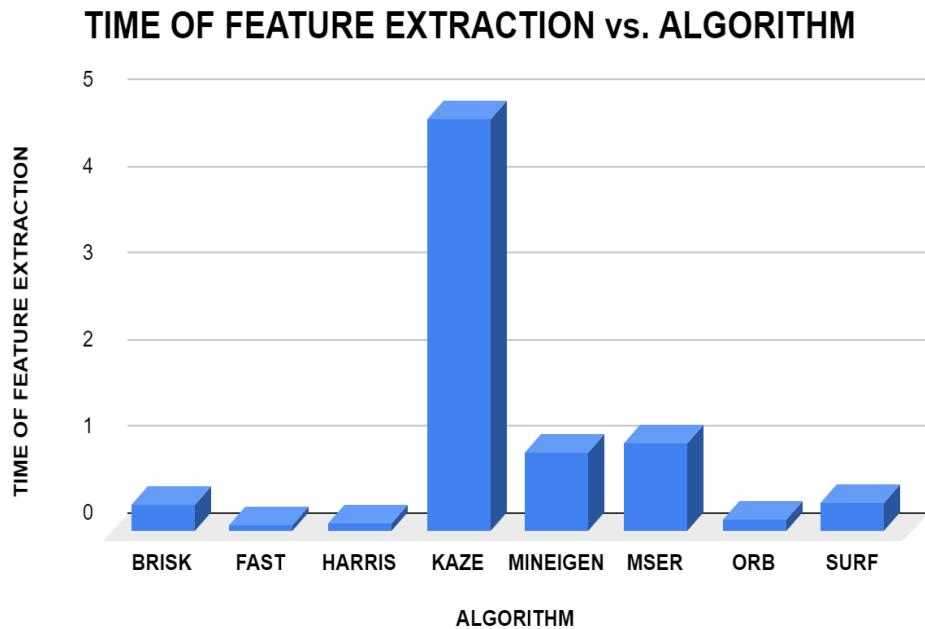


Fig 3.2: Time of feature extraction vs algorithm

The graphs are plotted between different types of feature extraction algorithms and the time taken by these algorithms to extract the feature points from a given reference image. It is evident from the graph that the highest time for feature extraction was for KAZE and the lowest time was recorded for the FAST feature extraction method. Although KAZE required a higher computation time, the number of matched points was comparatively higher for KAZE, emphasising the linear relation between computation time and the number of matched points. From the above visualisation, it can be observed that the SURF feature extraction method requires comparatively less computation time in all variants of inputs and provides a relatively higher number of match points in most cases. Thus it can be concluded that SURF is befitting in a situation where the raw inputs are provided.

The second section is sub categorised into three parts. The first part determines the time for evaluation of matching points and the number of matched points for images having angular distortion. For the experiment distortion angles of 0, 45, 90 degrees have been considered.

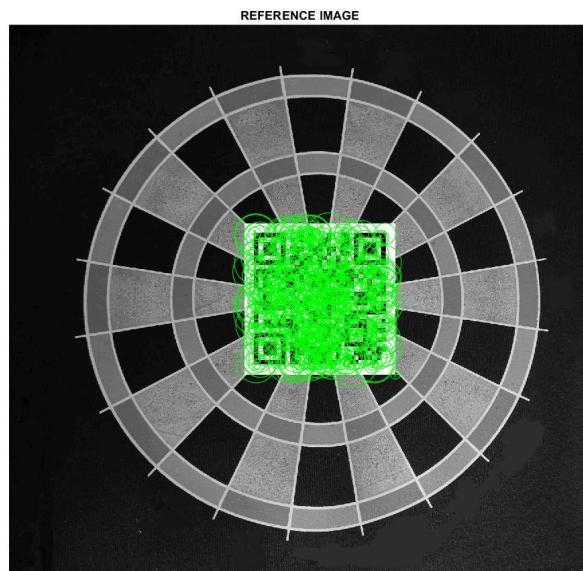


Fig 3.3: BRISK Reference image

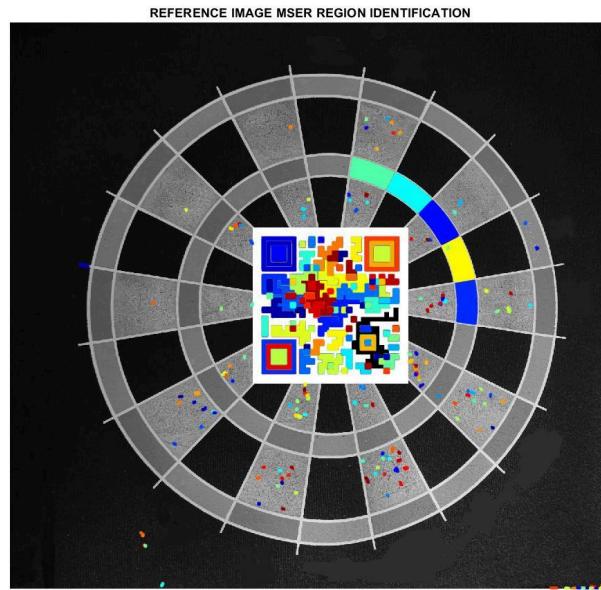


Fig 3.4: MSER Reference image

TILT - FEATURE EXTRACTION METHOD VS TIME			
ALGORITHM/ TILT ANGLE	45° TILT	90° TILT	0° TILT
BRISK	0.057429	0.046397	0.048213
FAST	0.064922	0.060837	0.067189
HARRIS	0.057784	0.055006	0.063457
KAZE	0.02712	0.027696	0.026556
MINEIGEN	0.053279	0.050198	0.05072
MSER	0.029002	0.031249	0.029311
ORB	0.06573	0.064043	0.05838
SURF	0.039339	0.03423	0.044163

Table 3.2: TILT - Time of feature extraction method vs time

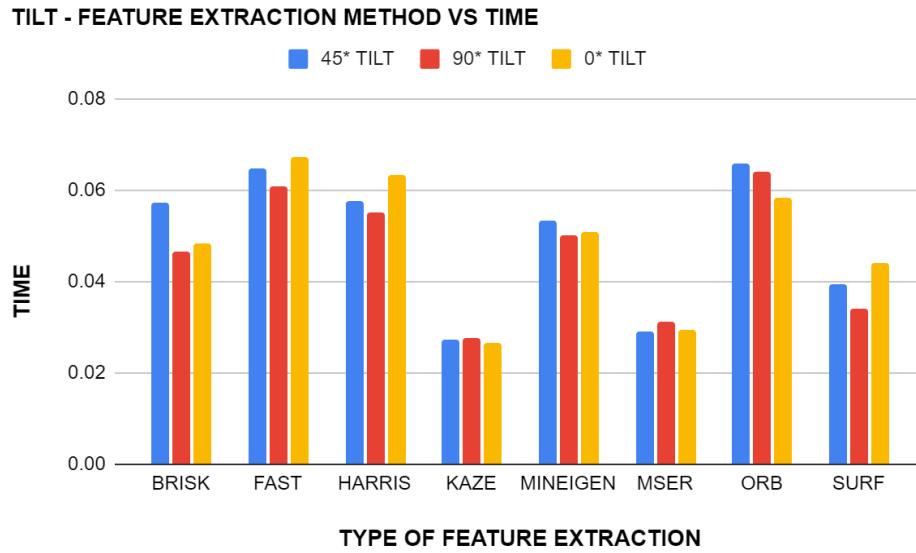


Fig 3.5: Type of Feature Extraction vs Time

The above graph is plotted between the different feature extraction algorithms and the time taken by these algorithms to cross-match the interesting point in the reference image to the points in the variants of the reference image with different tilts of $0^\circ, 45^\circ, 90^\circ$. As visible from the graph, the KAZE algorithm was able to cross-match all three cases of tilt with the reference image within a time range of 0.026 to 0.027 seconds. It can also be observed that the time taken by KAZE to cross-match is almost consistent for all the three types of tilt image inputs whereas methods like ORB show large variation time taken for different tilts. Moreover, the highest time of computation for the cross-matching process was observed in ORB and Fast methods making these methods less suitable for scenarios that require high computational efficiency and quick outputs.

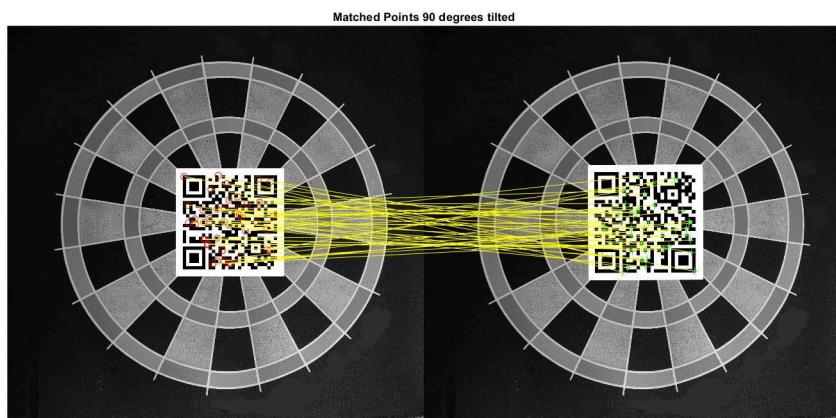


Fig 3.6: Brisk 90-degree rotation and cross-matching

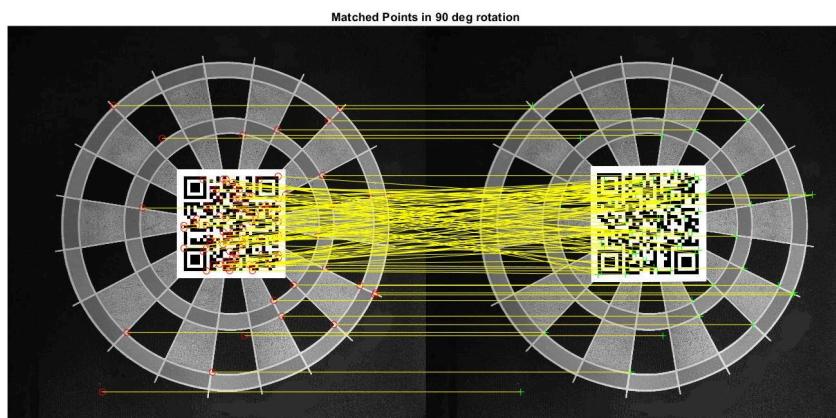


Fig 3.7: Fast 90 deg rotation and cross-matching

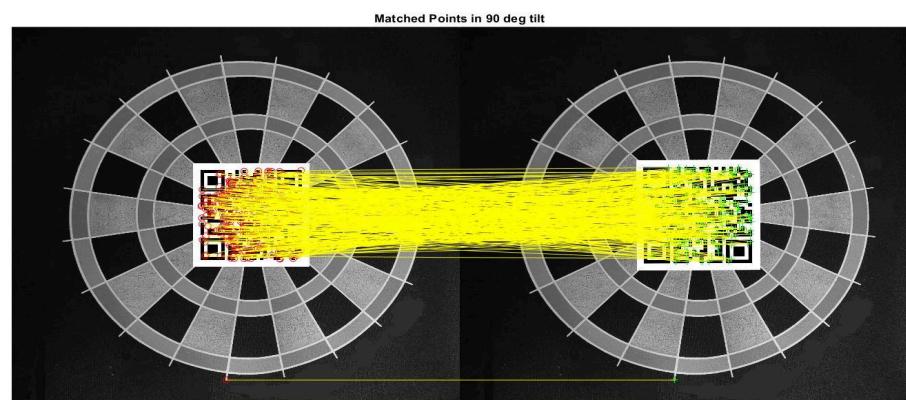


Fig 3.8: Surf 90 deg rotation and cross-matching

TILT - FEATURE EXTRACTION METHOD VS MATCHED POINT			
Algorithm	45° Tilt	90° Tilt	0° Tilt
BRISK	35	47	496
FAST	25	118	483
HARRIS	3	31	498
KAZE	44	41	500
MinEigen	3	46	500
MSER	140	193	463
ORB	41	41	500
SURF	142	327	500

Table 3.3: TILT - Feature Extraction Method and number of matched points

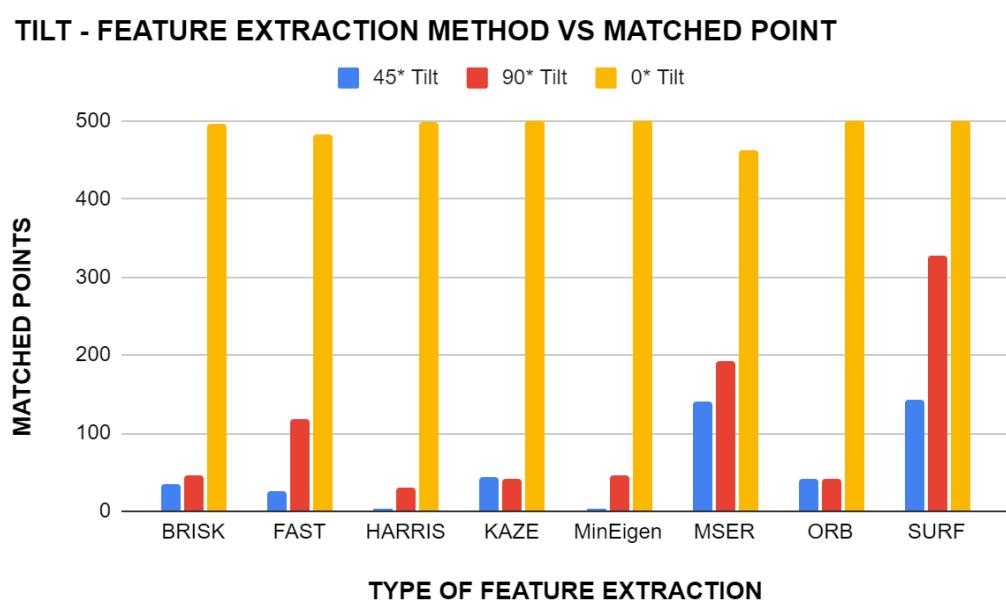


Fig 3.9: TILT - Feature Extraction Method vs Matched Points

The above graph is plotted between the different feature extraction algorithms and the number of matched points. As evident from the graph, the number of match points is maximum at 0° tilt for all the different types of feature extraction. Comparing the above two graphs of tilt input scenario it can be concluded that ORB requires higher computational infrastructure compared to methods like KAZE and MSER as the latter method was able to extract approximately the same number of interest points within one by the third fraction of time consumed by ORB and FAST methods. Also, contrary to the prior traits SURF has cross-matched the maximum number of points in all three types of tilt inputs.

The second part determines the time for evaluation of matching points and the number of matched points for images having noises. For the experiment, common noises like median, Gaussian and Dust and scratches are considered.

NOISE - FEATURE EXTRACTION METHOD VS TIME			
ALGORITHM	MEDIAN NOISE	GAUSS NOISE	DUST AND SCRATCH
BRISK	0.048426	0.044872	0.041388
FAST	0.047351	0.041478	0.04614
HARRIS	0.054686	0.05249	0.05528
KAZE	0.025917	0.026505	0.027484
MINEIGEN	0.054124	0.051961	0.050279
MSER	0.030573	0.035674	0.037759
ORB	0.060046	0.065243	0.066454
SURF	0.037223	0.033108	0.038761

Table 3.4: NOISE - Feature Extraction Method and time taken

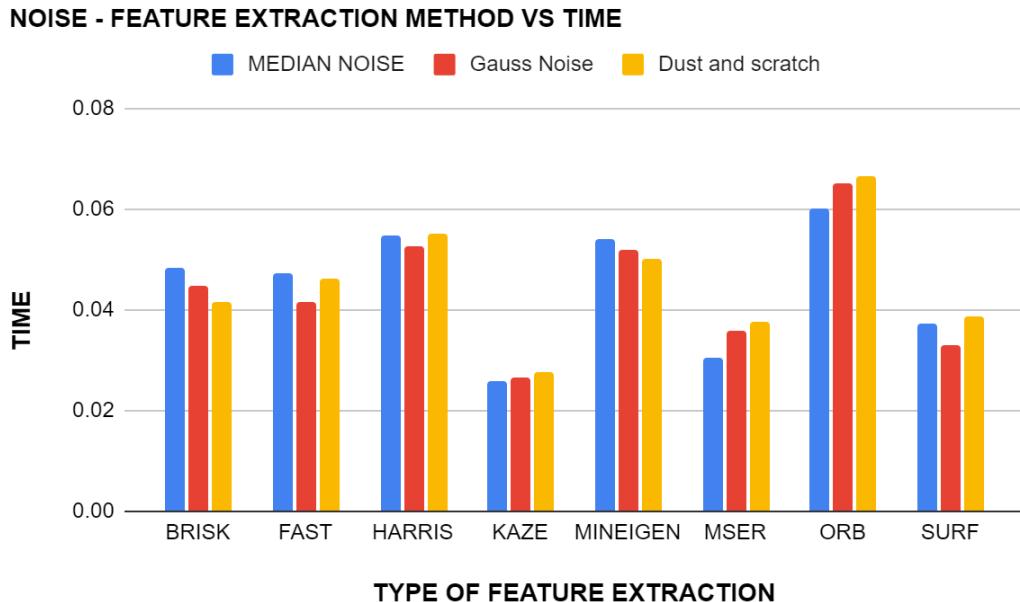


Fig 3.10: NOISE - Feature Extraction Method VS Time

The graph has been plotted for noise-feature extraction of three different types of noise against the time taken for these extractions by each method. From the graph, we can observe that KAZE took the least time, between 0.0259sec (for median noise) and 0.0274sec (for Dust and Scratch) for feature extraction for all three types of noise, followed by MSER and SURF. The time taken by KAZE is also consistent when compared to the other methods where the time taken differs from each other by a greater value. We can also observe that ORB is the method that took the longest for extracting the features for all three types of noises and thus making it the least suitable among the bunch if the images to be cross-matched has a significant amount of noise.

NOISE - FEATURE EXTRACTION METHOD VS MATCHED POINT			
ALGORITHM	MEDIAN NOISE	GAUSS NOISE	DUST AND SCRATCH
BRISK	15	103	13
FAST	0	20	1
HARRIS	4	9	2
KAZE	28	162	49
MINEIGEN	2	3	3
MSER	18	12	14
ORB	34	8	34
SURF	94	259	94

Table 3.5: NOISE - Feature Extraction Method and number of matched points

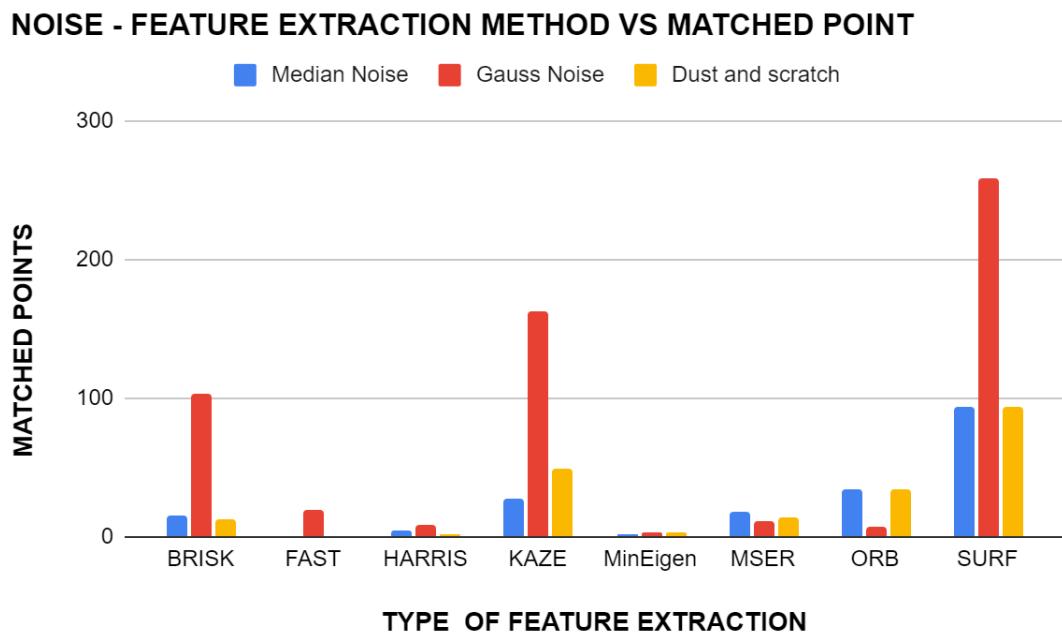


Figure 3.11: NOISE - Feature Extraction Method vs matched points

The graph has been plotted for noise-feature extraction of three different types of noise against the number of matched points they were able to come up with for the different method. It can be inferred from the graph that SURF has the highest number of cross-matched points with Gaussian noise having the highest number among the different noises. KAZE and BRISK are the methods that have some comparable stats and all the other methods falling behind them by a large margin. So when it comes to extracting the feature points from the images with noise, SURF is the best. If we are to combine time taken and the number of points they were able to come up with in that period, SURF is the best available choice for cases where noise is there.

The third part determines the time for evaluation of matching points and the number of matched points for images having scaled distortion. For the experiment distortion angles of 120, 150, 170, 190 percentages of the reference image has been considered.

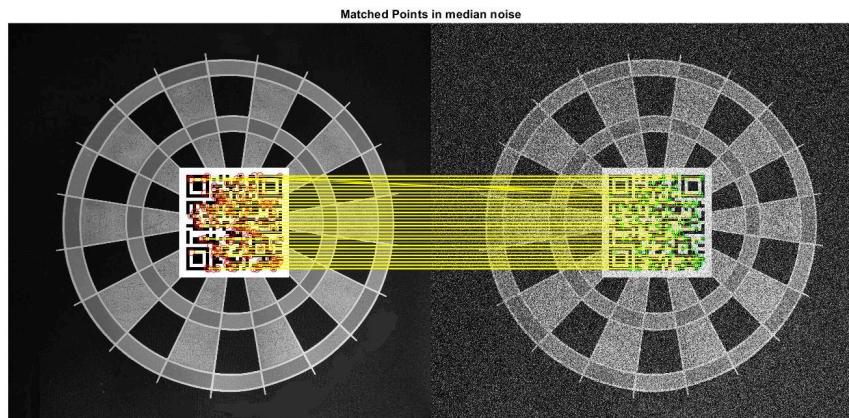


Fig 3.12: Cross-matching did with KAZE in Gaussian Noise

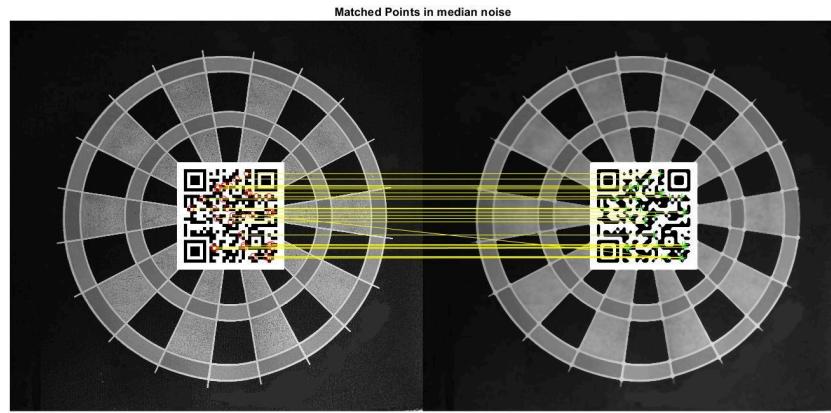


Fig 3.13: Cross-matching did with ORB in Median Noise

SCALED - FEATURE EXTRACTION METHOD VS TIME				
ALGORITHM/ SCALED FACTOR	120% SCALE	150% SCALE	170% SCALE	190% SCALE
BRISK	0.0439	0.04487	0.049317	0.042299
FAST	0.046347	0.043665	0.047194	0.047015
HARRIS	0.059786	0.052826	0.053238	0.053117
KAZE	0.026761	0.028547	0.026327	0.025562
MINEIGEN	0.051124	0.049958	0.052885	0.051355
MSER	0.042234	0.041882	0.030799	0.030086
ORB	0.062399	0.063155	0.070053	0.064498
SURF	0.037279	0.034314	0.038728	0.035153

Table 3.6: SCALE - Feature Extraction Method and time taken

SCALED - FEATURE EXTRACTION METHOD VS TIME

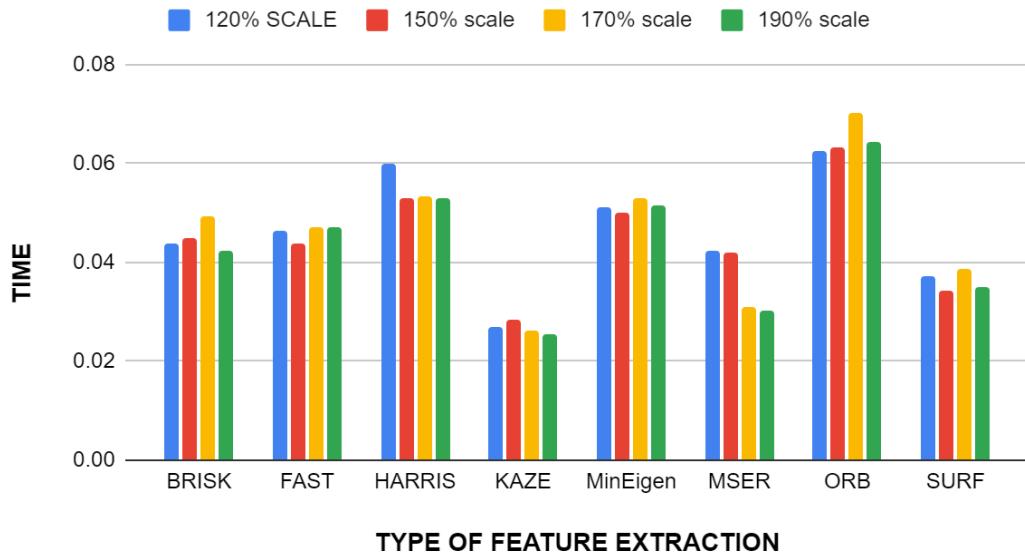


Fig 3.14: SCALE - Feature Extraction Method and time taken

The above graph is plotted between different feature extraction algorithms and the time taken by these algorithms for feature extraction. The scaled variants of the Preference image were provided as the input for plotting the graphs. It is evident from the graph that the KAZE method has the least computation time ranging from 0.026 to 0.028 whereas the ORB method requires the maximum computation time ranging from 0.062 to 0.070. It is also noteworthy that the SURF method maintained its trait by requiring a computation time slightly above the method with the least computation time.

SCALED - FEATURE EXTRACTION METHOD VS MATCHED POINT				
ALGORITHM	120% SCALE	150% SCALE	170% SCALE	190% SCALE
BRISK	35	23	19	7
FAST	45	50	14	6
HARRIS	25	8	2	0
KAZE	76	23	7	13
MINEIGEN	28	5	4	4
MSER	67	29	41	28
ORB	61	42	71	25
SURF	200	228	177	172

Table 3.7: SCALE - feature extraction method vs matched point

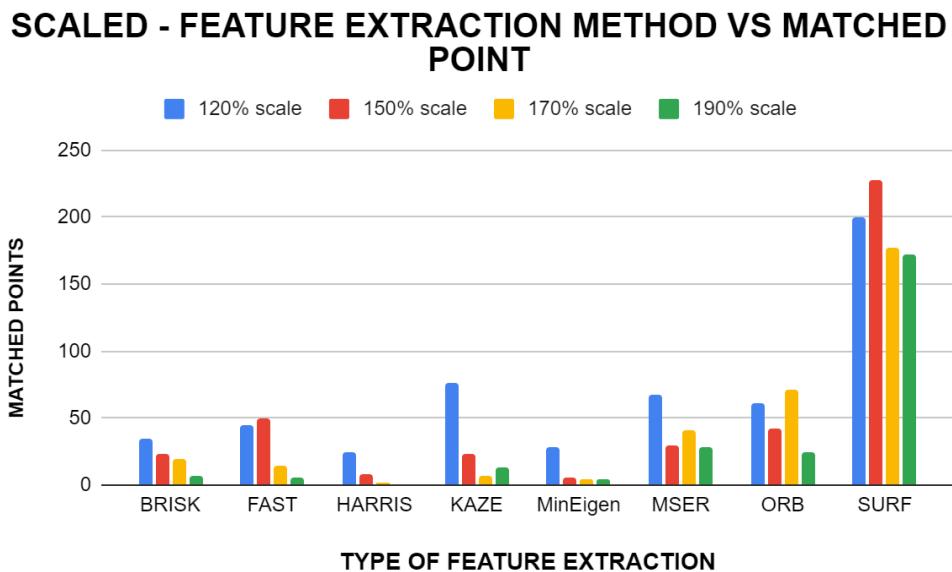


Fig 3.15: SCALE - feature extraction method vs matched point

The above figure shows the graph between different feature extraction algorithms and the maximum number of matched points of these algorithms. Scaled image variants of the reference image were provided as the input in the above graph and

the results were as expected with SURF uncovering the maximum number of match points. Although the KAZE method required the least amount of computation time (0.026-0.028) it was only able to locate relatively very few matchpoints. As evident from the above two graphs, SURF was able to locate the maximum number of match points, within a comparatively lesser amount of time. Thus it can be concluded that for scaled input variants SURF has the maximum computational efficiency.

As observed from the different graphs on time and number of match points for different types of input, the SURF method was able to attain adequate efficiency in all types of input. Thus it can be concluded that SURF is the best method for our scenario.

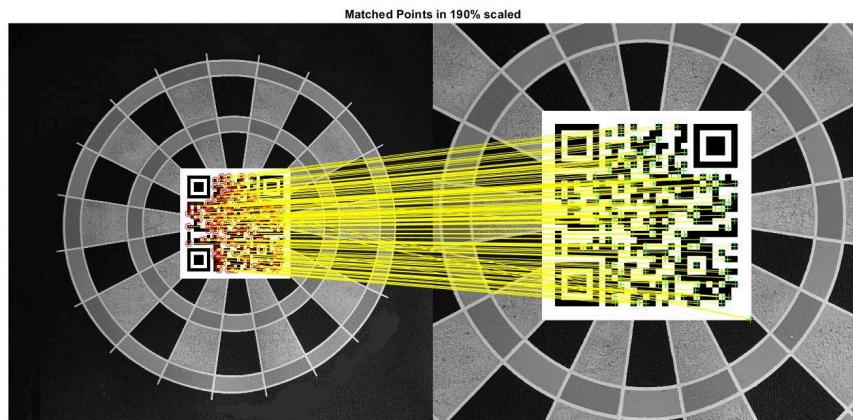


Fig 3.16: Cross-matching did with SURF with 190 percentage scaled image

4. SOFTWARE ARCHITECTURE

4.1 DRONE COMPOSITION

The composition of the drone includes a controller, the plant and sensors. The controller is an onboard processor with the capabilities of performing the simultaneous operation of both the drone actuator commands as well as the image

processing for the UAV used in the project objective. The whole system is divided into 3 main segments.

1. Controller

The controller part of the drone includes the flight control system(FCS), Image processing system, State estimator, path planning, landing system algorithm, PID controllers etc. It is the work that is done on the processor onboard for the UAV to propagate and do other tasks.

2. Plant

The plant segment consists of the motors, the sensors, the environment designs etc. It is the external components of the simulation environment like the drone dynamics, the environment in which the drone hovers.

3. Sensors

The sensors are the part of the external element used in the feedback loop which includes the inertial measurement unit, camera, ultrasound sensor, pressure sensor etc.

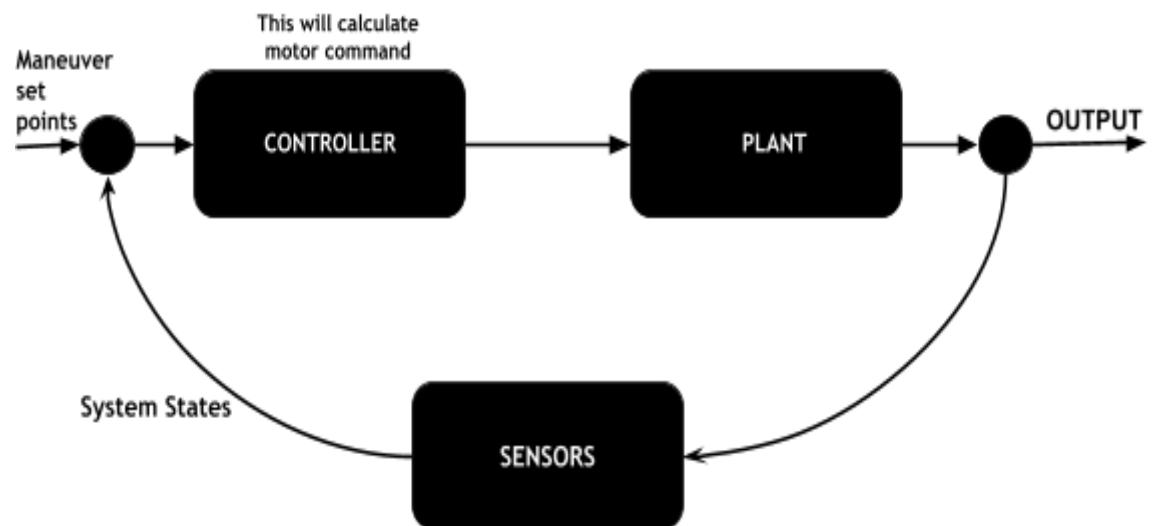


Fig 4.1: Flowchart for the Drone

4.2 IMAGE PROCESSING

The image processing block is where the cross-matching and detection of the April Tag take place. The image feed from the drone is in the Y1UY2V format rather than the normal RGB format. So we first convert the image feed from the drone camera to RGB. Now we have the R, G and B components separately and to combine them together to get our normal RGB image, we are using a MatLab function. This image is then fed to another MatLab function where the image is cross-matched with a reference image of the April Tag that we have already given. Now the output of the function is a number that indicates the total cross-matched points from both images. This number is then given to a comparator which compares it with a threshold value that we specify and gives out a boolean 1 if the function value is greater than the threshold. This comparator output is then fed to the next Control Block for the manoeuvring of the Drone.

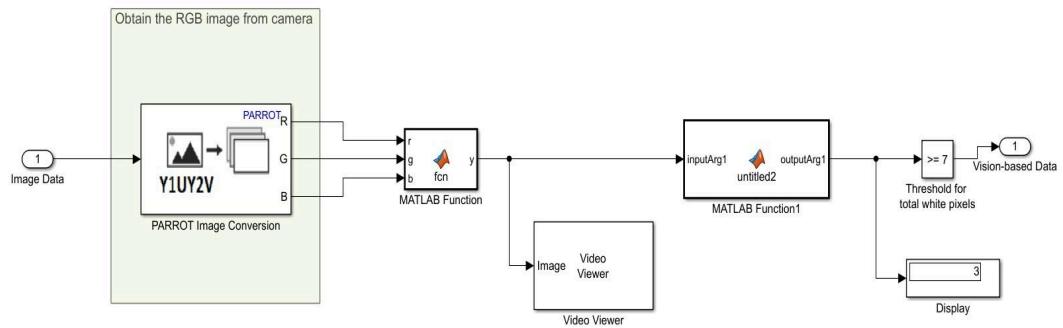


Fig 4.2: Image Processing Block

4.3 CONTROLLING LOGIC

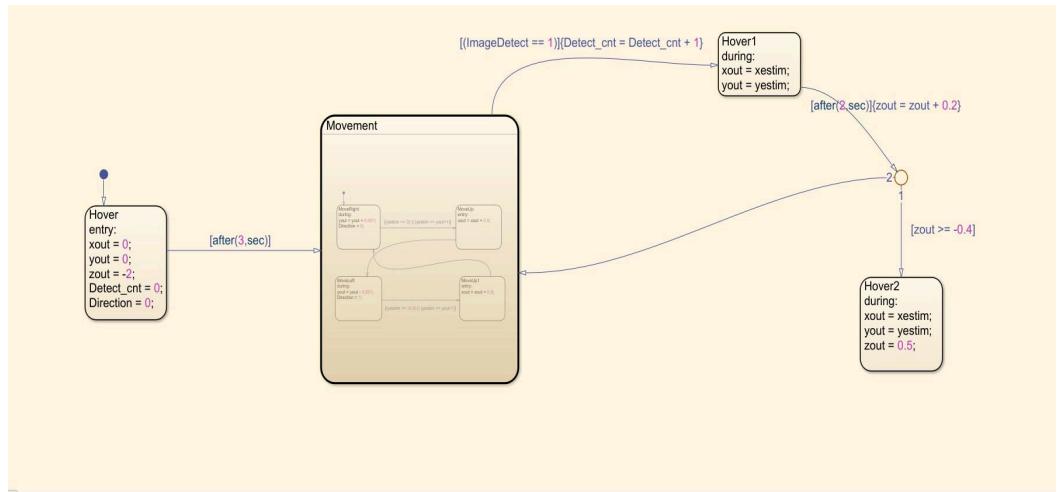


Fig 4.3: Controlling algorithm using Matlab Stateflow

First, the drone is made to hover at a particular height from the ground. Then after a small time period, it starts moving to the right till the boundary of our custom environment and if it reaches the boundary, the drone moves a small distance forward and then starts moving to the left. This movement is repeated till it detects(using the image processing) the April Tag which is placed in the environment as the landing platform.

Once the drone detects the Tag, the drone hovers for two seconds and then comes down a certain height. It then checks whether the height is below a certain threshold. If it is, then the drone will be at a height from which it could be landed upon the platform. If the height is above the threshold value, the drone will continue its movement using the same logic as before till it again detects the April Tag. This process keeps on repeating till the drone comes below the threshold height and land successfully.

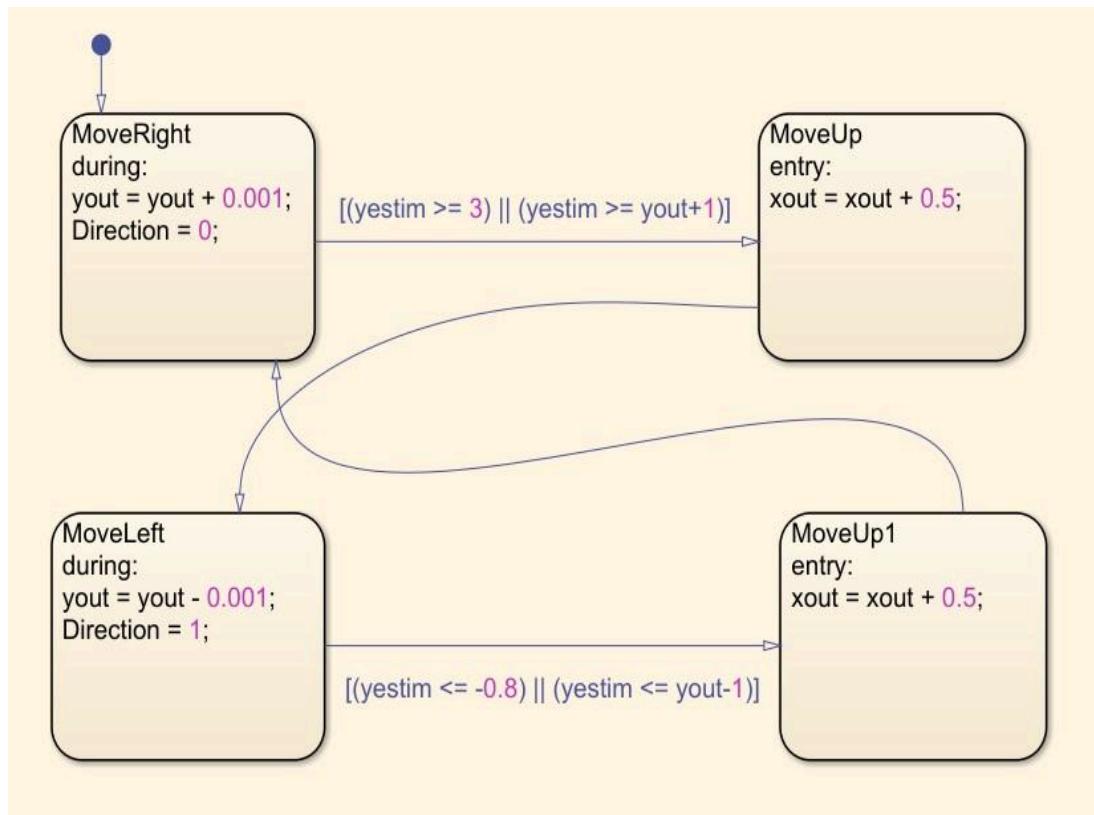


Fig 4.4: Landing algorithm in Stateflow chart

5. EXPERIMENTAL RESULTS AND ANALYSIS

The block diagram for the project is shown below done with Intel® core(TM) i7 10th gen processor with 16GB RAM and speed of 2.6-5GHz. The code was written in MATLAB R2021a on Windows 10 64 bits system.

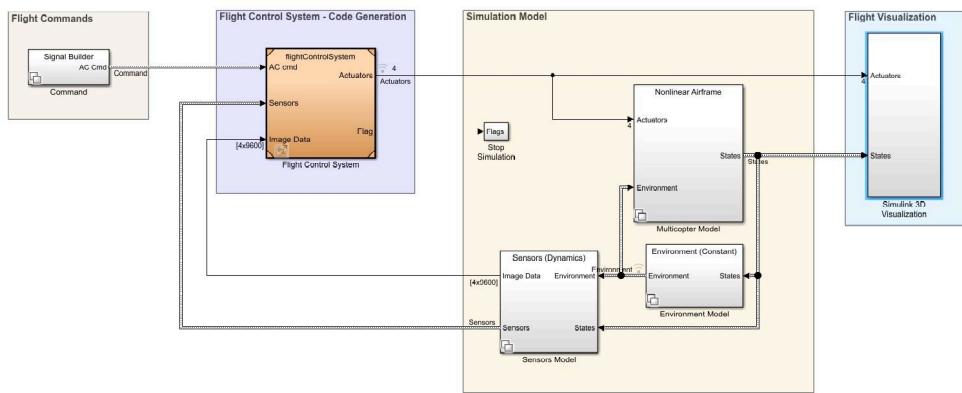


Fig 5.1: Block Diagram in Simulink MatLab

The shot of the UAV in the environment is illustrated below.

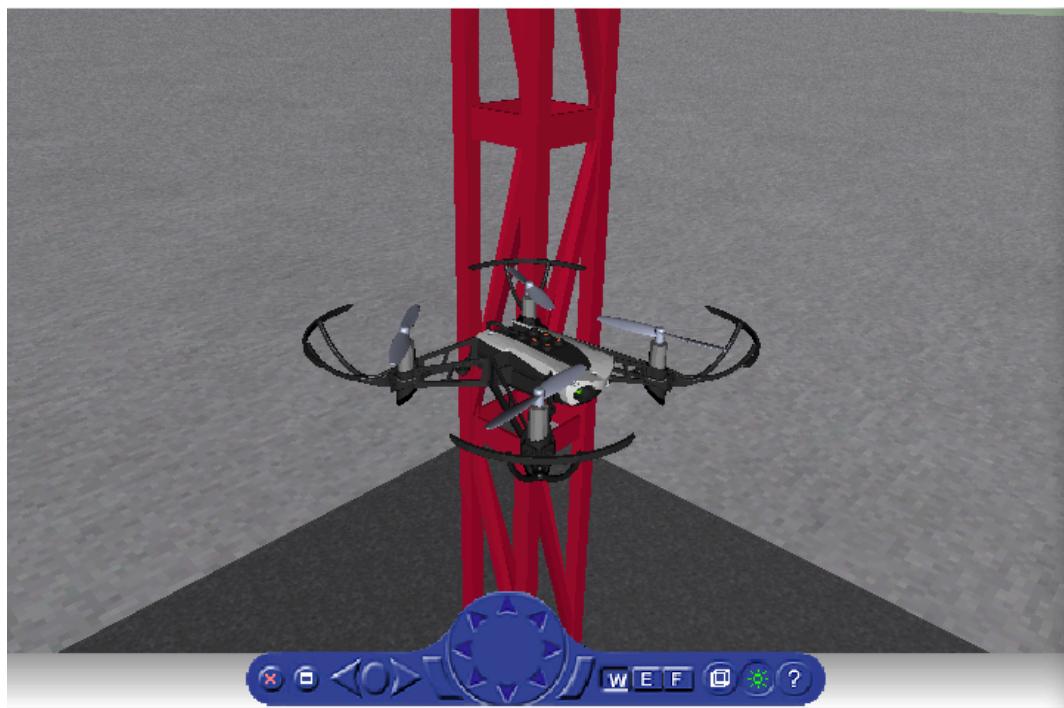


Fig 5.2: Depiction of drone in the environment within the simulation

The environment modelling is done on a 4X4 cage with an AprilTag which is considered as the landing pod. For analysing the system for variations in detection pattern extra platforms like red-coloured strips, a red circular platform, a blue landing pad are constructed within the environment.

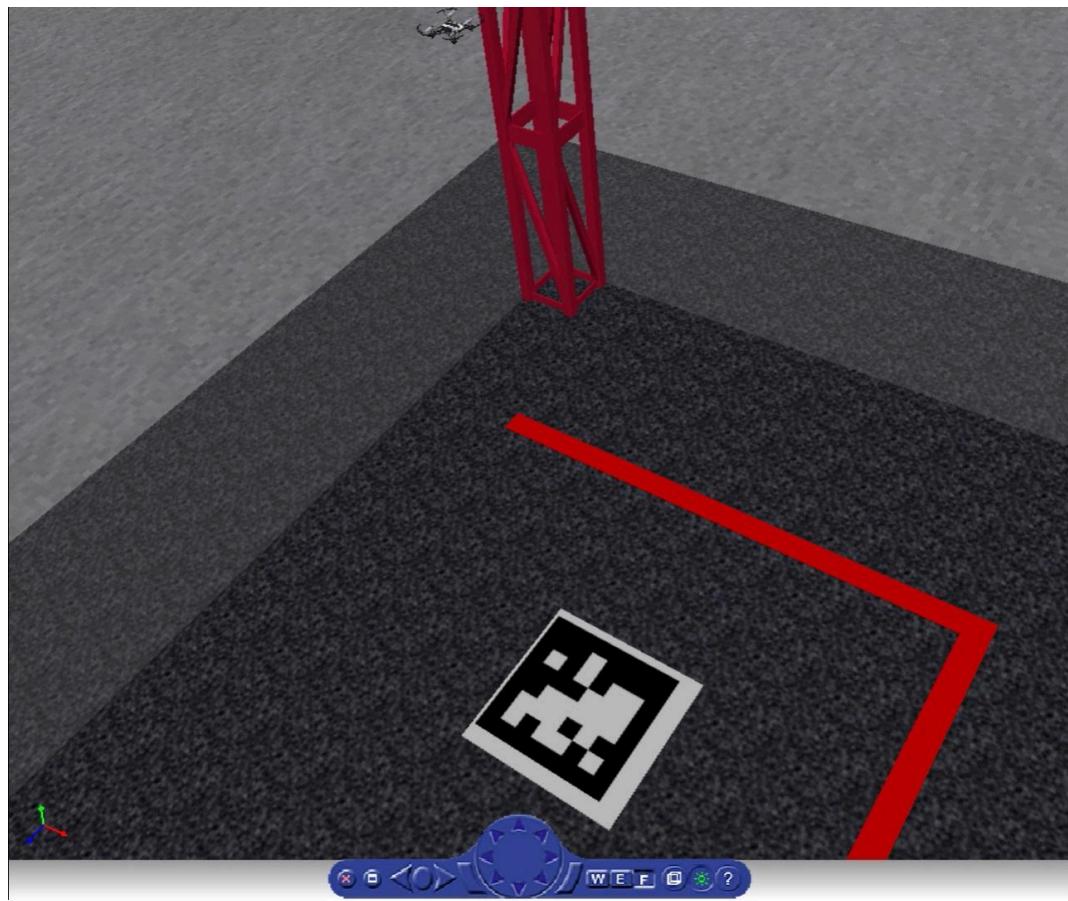


Fig 5.3: Alternative view of the Environment of simulation within the simulation
The top view of the environment model is depicted in the image below.

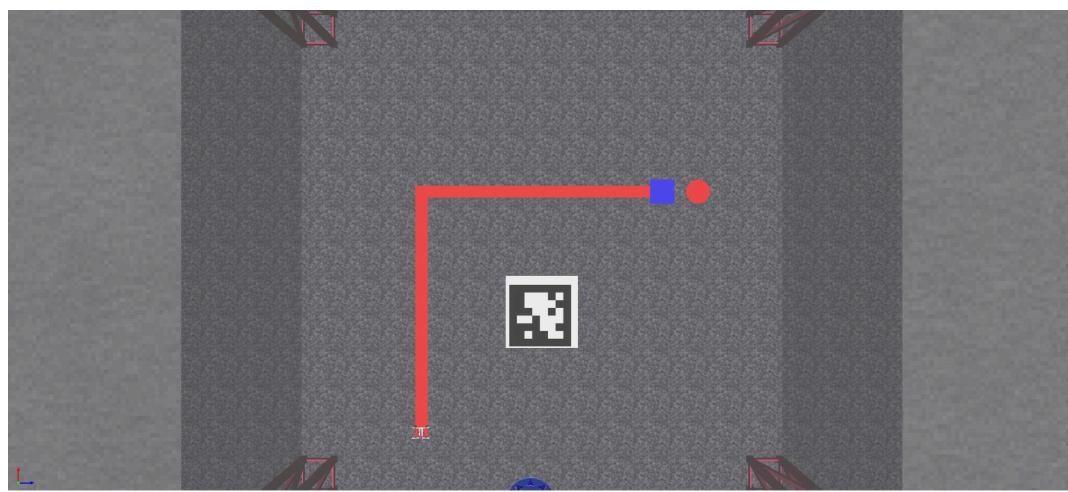


Fig 5.4: Top view of the environment with landing zone within the simulation
The effective landing is accomplished with the help of image processing and the landed view of the drone is depicted below.

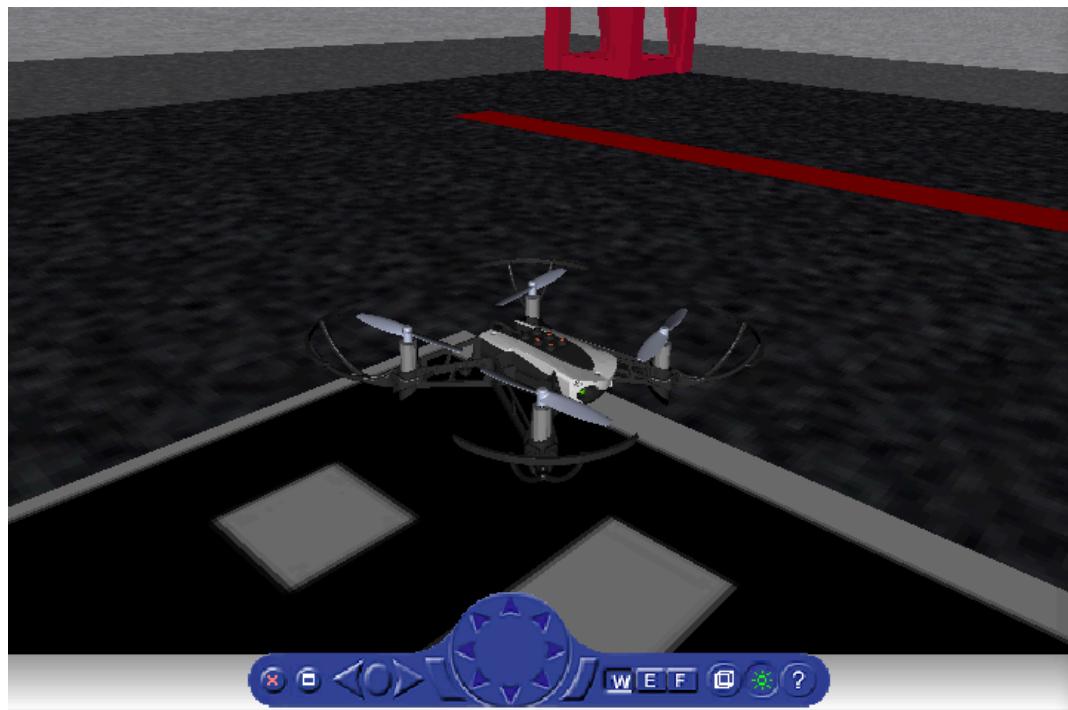


Fig 5.5: Isometric view of the drone after landing on the Apriltag platform

Drone states during the simulation time period is analysed based on the algorithm provided and the plot is depicted below.

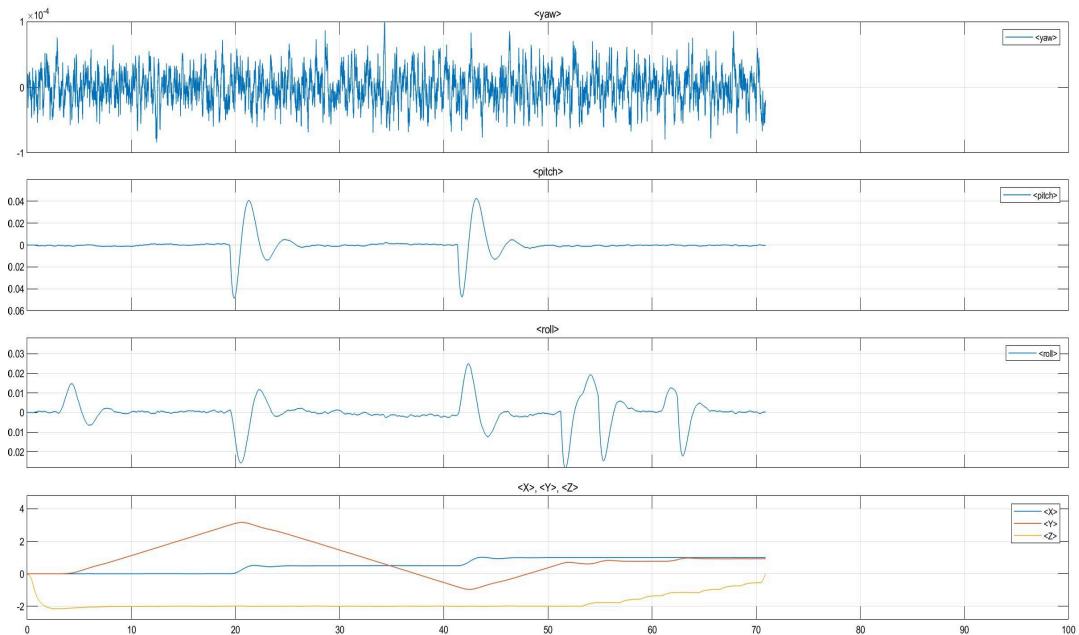


Fig 5.6: Graphical representation of Drone states during the simulation

6.CONCLUSION

Drones have found application in many fields like inspection, agriculture, search-and-rescue missions, wildlife monitoring, and consumer goods delivery. Although UAVs are becoming a pervasive sight in today's world, many of the applications require specialized human control or supervision. Thus drones with the capability of autonomous landing can augment the levels of application of these drones.

In this project, we intended to develop an optimized system for the autonomous landing of drones so initially as the primary objective we acquired a solid grip on the basics of drone dynamics. Next, an experiment was conducted with 8 prominent feature extraction methods to find the best-suited method for our application and concluded from the result data set on using the SURF method. In the next step, we integrate the best feature extraction method found from the above step into the drone navigation path for finding the platform. At last, the drone is step by step hovered down to the landing pad with a display of task completion.

The future scope of the project includes a similar landing on a dynamic platform in a realistic environment along with obstacle avoidance and other features.

BIBLIOGRAPHY

- [1] W. Hogpracha and S. Vongpradhip, "Recognition system for QR code on moving car," *2015 10th International Conference on Computer Science & Education (ICCSE)*, 2015, pp. 14-18, DOI: 10.1109/ICCSE.2015.7250210.
- [2] Zhang, H., Wang, G., Lei, Z., and Hwang, J.-N., "Eye in the Sky: Drone-Based Object Tracking and 3D Localization", *< i>arXiv e-prints</i>*, 2019.
- [3] T. Anezaki, K. Eimon, S. Tansuriyavong and Y. Yagi, "Development of a human-tracking robot using QR code recognition," *2011 17th Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, 2011, pp. 1-6, DOI: 10.1109/FCV.2011.5739699.
- [4] Saha, Debjoy & Udayagiri, Balaji & Agarwal, Parakh & Ghosh, Biswajit & Kumar, Somesh. (2019). Warehouse Management Using Real-Time QR-Code and Text Detection.
- [5] H. Zhang, C. Zhang, W. Yang and C. Chen, "Localization and navigation using QR code for mobile robot in an indoor environment," *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2015, pp. 2501-2506, DOI: 10.1109/ROBIO.2015.7419715.
- [6] Y. Feng, C. Zhang, S. Baek, S. Rawashdeh, and A. Mohammadi, "Autonomous Landing of a UAV on a Moving Platform Using Model Predictive Control," *Drones*, vol. 2, no. 4, p. 34, Oct. 2018.
- [7] E. Olson, "AprilTag: A robust and flexible visual fiducial system," *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3400-3407, DOI: 10.1109/ICRA.2011.5979561.
- [8] Roland Brockers, Sara Susca, David Zhu, Larry Matthies, "Fully self-contained vision-aided navigation and landing of a micro air vehicle

independent from external sensor inputs," Proc. SPIE 8387, Unmanned Systems Technology XIV, 83870Q (25 May 2012); <https://doi.org/10.1117/12.919278>

[9]Yang, S., Scherer, S.A. & Zell, A. An Onboard Monocular Vision System for Autonomous Takeoff, Hovering and Landing of a Micro Aerial Vehicle. *J Intell Robot Syst* 69, 499–515 (2013). <https://doi.org/10.1007/s10846-012-9749-7>

[10]P. Serra, R. Cunha, T. Hamel, D. Cabecinhas and C. Silvestre, "Landing of a Quadrotor on a Moving Target Using Dynamic Image-Based Visual Servo Control," in *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1524-1535, Dec. 2016, DOI: 10.1109/TRO.2016.2604495.

[11]T. Templeton, D. H. Shim, C. Geyer and S. S. Sastry, "Autonomous Vision-based Landing and Terrain Mapping Using an MPC-controlled Unmanned Rotorcraft," *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 1349-1356, DOI: 10.1109/ROBOT.2007.363172.

[12] D. Lee, T. Ryan and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 971-976, DOI: 10.1109/ICRA.2012.6224828.

[13]Mr Vivek A, Velayudhan A, Neeraj S, Nandakishore R Nair and Vishnu S, “*Analogy of Point Feature Extraction Techniques*”