Autonomous Navigation of Drone with Frontal Obstacle Avoidance

A Project Report Submitted

by

MEGHANA SRIPALLE

(200010028)

In Partial Fulfilment

of the Requirements for the award of the degree

BACHELOR OF TECHNOLOGY

Under the supervision of

Prof. Sudheer Siddapureddy, Prof. Ameer K Mulla

DEPARTMENT OF Computer Science Engineering

INDIAN INSTITUTE OF TECHNOLOGY DHARWAD

November 17, 2022

DECLARATION

I declare that

this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date : 12th November 2021
Place : Dharwad

(Meghana Sripalle)
(200010028@iitdh.ac.in)

ABSTRACT

One of the most challenging problems in the domain of unmanned autonomous vehicles (UAVs) is the designing of a robust real-time obstacle detection and avoidance system. For real-time applications, different works are based on stereo cameras due to their ability to create the 3D model of obstacles and estimate the depth.In this project, a method that mimics the human behavior of detecting the collision state of the approaching obstacles using monocular camera is implemented. The key of the proposed algorithm is to analyze the size changes of the detected feature points along with the expansion ratios of the areas of the convex hulls constructed using the detected feature points from consecutive frames. During the UAV motion, the detection algorithm estimates the changes in the size of the boundary area of the approaching obstacles. This is done by detecting the feature points of the obstacles, then extracting the obstacles that have the probability of getting close to the UAV followed by comparing the area ratio of the obstacle and the position of the UAV, the method decides if the detected obstacle may cause a collision. Finally, by estimating the obstacle 2D position in the image and combining with the tracked waypoints, the UAV executed the avoidance action. This detection algorithm did not prove to be effective enough since it detected distant walls as obstacles due to the existence of texture on the walls. Later, another algorithm is implemented partially. This algorithm focuses on detecting and avoiding obstacles while moving from a starting point to the goal point autonomously by using a single ultrasonic sensor which can detect the edges of the obstacle and thereby generate an avoidance maneuver.

Keywords: Autonomous navigation, obstacle detection, obstacle avoidance, ultrasonic sensor, monocular camera, size expansion

ACKNOWLEDGEMENTS

I would like to thank Prof. Ameer Mulla and Prof. Sudheer Siddapureddy for the supervision of this research and development project. Without their guidance, resourcesmand connections this project would not have been possible. I am also deeply grateful to my parents who have always encouraged me to work towards my endeavours and supported me though everything.

Contents

Chapter 1

Introduction

An Unmanned Aerial Vehicle (UAV), or a drone, is an aircraft, often a multirotor which is basically a rigid body with mechanically moveable blades. UAVs are designed for dangerous and complicated missions as the replacements of manned aircraft. Modern unmanned aerial vehicles (UAVs) strive for increased levels of autonomy and flight stabilisation due to advancements in technology as well as the variety and complexity of tasks. It is thought to be a challenging problem for autonomous UAVs to be able to recognise and avoid obstacles with a high level of accuracy. The challenge arises because UAVs are becoming smaller and lighter. Because of these characteristics, smaller UAVs are unable to transport bulky sensors like radar and laser. On-board cameras are light, have low power requirements and can provide rich information of the environment enabling them for the utilisation for obstacle detection. It has been observed that approaches which estimate the depth, 3D models of the obstacles and use stereo cameras to detect the size and distance of the obstacle from the UAV bear the disadvantage of being computationally expensive. The usage of monocular cameras could be a challenging issue but size expansion offers helpful data for locating the obstacles that are moving towards the UAV. The ability of the human visual system to accurately extract information from things that are moving toward them is seen from a bio-inspired perspective. The human visual system can detect an approaching item based on the increase in its size with either one or both eyes. In this project, a bio-inspired approach using a monocular camera is implemented to imitate the human behavior of obstacle detection and avoidance applied on UAVs. The approach consists of calculation of the change in the size of feature points in consecutive frames followed by the calculation

of the area ratio of the convex hulls constructed from these feature points in order to detect obstacles. After the obstacle detection, the avoidance algorithm follows. Since 3D models are not required here, we can infer that the algorithm is comparably computationally inexpensive. This implementation did not give satisfying results. A second approach using an ultrasonic sensor is implemented. In this approach, the front shape of the detected object and edges are approximated by rotating the drone.

Chapter 2

Related Work

2.1 Frontal Obstacle Avoidance

Add some more details about it

Frontal obstacle avoidance is the phenomenon that occurs when a UAV flies autonomously and avoids the obstacles in its path by deviating from its initial path to a new point.

2.2 Visual-Based Frontal Obstacle Avoidance

Vision is one of the most powerful and popular sensing method used for autonomous navigation of UAVs. Compared with other on-board sensing techniques, vision based approaches to navigation continue to demand a lot of attention from the mobile robot research community. This is largely due to its ability to provide detailed information about the environment. Monocular vision based systems avoid problems like high computational complexity of the avoidance algorithms and cost of sensors and are able to provide appropriate solution to the obstacle avoidance problem.

There are two fundamental groups of vision based obstacle avoidance techniques; those that compute the apparent motion called optical flow based techniques, and those that rely on the appearance of individual pixels for monocular vision based obstacle avoidance systems. We can see that in vision-based navigation systems, different approaches were presented to solve the problem of obstacle detection and avoidance.

Approaches such as [1, 5], built a 3D model of the obstacle in the environment. Other approaches calculate the depth (distance) of the obstacles, such as in [4]. All these approaches share the common the disadvantage of the high cost in the computational time. From the bio-inspired point of view, the human visual system has the ability to extract information correctly of the objects that are moving toward them [6]. In addition, Gibson illustrated the ability of the human visual system to identify the approaching of the objects related to the expansion of its size, by both eyes or even one eye [3].

In this project, we implement the paper "Obstacle Detection and Avoidance System Based on Monocular Camera and Size Expansion Algorithm for UAVs" [2].

Chapter 3

Approach

3.1 Visual-Based Approach

3.1.1 Detecting the features, descriptors and matching

The UAV may fly in unknown environments due to which the image captured may contain some noise and errors. In order to capture the features accurately and effectively even in such conditions, we use the SIFT detector algorithm from the OpenCV library. The SIFT detector algorithm is scale-invariant and rotation-invariant as well due to which it can localize and detect keypoints effectively. The features and descriptors in the two consecutive frames namely the previous frame and current frame are computed.

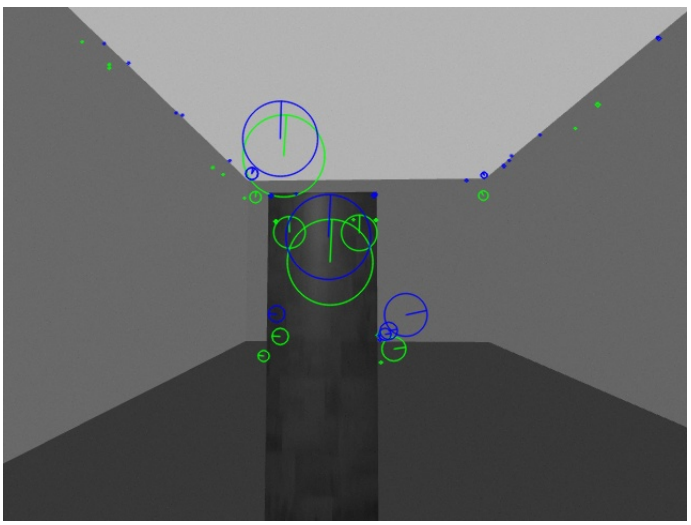


Figure 3.1: Blue represents the keypoints in the current frame, Green represents the keypoints in the previous frame

The position (x, y) and size (s) of the keypoint are stored in a vector. The features and descriptors from both the frames undergo matching using the Brute Force Matcher from the OpenCV library. The Brute Force Matcher uses the L2-Norm as the distance measurement. The k(which is 2 here) best matches of a keypoint are computed.

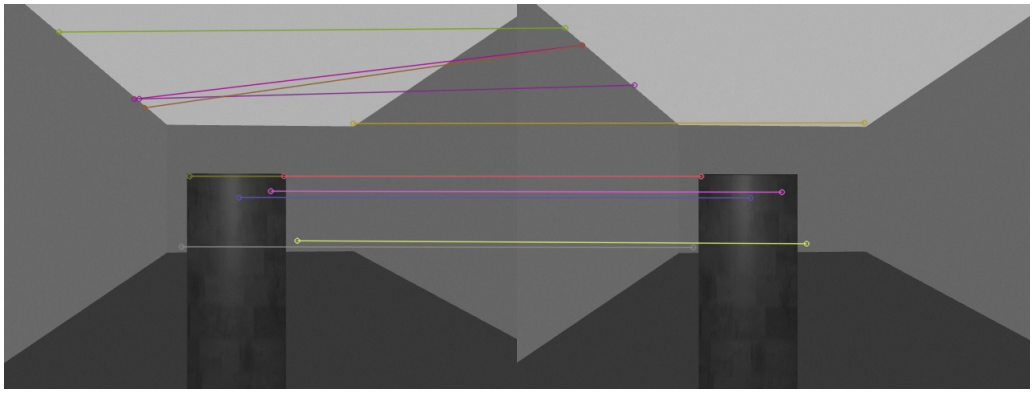


Figure 3.2: The matched keypoints in both the frames

The matched keypoints in the two consecutive frames then undergo size filtering. The size filtering is done so that the matched keypoints which are not growing with significantly with size are eliminated. The size filtering is done as follows where n denotes the number of matched keypoints: Add refrence

For $i \forall n$

$$mkp(i) = \begin{cases} (x, y, s) & size(mkp_t(i)) > size(mkp_{t-1}(i)) \\ 0 & x \leq 0 \end{cases}$$

Here, mkp_t is a keypoint in the current frame and mkp_{t-1} is a keypoint in the previous frame.

3.1.2 Computing the Object of Interest

The Object of Interest is created by creating a bounding box around the matched and filtered keypoints in the two consecutive frames. This bounding box is created by extracting the maximum and minimum x keypoint coordinates along with the maximum and minimum y keypoint coordinates.

The areas of the bounding boxes in both the frames are computed using the following equations where B_{area} is the bounding box area:

$$l = y_{max} - y_{min} \tag{3.1}$$

$$w = x_{max} - x_{min} \tag{3.2}$$

$$B_{area} = l * w \tag{3.3}$$

Now, the ratio of the sizes of matched and filtered keypoints in the current frame to the previous frame is calculated.

$$ratio(mkp) = \frac{1}{N}\left(\sum_{j=1}^{n}\left(\frac{size(mkp_t)}{size(mkp_{t_1})}\right)\right) \tag{3.4}$$

The ratio of the area of the bounding box in the current frame to the area of the bounding box in the previous area is computed.

$$ratio(B) = \frac{size(B_t)}{size(B_{t-1})} \tag{3.5}$$

The condition implemented for detecting the object as a frontal obstacle is as follows:

$$State = \begin{cases} 1 & ratio(mkp) \geq 1.2 \wedge ratio(B) \geq 1.35 \\ 0 & otherwise \end{cases} \tag{3.6}$$

The ratio thresholds for the size rato of keypoints and area ratio have been decided after perusing other research papers. Further different ratios have been experimented with. We found this threshold suitable to our requirements. The state becomes **1** only when the above condition is satisfied indicating that the collision with the obstacle is possible.

3.1.3 Drawbacks

We noticed that the bounding box for every frame was including the area of the walls in the field of view. We inferred that this was due to the existence of texture on the walls. The features on the walls were also being considered during keypoints detection and also during the matching of the keypoints. Also size expansion was occurring in the matched features of the surrounding walls. We concluded that keypoint detection and the size expansion technique for detection obstacle would not be accurate when there are walls surrounding the obstacle. Due to the texture on the walls, the exact obstacle is not being detected accurately.

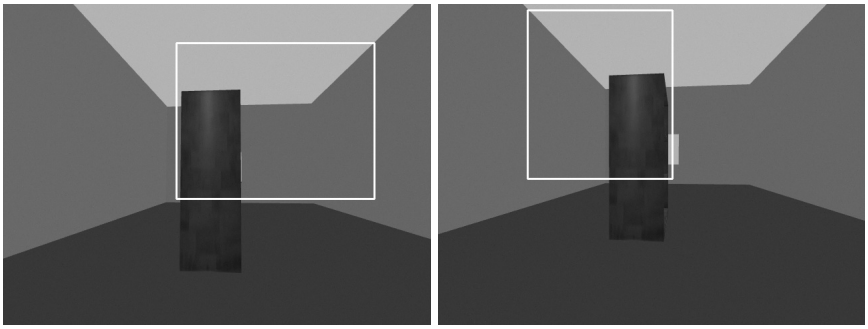


Figure 3.3: Images of the wall being included in the bounding box of the obstacle

3.2 Ultrasonic Sensor-Based Approach

A new approach using ultrasonic sensors is implemented. When ultrasonic sensors are used, the distance to the object is used as a deciding parameter for obstacle detection. If the walls are distant then they are not considered as obstacles. The approach utilises only one ultrasonic sensor and approximates the front shape of the detected object by sweeping the UAV. The UAV is rotated for shape approximation and edge detection of the entire object. The UAV is rotated so that the part of the object which is not is the field of view of the UAV is also captured. When the presence of an object is detected then, the following steps are executed: Obstacle detection,Speed Control, Edge Detection, Triangulation and Coordinates Determination and Decision. Until now, Obstacle detection,Speed Control, Edge Detection, Triangulation modules along with avoidance of three different types on obstacles have been implemented.

3.2.1 Obstacle Detection

During its flight, the UAV measures the distances of the objects in front of it with the help of the ultrasonic sensor. When the object in front of it is at a distance less than **1 meter**, the object is detected as an obstacle. This is followed by the **Speed Control module**.

Speed Control

This module is responsible for slowing down the UAV by halving the speed of the UAV. The UAV slows down as it approaches the obstacle. The UAV continue slowing down till it is at a distance below 0.2 meters from the detected obstacle. Once it reaches this

point, the **Edge Detection module** begins to be executed.

Edge Detection

This module is responsible for the rotation of the UAV in order to detect the distances of the left edge and right edge of the obstacle from the UAV. The UAV can be rotated 45°in the left and right directions. That is the limitation on the rotation of the UAV. First, the UAV is rotated to the left 1°at a time till the distance measured by the sensor crosses a certain threshold. This indicates that the left edge has been detected. The left edge flag is set to True. If the UAV has rotated left by 45°without a significant change in the frontal distance then the left edge is beyond the visibility range of the UAV.Then, the UAV is rotated back to its initial position. The UAV follows the same procedure to detect the right edge of the obstacle. After rotating right, the UAV rotates back to the initial position. Also, it has been noticed that there is a slight error in the rotation of the UAV which can be corrected with the utilisation of PID control. After setting the left edge and right edge flags, the **Triangulation module** is called.

3.2.2 Obstacle Avoidance

Triangulation

The distances to the left edge and right edge of the obstacle are used to compute the location of the edges if their corresponding edge flags evaluate to True. The angles at which the edges are detected are also computed and stored. The definitions of various variables used are below:

f_s = minimum safe distance allowed from the side of UAV

$h1$ and $h2$ = diagonal detected distance of the obstacle (edges) on the left and right sides respectively

$a1$ and $a2$ = distance of the object on the left and right side of the UAV respectively, calculated from the centre of the UAV

θ_1 and θ_2 = angles of the edges

d = width of the UAV

$a1$ and $a2$ are calculated as shown below:

$$a1 = h1 * \cos(90 - \theta_1) \tag{3.7}$$

$$a2 = h2 * \cos(90 - \theta_2) \tag{3.8}$$

The minimum safe distance allowed from the side of the UAV is initialised so that the UAV can decide if it has to reverse in case of obstacles like V-shaped obstacles where the side distance is equal to the minimum safe distance.

The UAV makes a decision taking into consideration the values of the left and right edge flags. The UAV executes the avoidance action differently when only the left edge is visible, only the right edge is visible or both the edges are visible.

Test case 1: When the left edge is visible

Here, the left edge flag is True while the right edge flag is False. The UAV makes the decision to turn left in this case. This turning is executed by rotating the UAV left by a certain angle (θ + θ_1) from its original position. After the rotation, it moves forward in order to avoid the obstacle. The angle θ is calculated as follows:

$$\sin\theta = \frac{d/2}{h1} \tag{3.9}$$

Test case 2: When the right edge is visible

Here, the left edge is not in the field of view of the UAV. The right edge flag is True while the left edge flag is False. The UAV makes the decision to turn right in this case. This turning is executed by rotating the UAV right by a certain angle (ϕ + θ_2) from its original position. After the rotation, it moves forward in order to avoid the obstacle. The angle ϕ is calculated as follows:

$$\sin\phi = \frac{d/2}{h2} \tag{3.10}$$

Test case 3: When both edges are visible

Here, both the edge flags are True i.e. the left and right edges are in the possible field of view of UAV after rotating 45°left and right. The minimum of $a1$ and $a2$ is calculated. If $a1$ is the minimum length then UAV turns left by rotating the UAV left by a certain angle (θ + θ_1) from its original position. The angle θ is calculated as we did in the test case for the visible left edge. If $a2$ is the minimum then the UAV turns right by rotating right by a certain angle (ϕ + θ_2) from its original position. The angle ϕ is calculated as above. After the rotation, it moves forward in order to avoid the obstacle.

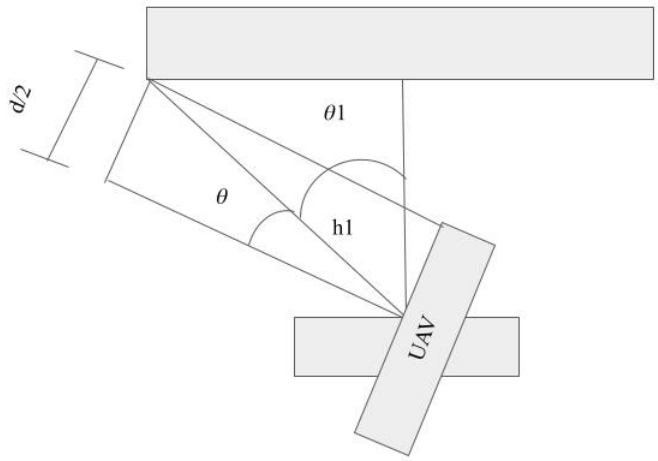


Figure 3.4: Calculation of the angle to rotate

Chapter 4

Results

We achieved negative results for the visual based approach. The obstacle detection is not accurate in this case. The sensor-based approach has not been implemented entirely. It is able to detect and avoid certain obstacles in some situations.

4.1 Results of the Visual-Based Approach

The negative results for the visual based approach are below. We can see that the obstacle is being detected but the walls are being detected as part of the obstacle as well. Any approach which uses keypoints to detect the obstacle will come across such results since texture on walls or surrounding objects is also detected as a feature. There will be size expansion in those features when we move towards the obstacle.

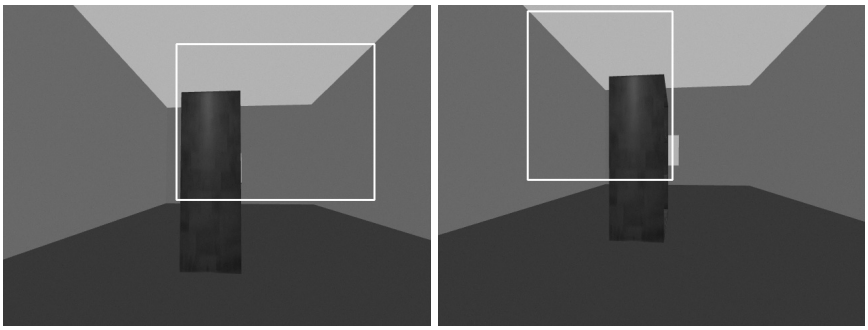


Figure 4.1: Results

4.2 Results of the Sensor-Based Approach

The output printed in the terminal and the movement of the UAV when both the edges of the obstacle are in the region of visibility of the UAV are below.

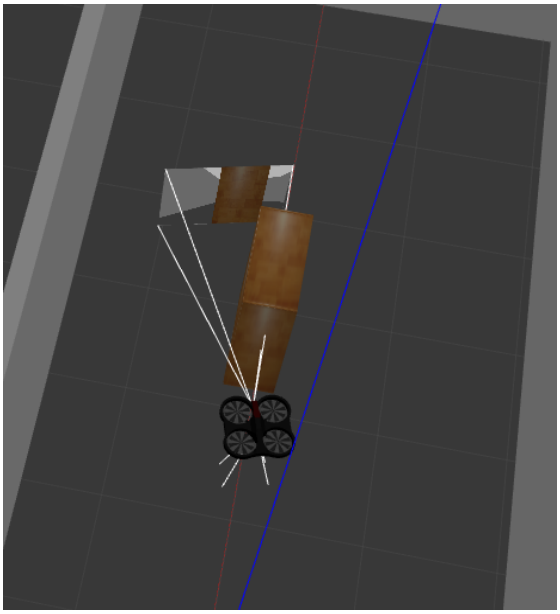
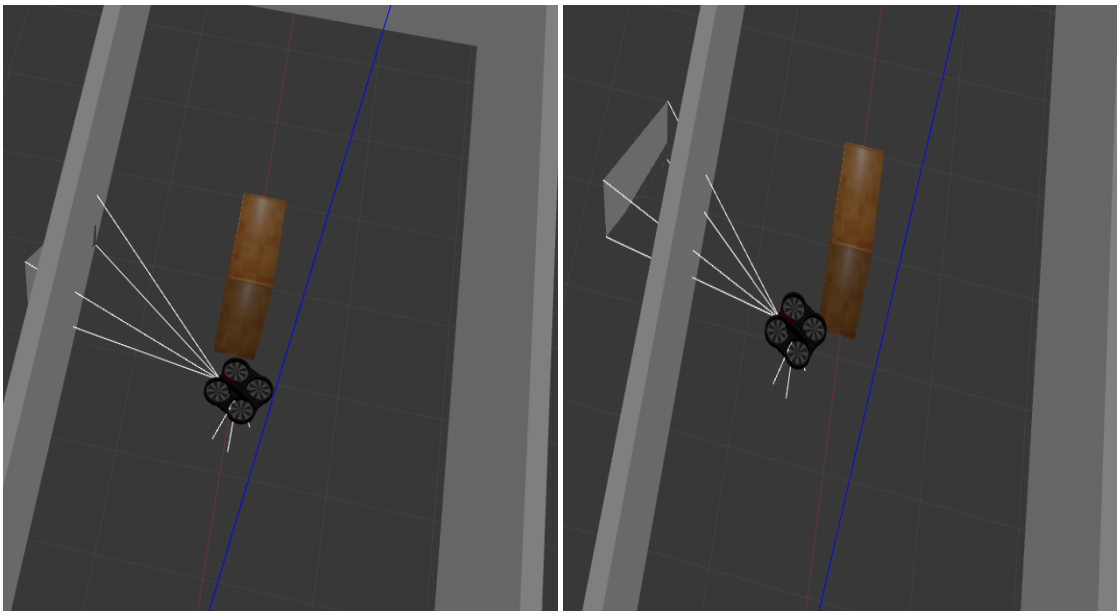


Figure 4.2: The UAV is detecting the edges



(a) The UAV rotates left (b) The UAV moves forward

Figure 4.3: Avoidance action when both the edges are visible

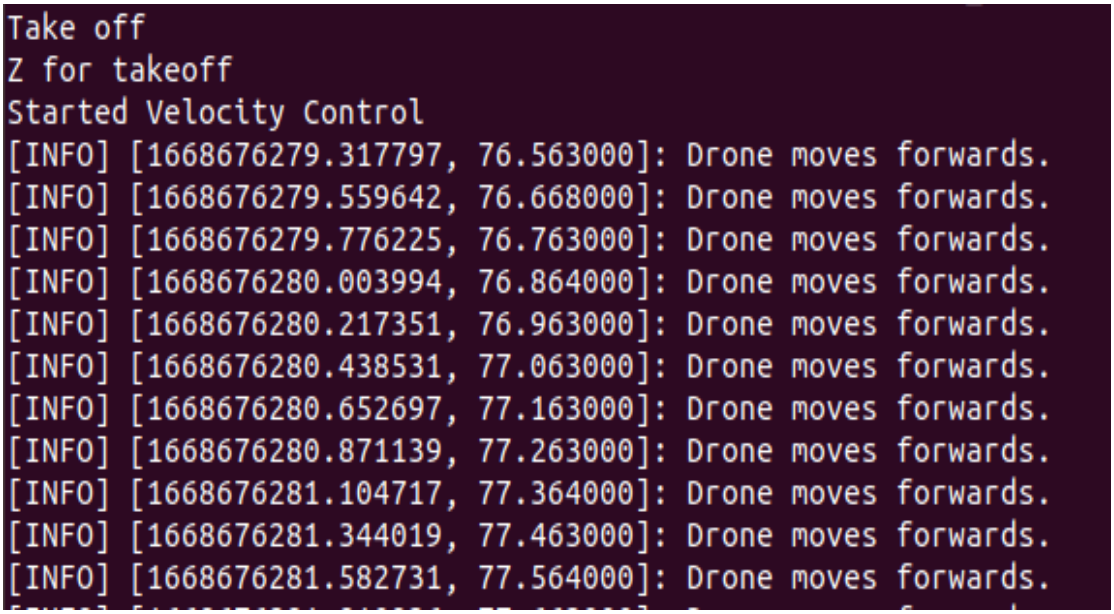


Figure 4.4: Commands generated in the terminal

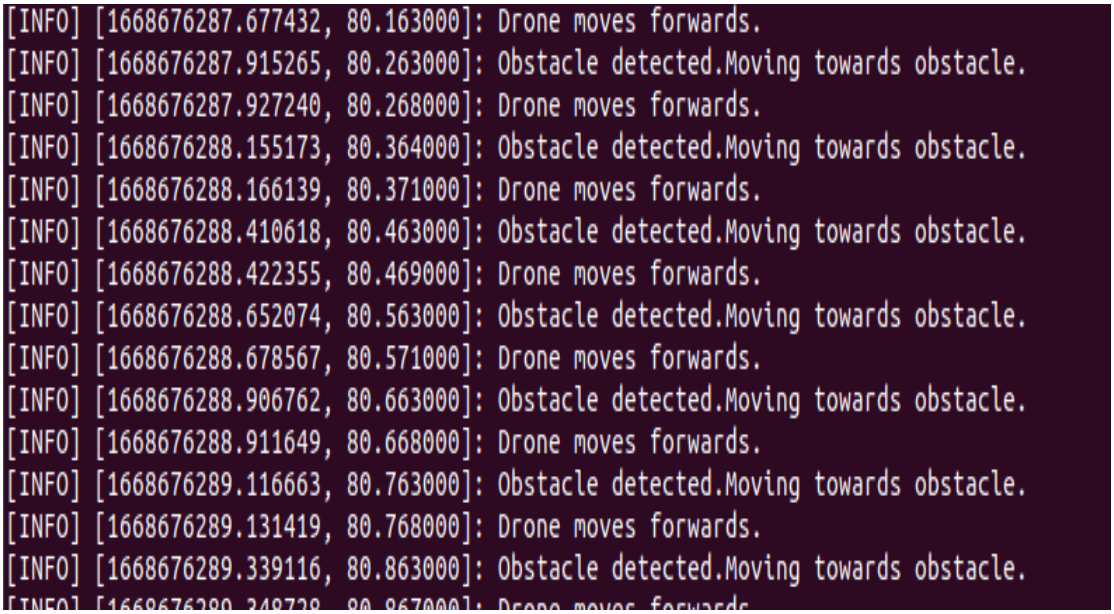


Figure 4.5: Moving towards the obstacle

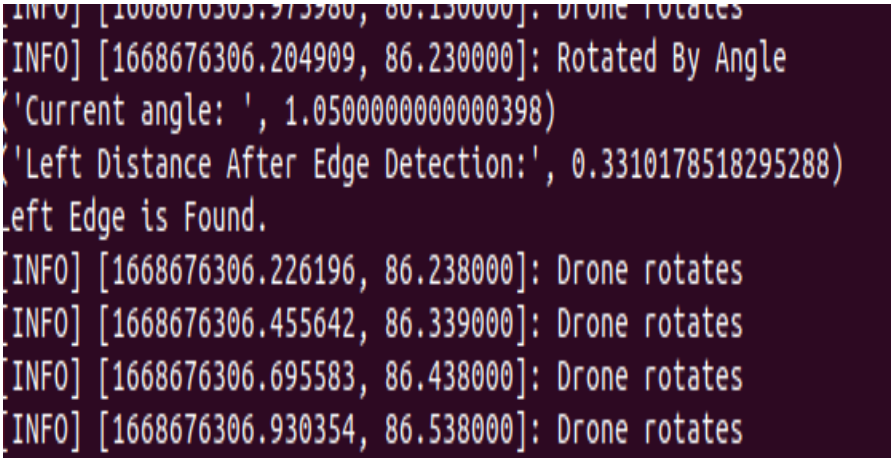


Figure 4.6: The left edge is found

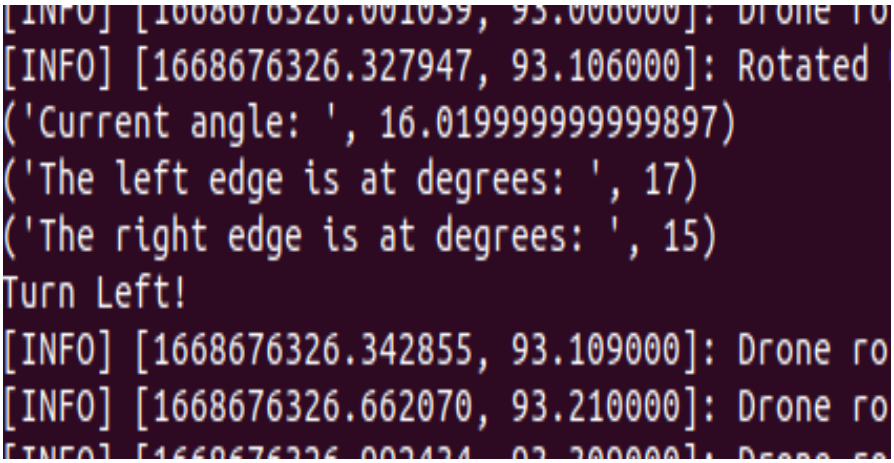


Figure 4.7: Turn Left Command

Can add a video showing both results Visual as well as distance based

Chapter 5

Conclusion

The first approach that was implemented involved using computer vision libraries like OpenCV to implement the size expansion technique for the detection of an obstacle. On implementation, we inferred that this method is not effective in the cases when the obstacle is surrounded by walls with texture. From hereon, we attempted to implement a sensor-based approach to effectively detect the obstacle and avoid it. The sensor detects the edges of the obstacle and the algorithm uses trigonometry to decide in which direction the UAV should move in order to avoid the obstacle. Future work of this project includes implementation of the avoidance maneuver of the UAV in case of obstacles which are V-shaped and obstacles whose edges are not visible even when the UAV is rotated for edge detection. Also, the rotation of the UAV has to be modified so that when it rotates by a particular angle, it does not rotate more than that angle. This can be done by implementing PID control for the rotation of the UAV.

Bibliography

[1] Broggi A et al. “A full-3D voxel-based dynamic obstacle detection for urban scenario using stereo vision”. In: *Proceedings of the 2013 16th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (2013), pp. 71–76.

[2] David Martín Abdulla Al-Kaff Fernando García, Arturo De La Escalera, and José María Armingol. “Obstacle Detection and Avoidance System Based on Monocular Camera and Size Expansion Algorithm for UAVs”. In: *Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC)* (2017). DOI: https://doi.org/10.3390%2Fs17051061.

[3] Gibson J.J. “The Ecological Approach to Visual Perception: Classic Edition”. In: *Psychology Press; Boston, MA, USA* (2014).

[4] Li X.-M Li J. “Vision-based navigation and obstacle detection for UAV”. In: *Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC)* (2011), pp. 1771–1774.

[5] Jeong H Na I. Han S.H. “Stereo-based road obstacle detection and tracking”. In: *Proceedings of the 2011 13th International Conference on Advanced Communication Technology (ICACT)* (2011), pp. 1181–1184.

[6] Yamaguchi M.K Shirai N. “Asymmetry in the perception of motion-in-depth”. In: (2004), pp. 1003–1011.