2023 International Conference on Unmanned Aircraft Systems (ICUAS)
June 6-9, 2023 | Lazarski University, Warsaw, Poland

2023 International Conference on Unmanned Aircraft Systems (ICUAS) | 979-8-3503-1037-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICUAS57906.2023.10156604

RHFSafeUAV: Real-Time Heuristic Framework for Safe Landing of UAVs in Dynamic Scenarios

Jaskirat Singh*[1], Neel Adwani*[1], Harikumar Kandath[2], and K. Madhava Krishna[2]

*Abstract***— This study presents a technique for multi-rotor unmanned aerial vehicles (UAVs) to efficiently and safely land in dynamic environments. The aim of this method is to locate a secure potential landing zone (PLZ) and choose the best one for landing. The PLZ is initially determined with an area estimation algorithm, which returns the empty region in the image where the UAV can possibly land. The obstacle-free regions that have a higher area than the vehicle's dimensions with tolerance are labeled as safe PLZs. In the second phase of this approach, the velocities of dynamic obstacles moving towards the PLZs are calculated, and their time to reach the zones is taken into consideration. The estimated time of arrival (ETA) of the UAV is calculated, and during the descent of the UAV, dynamic obstacle avoidance is executed. A ToF (Time of Flight) sensor is used for detecting altitude, while a depth camera is used for performing triangulation, area estimation, and computing distance to the target site. The approach, tested in real- world environments, has shown better results compared to existing work as the computation time is significantly lower, while the accuracy is competitive with deep learning counterparts.**

*Index Terms***— Safe Landing, Multi-rotor UAVs, Computer Vision, PLZ Detection, and Dynamic Environments**

I. INTRODUCTION

The development of autonomous Unmanned Aerial Vehicles (UAVs), which can self-navigate in a range of situations, has been aided by recent advances in artificial intelligence, control, and remote sensing technologies. The utilization of UAVs has increased significantly, showing applications in surveillance and security systems [1], [2], delivering of products [3], monitoring forest fires [4], motion and traffic analysis and various other research purposes [5], [6].

Many studies [7], [8], [9] focus on completely known and static environments, i.e environments that consist of objects at fixed locations. Given the application scenarios, there are missions where UAVs are susceptible to external events like loss of communication from ground stations, inability to navigate due to sensor malfunctioning, emergency landing during search and rescue operations, and weather disturbances. Hence, in the event of different scenarios, this could impact the operations, thus leading to the safe landing of UAVs. Most of the approaches have not considered highly dynamic environments where numerous obstacles are randomly moving at different speeds. However, these scenarios commonly exist in practice; for instance, many dynamic obstacles, such as humans and vehicles, may disturb UAVs during landing at a specific target location initially identified by the UAVs. Therefore, in these situations, UAVs must safely land at unstructured locations to reduce damage to themselves and avoid causing any injury to humans. To achieve the collision avoidance objective in different dynamic environments, it is essential for the UAV to track the position and velocity of the drifting obstacles, choose a region for landing that has enough area with respect to the size of the vehicle, and avoid colliding with any dynamic obstacles around. This is a spatio-temporal prediction-identification task. However, in most cases, during run-time, UAVs do not have prior information about the region of the potential landing area while aborting missions. Dynamic environments comprise various moving objects at ground level and in the air that change their positions with respect to time, considering the three different scenarios, viz., i) Urban, ii) Rural, and iii) Sub-urban. A considerable amount of research has been dedicated to detecting landing sites with statically positioned objects using various computer vision techniques [10], [11]. Although these methods operate well in some specific scenarios, it might be challenging to reliably and properly identify the landing region in more complicated situations, such as dynamic environments where objects are in continuous movement.

The main contributions of this paper are listed below:

1) Considering the challenges that UAVs encounter during the autonomous safe landing, we primarily have developed a framework to land safely in static and dynamic environments where any objects in motion are tracked. According to the author's knowledge, no research has been done that specifically addresses this issue in dynamic circumstances where the scenario changes with respect to time. The previous related works in [12], [13] have primarily focused on static scenarios.
2) Consequently, detecting a reliable potential landing zone (PLZ) is essential for safe operations. Therefore, we introduced the architecture given in Fig. 3 for finding the area of PLZ, and distance of the UAV to the PLZ. The existing literature estimates the resulting area of the potential landing zone using computationally intensive state-of-art deep learning networks. We have compared our results with existing literature showing higher accuracy and precision for area identification with a low computational cost.

[*] Denote equal contribution

[1]University of Petroleum and Energy Studies, Dehradun juskirat2000@gmail.com, neeltr.n@gmail.com
[2]International Institute of Information Technology, Hyderabad harikumar.k@iiit.ac.in, mkrishna@iiit.ac.in

3) Furthermore, we evaluated our proposed approach with field tests in real-world dynamic environments.
4) An open-source framework for the safe landing of UAVs in dynamic environments that can be embedded into any micro-processor for future work is developed.

This paper is organized as follows: Section II describes the related work for this study. The problem formulation is discussed in Section III. Section IV covers the proposed architecture for the autonomous safe landing of UAVs, PLZ detection, and real-time PLZ state estimation and navigation in dynamic scenarios. Experiments and real-time results are described in Section V. Section VI concludes this paper.

II. RELATED WORK

This section has been divided into two parts-Potential Landing Zone Detection, and Real-time Potential Landing Zone State Estimation and Navigation in Dynamic Scenarios. The overview is provided in Fig. 1.

A. Potential Landing Zone Detection

The safe landing of UAVs remains an open problem, especially in unknown environments [14], [15]. Recent research has been divided into two categories: a) Sensor-based detection systems [16], [17], [18], [19], and b) Vision-based detection systems [20], [21]. The authors in [22] created a density map for each image using a deep neural network and obtained a binary occupancy map aiming to overestimate people's locations. G. Castellano et al. [23] proposed a method for identifying safe landing zones using a lightweight, state-of-the-art CNN network. Meanwhile, Mukadam et al. [24] employed a more conventional SVM-based algorithm to detect potential landing zones by extracting features from colored satellite images.

B. Real-Time Potential Landing Zone State Estimation and Navigation in Dynamic Scenarios

The authors in [25] proposed a probabilistic graph approach and developed a cost function for a collision-free path tested in a simulation environment. C. Lyujie et al. [26] combined inexpensive sensors like binoculars and LiDAR for autonomous landing in hostile settings. Various approaches have been proposed in recent years, including geometric relations [27], [28], fuzzy logic [29], [30], and neural networks [31], [32]. The authors in [33] devised an algorithm for navigating through waypoints and avoiding obstacles using simplified geometry from a point cloud. Authors in [34] employed the time-obstacle dynamic map (TODM) to avoid dynamic obstacles.

In contrast to the aforementioned techniques summarized in Fig. 1, the approach presented in this paper detects the potential landing zones and, formulates the path for the UAV, by tracking the object velocities, and estimated time required for each object to reach the targeted location. Finally, we evaluate our framework in the real-world environment.

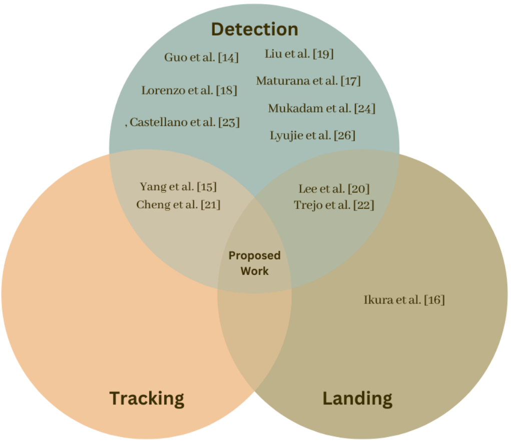
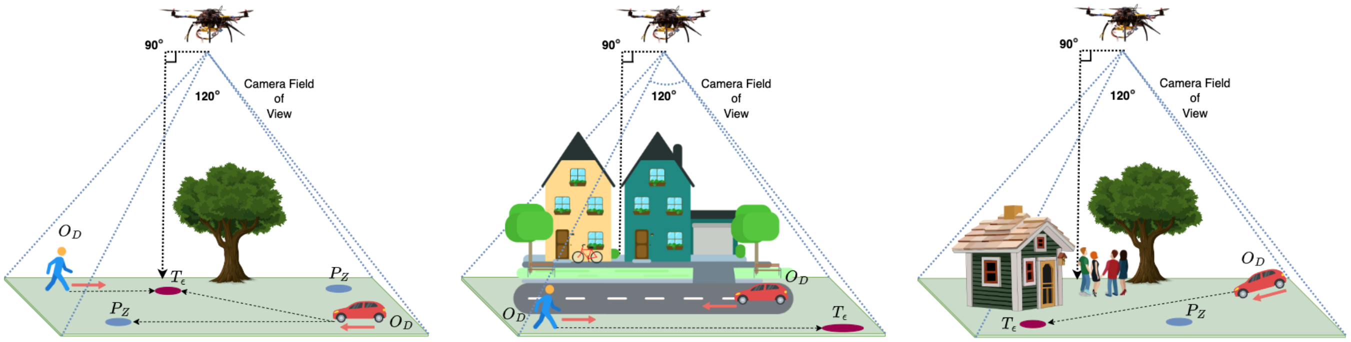


Fig. 1: A literature overview reveals existing work in detection, tracking, and landing categories, with our proposed research situated at the intersection of these domains.

III. PROBLEM FORMULATION

As illustrated in Fig.2, we consider three different heterogeneous scenarios for UAV navigation: Rural, Urban, and Sub-urban. The aim of this study is to determine a safe landing zone (PLZ) and a feasible path for the UAVs to reach the target spot while avoiding dynamic obstacles. The three environments are characterized by varying population densities, road structures, and the presence of moving objects such as vehicles, people, and other obstacles. (T_ϵ) and (P_Z) represent the target spot and identified PLZ, respectively, while (O_D) represents various dynamic obstacles moving at different velocities, which in practical environments include vehicles, humans, and other moving objects. We considered a situation where dynamic obstacles randomly move in a straight line within the environment at a prescribed constant velocity, and the goal of the UAV navigation problem is to find a potential landing zone and a feasible, collision-free path from the origin to the target through the cluttered, moving obstacles.

To gather observations on the surroundings, a variety of UAV sensors, such as visual cameras, radar, and ultrasonic rangefinders, are often utilized. Taking into account different use cases, we employ a depth camera and a time-of-flight sensor to enable the UAV to detect its surroundings by measuring the distance between the UAV and its surroundings and tracking various velocities in real-time within its frame. As demonstrated, the observation vectors at time (t) are composed of UAV distance-measuring readings from various directions as $\theta_t = [d_t^1, d_t^2,...., d_t^n]$, where $0 \leq d_t^i \leq$ m (m=20, in this case). If d_t^i is shorter, the current UAV is closer to the obstacles in this direction. In contrast, it is safer if d_t^i is larger. Therefore, taking into account the conceptual understanding, we can locate the PLZ and land safely in any conditions. a) Rural Scenario: In rural areas, there is a vast open space with few moving objects and no proper roads. We calculate the ETA by the UAV and the object that



(a) Rural Scenario (b) Urban Scenario (c) Sub-Urban Scenario

Fig. 2: Different types of scenarios considered as a part of which problem is defined.

collides in the line of the identified PLZ, which has been marked red in Fig. 2a. b) Urban Scenario: As depicted in Fig. 2b, it represents a city view with tall buildings and a few static objects [st_1, st_2, ..., st_n] such as vehicles situated at positions [τ_1, τ_2, ..., τ_n]. This case consists of well-defined roads and footpaths for vehicles and pedestrians, respectively. Since roads are defined, vehicles move in a straight line, and UAV identifies PLZ along the roadside only. Generally, in this case, the UAV is more exposed to pedestrians walking or standing on the side trails. c) Sub-urban Scenario: In the sub-urban scenario, we assume two different cases; in Case 1, three people are playing cricket, and the ball as an object is considered within the UAV's camera field of view. In Case 2, we assume a four-wheeled car moving in a straight line on a road. Here, the UAV must accurately estimate the speed and trajectory of the car and keep it within its FOV.

IV. Methodology

This section has been divided into system overview, potential landing zone detection, and state estimation and navigation.

A. System Overview

The overview of the proposed system is presented in Fig. 3. During on-flight the images are captured at 30 FPS with the help of ZED[1] depth stereo camera where each frame is given as an input to the Canny Edge detection algorithm [35]. Diameter-Area estimation algorithm as described in Section IV-B is applied on the Canny Edge binary output image which forms the different possible circles on finding the empty spot, having no edges and the circle having the minimum 3 meters sq. area is referred as Potential landing Zone (PLZ). At the same time, within the input image, moving objects are detected using the color change in pixels. For every moving object, its distance to PLZ, and velocity are calculated and stored. Time taken by the object (T_x), and UAV to reach to nearest PLZ, is taken into consideration. The estimated time of arrival (δ) of the UAV to reach that PLZ spot is calculated by translating the pixels into the distance in X-axis, and Y-axis, and considering the altitude in Z-axis. We put on the deciding factor, considering if (T_x - δ) is greater than θ seconds, where θ = 20 (within this experiment), that denotes the specific PLZ is cleared for landing. The auto-land command is initiated as a part of autopilot mode, the UAV moves to the position in X-axis, and Y-axis. At the time of descent in Z-axis, the real-time obstacle avoidance algorithm, divides the present frame into four quadrants that makes an occupancy grid map [36], where every quadrant contains the depth matrix of every pixel. The average depth is taken from the matrix from each quadrant and is stored in an array. The quadrant with the highest depth is considered as the emptiest quadrant and the UAV would descend towards that quadrant. The on-board Time-of-Flight (ToF) based range measuring sensor[2], verifies the height of the UAV, along with the average depth from the depth camera. If the distance doesn't match, it means that there's an obstacle beyond it, which is the case when UAV will move to a different location, re-adjusts itself, and descend further till landing.

B. Potential Landing Zone Detection

As shown in Fig. 4, we divided our method into two parts to locate the PLZ. In *Stage I*, just after the UAV takes off, images are captured with the help of a stereo-depth camera integrated into the UAV at 30 FPS. As part of continuous evaluation, the frames are stored in the secure digital (SD) card embedded in a Jetson Nano. The color image is first converted to a grayscale image, and then the grayscale image is subjected to the Canny Edge detection method using (ρ) and (ν) as the lower and upper thresholds, respectively. The image's gradient is used by Canny Edge detection to locate the edges. The gradient of the image is determined using a Gaussian filter's derivative. The binary output image produced by the Canny Edge detection has edge pixels designated by 1 and non-edge pixels denoted by 0.

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(\frac{-x^2}{2\sigma^2}\right) \tag{1}$$

[1] ZED camera specifications and details can be found at the official Stereolabs website: https://www.stereolabs.com/

[2] The ToF sensor can be found at: https://www.terabee.com/shop/lidar-tof-range-finders/teraranger-evo-60m/

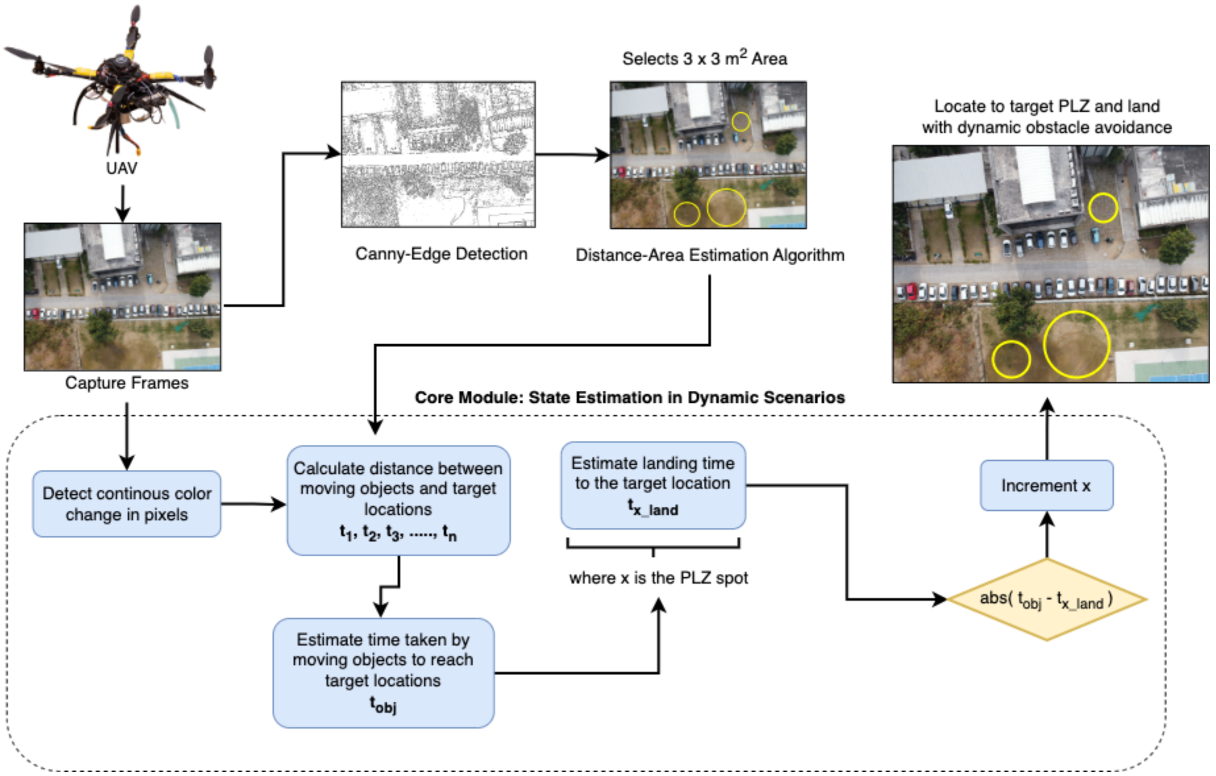

UAV
Capture Frames
Canny-Edge Detection
Selects 3 x 3 m^2 Area
Distance-Area Estimation Algorithm
Locate to target PLZ and land
with dynamic obstacle avoidance
Core Module: State Estimation in Dynamic Scenarios
Detect continous color change in pixels
Calculate distance between moving objects and target locations
t_1, t_2, t_3, ..., t_n
Estimate landing time to the target location
t_{x_land}
where x is the PLZ spot
Increment x
abs(t_{obj} - t_{x_land})
Estimate time taken by moving objects to reach target locations
t_{obj}

Fig. 3: Architecture of the *Safe Landing of UAVs*. UAV capture frames which are input for the *Area Estimation* and *Core Module* that initiates the auto landing.

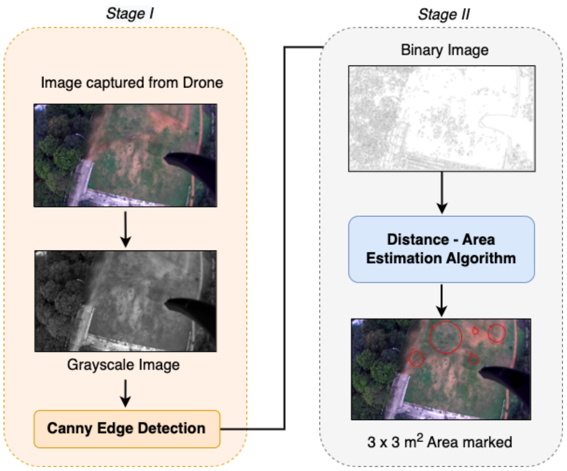


Fig. 4: Architecture of potential landing zone detection.

$$g_\sigma(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-y^2}{2\sigma^2}\right) \tag{2}$$

Eqs. 1, 2 show the response of the Gaussian filter in a single dimension.

$$G_\sigma(x, y) = g_\sigma(x) \cdot g_\sigma(y) \tag{3}$$

Eq. 3 is a 2-dimensional product of Eqs. 1, 2.

In *Stage II*, we apply the diameter-area estimation algorithm. Clustering is done by applying a Euclidean distance between the contours. Contours having a distance of less than 30 pixels are clustered into one set and a polygon is formed. The shortest distance between two different polygons is calculated using Eq. 4 which gives the distance between the two edges ($\delta(\mu, \psi)$), where μ and ψ represents different sets of the coordinates of edges.

$$\delta(\mu, \psi) = \sqrt{\sum_{i=1}^{n} (\psi_i - \mu_i)^2} \tag{4}$$

Hence, the distance between the two objects in real-world (D_o), can be found using the Eq. 5 where (υ) is the height of the UAV from the surface, (Θ) is the distance between two objects in image, and (f) is the focal length of the camera. Using Eqs. 4, 5, we can calculate the non-edge area as shown in the Eq. 6

$$D_o = \frac{\Theta \times \upsilon}{f} \tag{5}$$

$$A_{PLZ} = \pi \cdot \frac{D_o}{2} \cdot \frac{D_o}{2} \tag{6}$$

C. Real-Time Potential Landing Zone State Estimation and Navigation in Dynamic Scenarios

The approach for dynamic scenarios is divided into two stages. *Stage I* is a **Decision-Making** process where images are captured from the UAV, and each frame is evaluated to consider the dynamic motions of objects. To understand the scenario more precisely, moving objects (O_1, O_2, O_3, O_4,....., O_n) are detected using color changes in pixels. PLZs are represented by (t_1, t_2, t_3, t_4,....., t_n). For every PLZ (t_x), the distance from each object ($S_{1,X}$, $S_{2,X}$, $S_{3,X}$, $S_{4,X}$,....., $S_{n,X}$) located on the same axis is calculated using Eq. 4. The velocity of each object ($V_{1,x}$, $V_{2,x}$, $V_{3,x}$, $V_{4,x}$,....., $V_{n,x}$) is calculated based on the number of pixels shifting per second as shown in Eq. 7. The time taken (T_x) by the moving object (O_m) to reach its nearest PLZ (t_s) is calculated using Eq. 8, where ($S_{o,x}$) is the distance between the object and target location, and ($V_{o,x}$) is the velocity of an object to reach the target location.

$$V_a = \frac{V_p \times \upsilon}{f} \tag{7}$$

$$T_x = \frac{S_{o,X}}{V_{o,x}} \tag{8}$$

The ETA of the UAV (T_d) to the target PLZ is calculated by translating the pixels into distances using the Eq. 4 in X-axis, and Y-axis, and with the help of ToF sensor altitude is calculated in Z-axis. If the absolute value of (T_x - T_d) is greater than (α) seconds, that indicates that the specific PLZ (t_x) is cleared for landing, else the UAV slows down the speed in order to get the clear landing, while waiting for the moving objects to pass through the target location.

The *Stage II* is **Navigation**. Generally, there are 4 degrees of freedom to control a UAV: pitch, roll, yaw, and altitude. After the UAV makes the desired decision to land at a specific PLZ, our navigation algorithm allows the UAV to be easily controlled by providing them with the desired navigation position in the X-axis and Y-axis such that the target PLZ accounts in the center position of the frame while achieving the desired UAV velocity using the modern autopilots system, such as Pixhawk[3]. Thereby, for simplicity, the inner loop of the control is ignored and system concentrates on computing the desired positioning of the UAV. The UAV action vectors are defined using yaw angles in various directions as $\lambda_t = \left[\Delta\beta_t^1, \Delta\beta_t^2, ..., \Delta\beta_t^i\right]$, where $0 \leq \Delta\beta_t^i \leq 2\pi$. The UAV navigation can be described as

$$\delta(\psi) = \begin{cases} x_{t+1} = x_t + \nu \cos(\Delta\beta_{t+1}^i) \\ y_{t+1} = y_t + \nu \sin(\Delta\beta_{t+1}^i) \end{cases} \tag{9}$$

where $\rho_0 = [x_t, y_t]$ is the UAV position in Cartesian coordinate system and (ν) is the maximum speed. Generally, the navigation direction of a UAV cannot be abruptly changed; therefore, $\Delta\beta = \left|\Delta\beta_{t+1}^i - \Delta\beta_t^j\right| \leq \frac{\pi}{4}$, where to i and j represents the (i^{th}) and (j^{th}) actions in (λ_{t+1}) and (λ_t), respectively. Thus, the objective of the UAV navigation in dynamic scenario can be formulated as

$$\begin{cases} \min \sum_{j=0}^{T} ||P_{j+1} - P_j||_2 \\ s.t. d_{min} > \bar{d} \\ \rho_0 = [x_0, y_0]\,, \rho_T = [x_t, y_t] \end{cases} \tag{10}$$

where ($\bar{d}$) is the distance threshold for sustaining with the security, (ρ_0), (ρ_T) represents the initial and end positions respectively. When the landing of the UAV is initiated, i.e., at the time of descent, the real-time obstacle avoidance algorithm is activated, dividing each frame taken from the depth camera embedded in the UAV into four quadrants and creating an occupancy map. The occupancy grid map represents the area as a uniformly spaced grid of binary random variables, each indicating the presence or absence of an obstruction within that cell. Occupancy grid methods generate approximate posterior estimates for these random variables. Every quadrant of the frame contains the depth matrix of every pixel. When the distance of the UAV to the target is large enough compared to the views of two different cameras, then the angle (θ = ϕ), the distance (D_1 = D_2), and the distance between the depth camera and the target (L_1 = L_2). Hence, using Eq. 11, we can find the depth of a specific pixel in a quadrant, where H represents the altitude of the UAV from the plane surface, and D represents the depth of the pixel.

$$D = \frac{H}{\cos\theta} \tag{11}$$

The average depth taken from the matrix from each quadrant is stored into an array. This generally denotes that the quadrant having the highest depth is considered as the quadrant having no obstacle, indicating UAV to descend towards that specific quadrant. The on-board ToF sensor is used to verify the altitude of the UAV, along with the depth from depth camera. If the distance doesn't match, it means that there's an obstacle beyond it, which is the case when UAV will move to a different location and re-adjust itself and descend further.

V. EXPERIMENTS AND RESULTS

This section presents the extensive experiments conducted in real-world environments, and the results have been evaluated for various use case scenarios. The proposed framework is tested on an S-500 model quadrotor for the safe landing of the UAV, as shown in Fig. 5. The quadrotor is controlled by an onboard Jetson Nano and a Pixhawk PX4 flight controller, with depth images and altitude range measurements obtained from the depth camera and ToF sensor, respectively.

A. Potential Landing Zone Detection

We validated our PLZ detection module in real-world scenarios. Specifically, we tested the module in Rural, Urban, and Sub-urban scenarios, as shown in Fig. 6a. The yellow circles correspond to the identified PLZs, and their resulting diameters (distances) along with the area occupied can be visualized in Table I. After identifying the area, the algorithm

[3] Pixhawk product details can be found at the official Pixhawk website: https://pixhawk.org/

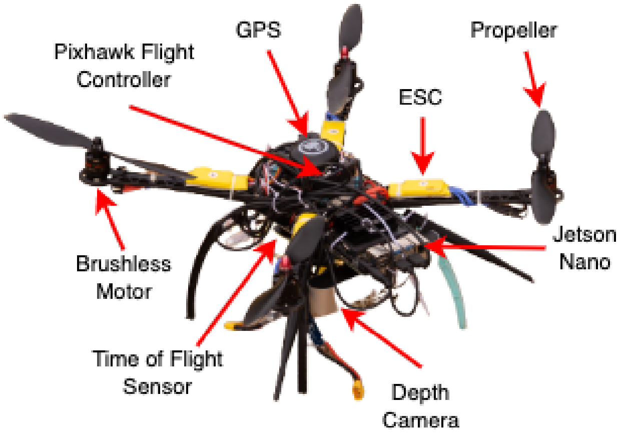


Fig. 5: Quadrotor UAV used in experiments

only considers areas with a value greater than 3 sq. m. We use a Time-of-Flight (ToF) sensor for obtaining the ground truth, which has a maximum range of 60 meters. The average percentage error for our distance and area-estimation method is recorded as 0.9692% and 1.9427%, respectively. To the best of our knowledge, we didn't find any work that calculates the resulting area for PLZ with UAVs without state-of-the-art deep learning networks. We also compared the accuracy to that of Google Earth, which has been listed as ⩽1% in [37]. It should be noted that, due to the lack of 3D imagery, some diameters used for identifying the area cannot be measured using Google Earth. We also compared our area-estimation resulting average percentage error with the work done by authors that claims for ⩽5% error in [38].

B. Real-Time Potential Landing Zone State Estimation and Navigation in Dynamic Scenarios

We studied different scenarios, as shown in Fig. 7, identifying various moving objects in the camera's field of view. We validated our algorithm, where we determined the speeds of the moving objects (vehicles), as shown in Table II. The ground truth was measured through the vehicles' odometry. Hence, the results yielded an average percentage error of 2.37%. To the best of our knowledge, we haven't found any work determining speeds using UAVs in dynamic scenarios. Fig. 8 shows the 3D planned trajectory of the UAV during landing, while Fig. 9 is a graph between the distance from the nearest object and time, which shows the reduction in the distance between the obstacle and the UAV as it descends in all three different scenarios. Through this framework, we were able to reduce computation time by 50% by eliminating any deep learning-based technologies. Additional details and code can be accessed at `https://github.com/jaskiratsingh2000/Research-UAVs-Safe-Landings`

VI. Conclusion

In this study, a large number of strategies that have been devised to ensure the safe landing of UAVs in dynamic environments while taking into account moving objects were put into practise. Safe landing of UAVs involves estimation of different safe potential landing zones. Hence, it is important to estimate the parameters that could identify safe PLZ, and navigate to land there. In particular, we identify PLZ using Canny Edge detection, which is further used to calculate the required area using nearest neighboring contours. Using diameter-area calculation, we did real-time state estimation and navigation in dynamic scenarios. We evaluated all our results on the real-world data, aiming at better approaches to land safely in dynamic scenarios. Future work includes considering the estimating the trajectories of the moving objects, and considering the run-time evaluation while UAV itself is in dynamic conditions.

VII. Acknowledgement

The authors acknowledge the financial support provided by IHUB IIIT Hyderabad and the Faculty SEED fund IIIT Hyderabad for carrying out this research.

References

[1] M. Bhaskaranand and J. D. Gibson, "Low-complexity video encoding for uav reconnaissance and surveillance," in *2011 - MILCOM 2011 Military Communications Conference*, 2011, pp. 1633–1638.

[2] P. Doherty and P. Rudol, "A uav search and rescue scenario with human body detection and geolocalization," in *AI 2007: Advances in Artificial Intelligence*, M. A. Orgun and J. Thornton, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 1–13.

[3] M. Haque, M. Muhammad, D. Swarnaker, and M. Arifuzzaman, "Autonomous quadcopter for product home delivery," in *2014 International Conference on Electrical Engineering and Information & Communication Technology*, 2014, pp. 1–5.

[4] K. Harikumar, J. Senthilnath, and S. Sundaram, "Multi-uav oxyrrhis marina-inspired search and dynamic formation control for forest firefighting," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 863–873, 2019.

[5] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[6] B. Pareek, P. Gupta, G. Singal, and R. Kushwaha, "Person identification using autonomous drone through resource constraint devices," in *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 2019, pp. 124–129.

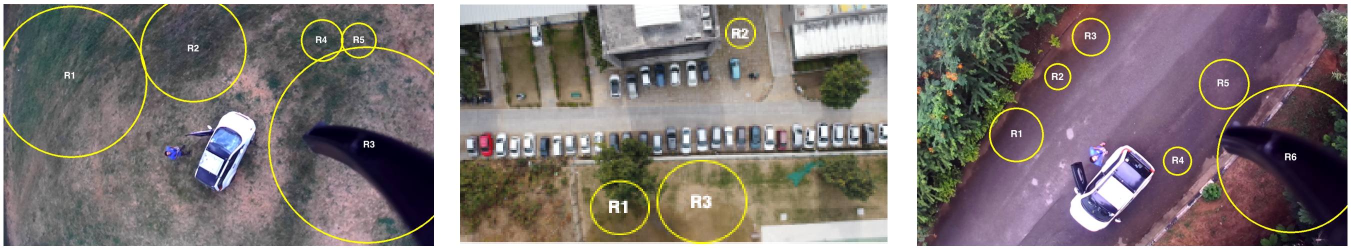
[7] P. Yao, H. Wang, and Z. Su, "Uav feasible path planning based on disturbed fluid and trajectory propagation," *Chinese Journal of Aeronautics*, vol. 28, no. 4, pp. 1163–1177, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S100093611500120X

[8] Y. LIU, X. ZHANG, Y. ZHANG, and X. GUAN, "Collision free 4d path planning for multiple uavs based on spatial refined voting mechanism and pso approach," *Chinese Journal of Aeronautics*, vol. 32, no. 6, pp. 1504–1519, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1000936119301542

[9] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, 2014, modelling of Mechanical and Mechatronic Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187770581403149X

[10] D. Safadinho, J. Ramos, R. Ribeiro, V. Filipe, J. Barroso, and A. Pereira, "Uav landing using computer vision techniques for human detection," *Sensors*, vol. 20, no. 3, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/3/613

[11] P. Eendebak, A. W. M. van Eekeren, and R. J. M. den Hollander, "Landing spot selection for uav emergency landing," in *Defense, Security, and Sensing*, 2013.



(a) Rural Scenario

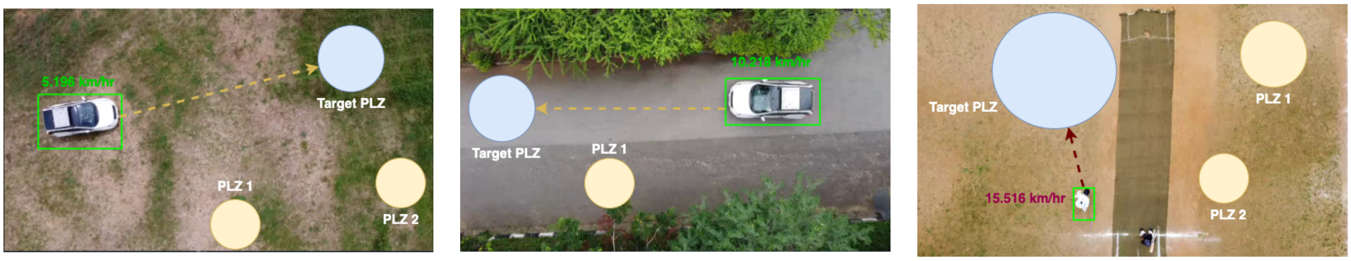
(b) Urban Scenario

(c) Sub-Urban Scenario

Fig. 6: Real-time distance-area estimation through flying UAV for the PLZ in different scenarios

Scenarios	Reference	Estimated Distance (m)	Ground Truth Distance (m)	Distance Error (%)	Estimated Area (sq. m)	Ground Truth Area (sq. m)	Area Error (%)
Rural	R1 in Fig. 6a	3.8976	3.9165	0.4826	11.9312	12.0472	0.9629
	R2 in Fig. 6a	2.7144	2.71	0.1624	5.7868	5.768	0.3259
	R3 in Fig. 6a	5.0808	5.125	0.8624	20.2747	20.629	1.7175
	R4 in Fig. 6a	1.0674	1.045	2.1435	0.8948	0.8577	4.3255
	R5 in Fig. 6a	0.9046	0.907	0.2646	0.6427	0.6461	0.5262
Urban	R1 in Fig. 6b	3.065	3.078	0.4224	7.3782	7.4409	0.8426
	R2 in Fig. 6b	0.7492	0.743	0.8345	0.4408	0.4336	1.6605
	R3 in Fig. 6b	5.33	5.286	0.8324	22.3123	21.9454	1.6719
Sub-Urban	R1 in Fig. 6c	0.7618	0.753	1.1687	0.4558	0.4453	2.358
	R2 in Fig. 6c	0.3714	0.3745	0.8278	0.1083	0.1102	1.7241
	R3 in Fig. 6c	0.532	0.522	1.9157	0.2223	0.214	3.8785
	R4 in Fig. 6c	0.4	0.4112	2.7237	0.1257	0.1328	5.3464
	R5 in Fig. 6c	0.6952	0.7001	0.6999	0.3796	0.385	1.4026
	R6 in Fig. 6c	2.0952	2.1	0.2286	3.4478	3.4636	0.4562

TABLE I: Landing area calculated for different PLZ using Our Method and Ground Truth for Rural, Urban, and Sub-Urban Scenarios through UAV.



(a) Rural Scenario

(b) Urban Scenario

(c) Sub-Urban Scenario

Fig. 7: Real-time velocity estimation through flying UAV, estimating time of arrival to reach the target PLZ in different scenarios

Scenarios	Velocity Measured using Accelerometer (Km/hr)	Estimated Velocity (Km/hr)	Absolute Difference	Percentage Error (%)
Scene 1	5	5.196	0.196	3.92
Scene 2	10.2	10.218	0.018	0.17
Scene 3	16	15.516	0.484	3.025

TABLE II: Velocity Estimation through translating the pixels. Ground truth has been measured through Accelerometer

[12] "An automatic zone detection system for safe landing of uavs," *Expert Systems with Applications*, vol. 122, pp. 319–333, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417419300272

[13] J. Guerin, K. Delmas, and J. Guiochet, "Evaluation of runtime monitoring for uav emergency landing," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9703–9709.

[14] X. Guo, S. Denman, C. Fookes, and S. Sridharan, "A robust uav landing site detection system using mid-level discriminative patches," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2016, pp. 1659–1664.

[15] T. Yang, P. Li, H. Zhang, J. Li, and Z. Li, "Monocular vision slam-based uav autonomous landing in emergencies and unknown

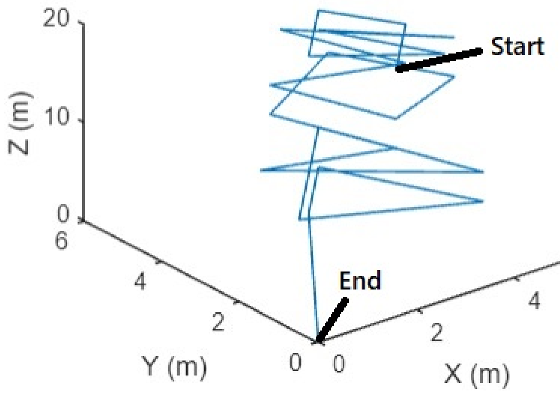


Fig. 8: 3D Planned Trajectory of the UAV

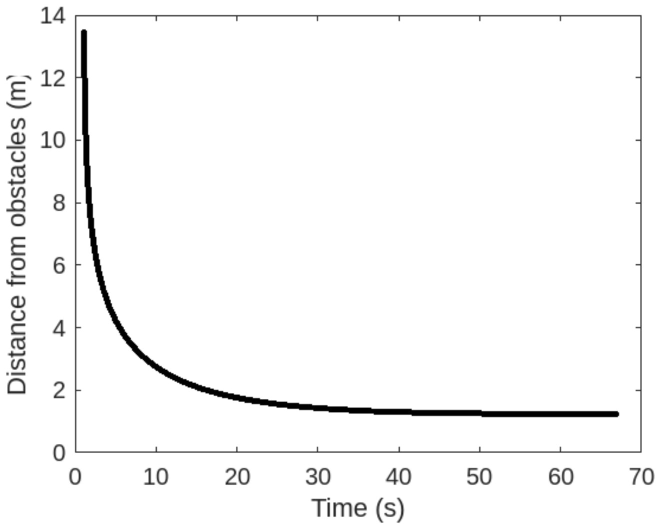


Fig. 9: Distance of the UAV from the nearest obstacle during descent (shown with respect to time). This clearly indicates the safe landing of the UAV.

environments," *Electronics*, vol. 7, no. 5, 2018. [Online]. Available: https://www.mdpi.com/2079-9292/7/5/73

[16] M. Ikura, L. Miyashita, and M. Ishikawa, "Real-time landing gear control system based on adaptive 3d sensing for safe landing of uav," in *2020 IEEE/SICE International Symposium on System Integration (SII)*, 2020, pp. 759–764.

[17] D. Maturana and S. A. Scherer, "3d convolutional neural networks for landing zone detection from lidar," *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3471–3478, 2015.

[18] O. G. Lorenzo, J. Martínez, D. L. Vilariño, T. F. Pena, J. C. Cabaleiro, and F. F. Rivera, "Landing sites detection using LiDAR data on manycore systems," *The Journal of Supercomputing*, vol. 73, no. 1, pp. 557–575, Jan. 2017.

[19] F. Liu, J. Shan, B. Xiong, and Z. Fang, "A real-time and multi-sensor-based landing area recognition system for uavs," *Drones*, vol. 6, no. 5, 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/5/118

[20] M.-F. R. Lee, A. Nugroho, T.-T. Le, Bahrudin, and S. N. Bastida, "Landing area recognition using deep learning for unammaned aerial vehicles," in *2020 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, 2020, pp. 1–6.

[21] H.-W. Cheng, T.-L. Chen, and C.-H. Tien, "Motion estimation by hybrid optical flow technology for uav landing in an unvisited area," *Sensors*, vol. 19, no. 6, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/6/1380

[22] J. González-Trejo, D. Mercado-Ravell, I. Becerra, and R. Murrieta-Cid, "On the visual-based safe landing of uavs in populated areas: A crucial aspect for urban deployment," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7901–7908, 2021.

[23] G. Castellano, C. Castiello, C. Mencar, and G. Vessio, "Crowd detection in aerial images using spatial graphs and fully-convolutional neural networks," *IEEE Access*, vol. 8, pp. 64 534–64 544, 2020.

[24] K. Mukadam, A. Sinh, and R. Karani, "Detection of landing areas for unmanned aerial vehicles," in *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, 2016, pp. 1–5.

[25] J. L. Sanchez-Lopez, M. Wang, M. A. Olivares-Mendez, M. Molina, and H. Voos, "A real-time 3d path planning solution for collision-free navigation of multirotor aerial robots in dynamic environments," *Journal of Intelligent & Robotic Systems*, vol. 93, no. 1, pp. 33–53, Feb 2019. [Online]. Available: https://doi.org/10.1007/s10846-018-0809-5

[26] L. Chen, X. Yuan, Y. Xiao, Y. Zhang, and J. Zhu, "Robust autonomous landing of uav in non-cooperative environments based on dynamic time camera-lidar fusion," *ArXiv*, vol. abs/2011.13761, 2020.

[27] J. Guo, C. Liang, K. Wang, B. Sang, and Y. Wu, "Three-dimensional autonomous obstacle avoidance algorithm for uav based on circular arc trajectory," *International Journal of Aerospace Engineering*, vol. 2021, p. 8819618, Apr 2021. [Online]. Available: https://doi.org/10.1155/2021/8819618

[28] R. A. Sasongko, S. S. Rawikara, and H. J. Tampubolon, "Uav obstacle avoidance algorithm based on ellipsoid geometry," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 2, pp. 567–581, Dec 2017. [Online]. Available: https://doi.org/10.1007/s10846-017-0543-4

[29] R. Singh and T. K. Bera, "Obstacle avoidance of mobile robot using fuzzy logic and hybrid obstacle avoidance algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 517, p. 012009, apr 2019. [Online]. Available: https://doi.org/10.1088/1757-899x/517/1/012009

[30] M. Khairudin, R. Refalda, S. Yatmono, H. Pramono, A. Triatmaja, and A. Shah, "The mobile robot control in obstacle avoidance using fuzzy logic controller," *Indonesian Journal of Science and Technology*, vol. 5, no. 3, pp. 334–351, 2020. [Online]. Available: https://ejournal.upi.edu/index.php/ijost/article/view/24889

[31] X. Han, J. Wang, J. Xue, and Q. Zhang, "Intelligent decision-making for 3-dimensional dynamic obstacle avoidance of uav based on deep reinforcement learning," in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2019, pp. 1–6.

[32] X. Dai, Y. Mao, T. Huang, N. Qin, D. Huang, and Y. Li, "Automatic obstacle avoidance of quadrotor uav via cnn-based learning," *Neurocomputing*, vol. 402, pp. 346–358, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220305853

[33] E. Aldao, L. M. González-deSantos, H. Michinel, and H. González-Jorge, "Uav obstacle avoidance algorithm to navigate in dynamic building environments," *Drones*, vol. 6, no. 1, 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/1/16

[34] R. Wei, Q. Zhang, Z. Xu, K. Zhou, and X. Zhao, "Uavs' autonomous collision avoidance in urban space," in *2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)*, 2018, pp. 1–6.

[35] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[36] K. Schauwecker and A. Zell, "Robust and efficient volumetric occupancy mapping with an application to stereo vision," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6102–6107.

[37] J. Wirth, E. Bonugli, and M. Freund, "Assessment of the accuracy of google earth imagery for use as a tool in accident reconstruction," in *SAE 2015 World Congress & Exhibition*. SAE International, apr 2015. [Online]. Available: https://doi.org/10.4271/2015-01-1435

[38] J. Chen, G. Wang, L. Luo, W. Gong, and Z. Cheng, "Building area estimation in drone aerial images based on mask r-cnn," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 5, pp. 891–894, 2021.