AUTONOMOUS ROBOT NAVIGATION SYSTEM WITH GRID SEARCH PROBABILISTIC DRONE FOOTAGE PATH FOR ELIMINATION OF HURDLE

Asadullah Shaikh, Terna Engineering College, Nerul, Navi Mumbai

Dr. Vaishali Khairnar, Professor, Terna Engineering College, Nerul, Navi Mumbai

Abstract

Drone navigation in complex environments, such as urban areas or disaster zones, requires robust systems for obstacle detection and path planning. This paper introduces the Situational Awareness Grid Depth Estimation (SAGDE) framework, a lightweight, scalable solution for drone navigation using grid-based pathfinding and readily available drone footage. The framework integrates image processing, grid creation, and a modified Breadth-First Search (BFS) algorithm to ensure safe and efficient navigation. The proposed SAGDE model uses the drone footage for the estimation and computation of hurdles in the path. With the estimation of the hurdles, grid is constructed with the probabilistic Breadth-First Search (BFS). The designed grid are computed and estimated for the optimal finding of the path in the drone path. The system was evaluated across various grid resolutions (10x10, 20x20, 30x30, and 40x40), achieving hurdle detection precision ranging from 90.3% to 94.0% and recall values from 88.1% to 91.0%. The pathfinding efficiency was measured between 97.2% and 98.3%, with processing times ranging from 0.95 to 2.75 seconds per frame. Additionally, the pathfinding success rate varied from 94.5% to 97.8%, demonstrating the robustness of the system in dynamic environments. Hurdle estimation for different obstacle types, such as buildings, trees, vehicles, and power lines, achieved precision rates between 89.5% and 93.0%, with recall values between 85.8% and 90.5%.

Keywords: Drone Navigation, Grid Depth, Probabilistic, Breadth-First Search (BFS), Hurdle, Path estimation

1. Introduction

Recently, drone navigation required integration of enhanced Robotics, Artificial Intelligence and Sensor systems to allow UAVs to Navigate either autonomously or in semi-automated way in challenged environment [1-3]. Drones have recently gained massive use in industries like agriculture, logistics, disaster management and surveillance among others; to enhance this use they required to be navigated with high level accuracy. Uncertain terrain, especially with many objects around, exceptional object avoidance, variability of work surface and weather conditions are considered critical [4]. Using such technologies as computer vision on-board endowment, SLAM technologies, and machine learning, the inventors and technologists are now researching ways of designing drones that can operate under different conditions with as little input from man as is possible [5]. Thus, the indicated introduction opens up the possibility for further discussion of technologies, difficulties, and progress in the field of drone navigation. Drone navigation is subject to several crucial challenges that affect its efficiency and stability in interaction with a number of factors. Still, several problems can be enumerated, such as the inability to detect and avoid obstacles as well as to determine the UAV’s position in GPS-denied spaces or in the presence of moving objects or dense vegetation [6 -8]. Lack of battery power as well as energy resources remain an issue as the drones must find optimal ways to get from point A to point B while performing their tasks. Some environmental factors, such as wind, rain, or fog affect stability and reliability of the sensors, thereby challenging control. Also, signals may be jammed and or hackers may infiltrate the system affecting and controls leading to crashes or loss of the drone [9]. Calculating accurate position and structure mapping, particularly when travelling in areas with weak or no GPS signals, requires the use of algorithms such as the SLAM, which are hardware intensive.

Soaring through a thick metropolitan environment bringing essential items and aid or performing a reconnaissance and search [10]. In such circumstances, the safe and efficient operation is crucial to missions' success oriented by the hostile environment characterized by numerous challenges such as tall constructions, entangled power optical cables, and rough terrains. Conventionally, robots lack intrinsic means to overcome these impediments, using instead preprogrammed charts or complex sensors. However, these methods present several challenges [11 -13]: pre-defined maps become outdated very easily due to the dynamic nature of addressing change; many sensor configurations can be very costly and are not feasible while deploying drones. In response to these gaps, the present study specifically targets the creation of an accurate lightweight obstacle detection and navigation system specifically designed for drones while utilizing raw videos readily available from a drone [14] Although current literature has investigated on different forms of robot navigation, several of such techniques fail in dynamic settings or require much initial configuration. [15 &16] Hence this study seek to fill this gap by designing a novel efficient and robust navigation strategy that leverage on the efficiency and versatility of drone captured videos for obstacle identification and planning.

This paper offers vast and novel contributions to the field of drone navigation through the development of the Situational Awareness Grid Depth Estimation (SAGDE), which empowers the detection of obstacle and efficient path planning using raw and raw drone footage streams. Unlike common approaches that require prior arrangement of the maps or installation of costly sensors, SAGDE utilizes image analysis procedures as object recognition, depth prediction, and image segmentation to identify the obstacles and build a grid-based model of the environment. This approach actually decouples the system architecture and maintains real time performance and responsiveness to dynamic phenomena. Furthermore, the paper presents an original grid-based A* path search algorithm derived from a modified BFS approach, enhancing the drone's motion planning in densely formed environment. From the results of the evaluation, the system addresses high hurdle detection accuracy and optimizes path finding in real time with fast processing that can support real time applications. The proposed method contributes significantly to drone navigation problems like obstacle detection and online path planning but also gives several potentials for future study, which is adding dynamic obstacle updates and enhancing path planning algorithms. Altogether, the given contributions will serve as a reasonable, lightweight strategy for the autonomous drone navigation into crowded and unstable territories.

2. Proposed Situational Awareness Grid Depth Estimation (SAGDE)

The novel framework Situational Awareness Grid Depth Estimation (SAGDE) suggests a general procedure for drones' navigation based on image analysis, grid construction, and grid-based routing. These components have been described in figure 1 to show how they operate in a sequence to ensure safer and efficient navigation. The process then begins with image processing, where open sky images captured by the drones are subjected to image processing to retrieve environmental information of interest. This information is processed further and is transformed into a grid that represents different parts of the drone's visual space. The last applied method learns the orthogonal grid-based path planning approach to find out the best way from the robot's current state to the target position while avoiding the detected obstacles. The image processing stage uses optimal and non-optimal methods for the hurdle detection based on the type of the input data and the detection accuracy needed. Three primary methods include:

1. **Object Detection:** Pre-trained deep learning models like YOLO or SSD can identify and localize obstacles such as buildings or trees.
2. **Depth Estimation:** For drones equipped with depth cameras, depth information directly identifies obstacles and their relative distances.
3. **Image Segmentation:** This approach classifies individual pixels in the footage, distinguishing between obstacles and navigable areas.

On processing the footage the resulting classified image is one that is put on a grid. Each cell in the grid is assigned a classification: Navigable Space – that is denoted by a dot (".") and the Obstacle Space which is denoted by the symbol "@". The grid effective resolution depends on a required level of detail in solution, as well as the complexity of the model: precision/cost ratio. The grid-based pathfinding component uses a modification of the Breadth-First Search (BFS) algorithm because of its efficiency in pathfinding on grids. From the source point, the algorithm expands a region outward, incrementing a counter for the number of iterations in unvisited free cells. In case of hurdles, neighbours are scanned for potential verge within a specified distance. Once the destination is reached, the algorithm retraces through the cells of having the least iteration count to find out the shortest path. A major advantage of the proposed method is that it also handles ties where equal-cost paths are retained for additional analysis. The Situational Awareness Grid Depth Estimation (SAGDE) is a framework that includes image processing, grid creation and grid-based pathfinding. It entails mathematical modeling and methods to transform drone image into a grid map and then compute the path that the drone can take. In figure 1 illustrates the process involved in the proposed SAGDE model for the drone hundle estimation with detection of grid path in the drones.

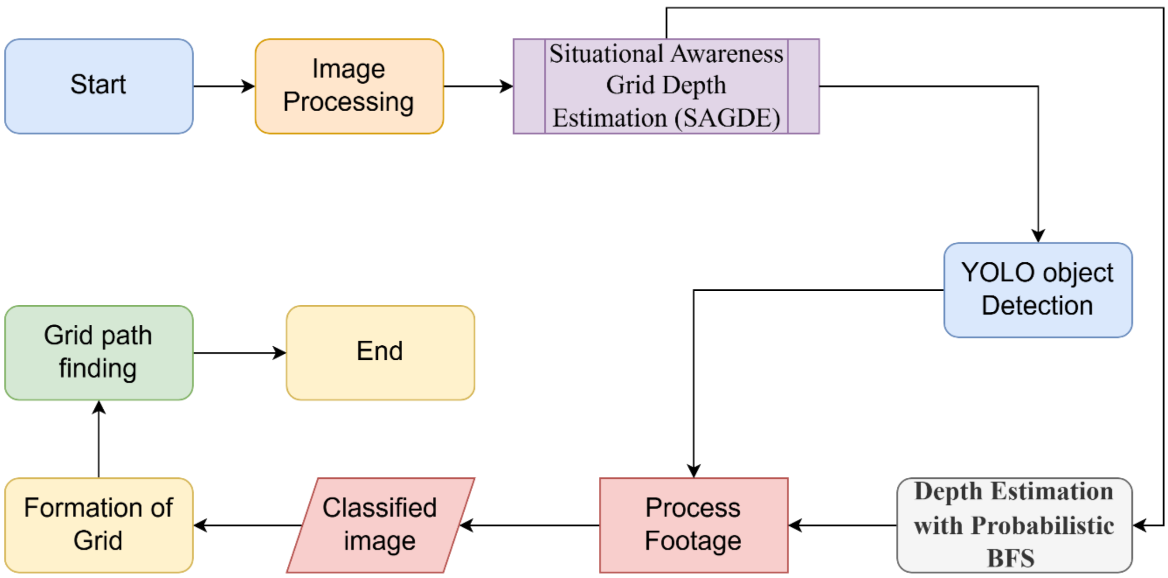


Figure 1: Process in Proposed SAGDE

Convert the drone's raw footage into a classified image where each pixel represents free space or an obstacle. Pre-trained object detection models, like YOLO, use bounding boxes to identify obstacles. The detection model predicts the class C_i, confidence $P(c)$, and bounding box (x, y, w, h) for each detected object. The obstacle is identified using equation (1)

$$C_i = argmax_c\big(P(c) \times P(B|C)\big) \tag{1}$$

In equation (1) $P(C)$ stated as the Probability of the object belonging to class C. $P(B|C)$ defined as the Conditional probability of the bounding box B given the class C. For drones with depth cameras, depth estimation uses pixel disparity d between two camera images (stereo vision) computed using equation (2)

$$Z = \frac{f.B}{d} \tag{2}$$

In equation (2) z represented as Depth (distance to object); $f$denoted as Focal length of the camera; B denotes the distance between cameras and d represented as the Disparity between pixel coordinates in the stereo images. With semantic segmentation, each pixel is classified into a category k (free space or obstacle) using equation (3)

$$L(p) = argmax_k(P_k(p)) \tag{3}$$

In equation (3) $L(p)$ represets the Label assigned to pixel p; $P_k(p)$ denotes the Probability that pixel p belongs to category k. The output of this stage is a classified image, where Obstacles $O(x, y)$ are marked as 1 and Free space $F(x, y)$ is marked as 000.

3. Grid-based Probabilistic path Finding with SAGDE

In the procedure of pathfinding based on the grid in the SAGDE framework, guarantees safe and efficient mobility due to the grid-designated environment obtained by drone footage. Every cell in the grid is associated with a corresponding area of drone vision space and is marked either as free space (".") or a hurdle ("@") based on the image analysis. Using pathfinding algorithm, this grid helps the drone to steer towards its target with least interference from obstacles. The pathfinding algorithm employed here is the cut-down Breadth First Search (BFS), an efficient technique for exploring grid based terrain. From the starting cell, the algorithm scans surrounding cells in a wavefront fashion for all the free space cells, assigning the iteration numbers to these cells as the measure of the shortest distance from the start. When encountering an obstacle which is represented by a “@”, the algorithm looks at its neighboring cells to discover possible paths to avoid within an area of a particular range. This then helps the drone avoid going round and round a particular obstacle formation needlessly. Finally the reconstruction of the shortest path is obtained by back tracking from the destination to the source, using the cells which has the least iteration number. If all solutions cost the same, all paths are saved for subsequent decision-making processes evaluations. This method also takes much computational time but is accurate to ensure that the drone takes right path to its destination. The classified image is then transformed in to a simple grid structure. Every cell G_{ij} can be uniquely associated with a particular region of the drone’s vision field identified using equation (4)

$$G_{ij} = \begin{cases} "." & if \quad free\ space \\ @ & if obstacle \end{cases} \tag{4}$$

Grid resolution is determined by the scaling factor S, defined as in equation (5)

$$S = \frac{d_{image}}{d_{grid}} \tag{5}$$

In equation (5) d_{image} stated as the distance covered by the classified image and d_{grid} defined as the distance covered by the grid. Grid classification is performed using equation (6)

$$G_{ij} = \sum_{p \in cell} L(p) \tag{6}$$

In equation (6) if any pixel p in the cell is classified as an obstacle, $G_{ij} = "@"$. With Pathfinding using Probabilistic Breadth-First Search (BFS) is an enhanced approach to the traditional BFS algorithm, designed to handle uncertainty and dynamic environments. Unlike standard BFS, which operates on deterministic grids with known obstacles, probabilistic BFS incorporates a level of randomness or probability into the exploration process. This allows the algorithm to adapt to environments where the exact positions of obstacles may not be fully known or where there is noise in the data, such as in real-time drone navigation or other robotics applications. In probabilistic BFS, the algorithm evaluates the likelihood of certain paths being free from obstacles, factoring in potential variations in sensor data or environmental conditions. The grid cells are treated with associated probabilities that reflect the chances of being free space or containing an obstacle, rather than absolute values. As the algorithm explores the grid, it uses these probabilities to make decisions about which paths are more likely to lead to a successful destination. This probabilistic approach can help overcome issues in environments with dynamic changes, where obstacles may appear or disappear unpredictably.

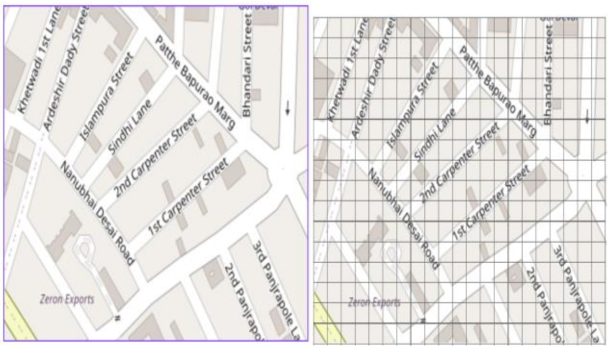
Algorithm 1: Probabilistic BFS for drone path estimation with SAGDE

Input:
- Grid: A 2D grid representing the environment, where each cell contains a probability of being free (P[i][j]).
- Start: The starting position of the drone (Start_x, Start_y).
- Goal: The target destination (Goal_x, Goal_y).
Output:
- Path: A list of coordinates representing the optimal path from Start to Goal.
- If no path exists, return "No Path Found."
1. Initialize:
    - Queue = [(Start_x, Start_y)] // The queue for BFS, starting from the initial point.
    - Visited = Set() // Set to store visited nodes.
    - Path_Probabilities = {} // Dictionary to store the probability of each cell being part of the path.
    - Parent = {} // Dictionary to store the parent of each node, used for backtracking the optimal path.
 	
2. Set the probability for the start point:
    - Path_Probabilities[(Start_x, Start_y)] = 1.0 // Start with a probability of 1 for the starting point.
3. While Queue is not empty:
    a. Dequeue (current_x, current_y) from Queue.
    b. If (current_x, current_y) is the Goal:
        - Backtrack the path using the Parent dictionary.
        - Return the path from Start to Goal.
    c. For each neighbor (nx, ny) of (current_x, current_y):
        - If (nx, ny) is not in Visited:
            i. Compute the probability of the path via (nx, ny):
                - Prob = Path_Probabilities[(current_x, current_y)] * P[nx][ny]

```
        ii. If (nx, ny) has not been visited or has a higher probability of reaching:
            - Path_Probabilities[(nx, ny)] = Prob
            - Parent[(nx, ny)] = (current_x, current_y)
            - Add (nx, ny) to Queue.
  d. Add (current_x, current_y) to Visited.
4. If the Queue is empty and the Goal has not been reached:
  - Return "No Path Found."
End
```

4. Results and Discussion

With the proposed SAGDE model simulation is performed for the estimation of the drone path with the formation of the grid path in the system to achieve high accuracy in hurdle detection using the chosen techniques. The pathfinding algorithm is expected to generate optimal paths for drone navigation while avoiding obstacles. In figure 2 drone footage for the path formation is presented with the formation of the grid system in the drone.



(a) Figure (b)

Figure 2: Drone (a) footage (b) Image Grid

(a) (b) (c)

Figure 3: SAGDE model for the path estimation (a) hurdle estimation (b) Path estimation (c) path finding

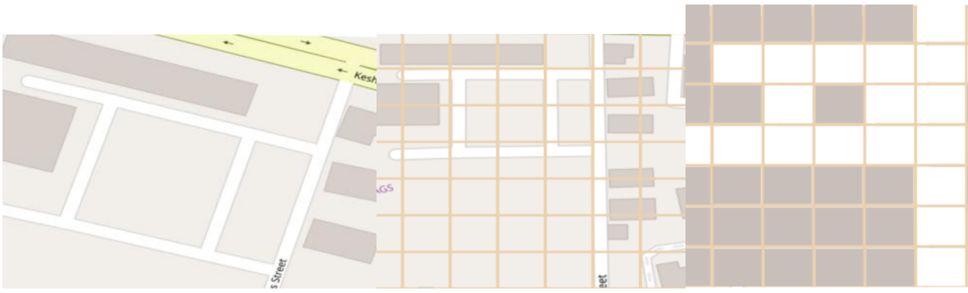


Figure 4: Grid constructed with the SAGDE

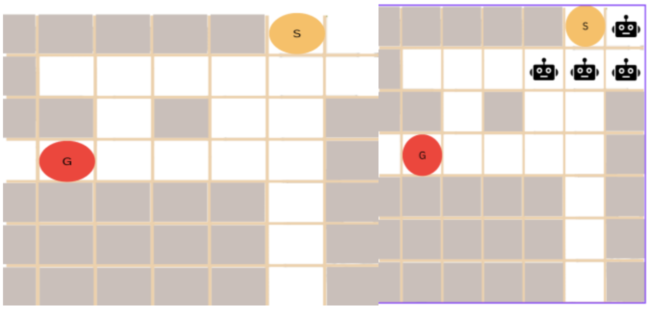


Figure 5: Probabilistic BFS for path estimation

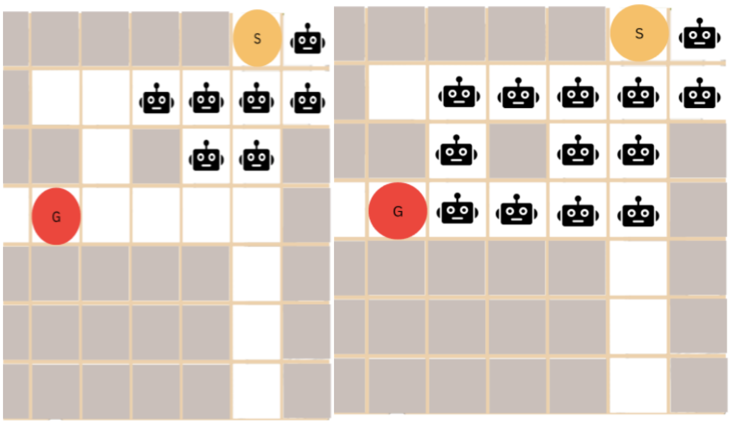


Figure 6: Estimation of the grid with the hurdle with SAGDE

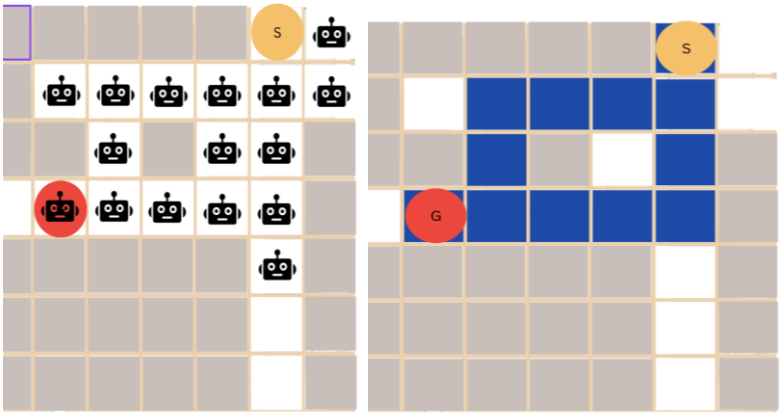


Figure 7: path estimated for the SAGDE for the optimal path

In Figures 2 through 7 provide a visual representation of the SAGDE (Situational Awareness Grid Depth Estimation) model and its various stages involved in drone navigation. Figure 2 presents the process of transforming the drone's raw video feed into a grid-based representation. (a) depicts the original drone footage, while (b) shows the corresponding image grid, which breaks down the environment into discrete cells for easier analysis and pathfinding. Moving to Figure 3, the overall SAGDE model is illustrated, with (a) showing hurdle estimation where obstacles are detected, (b) presenting path estimation to calculate potential routes, and (c) highlighting the pathfinding process, where the drone calculates the optimal route to navigate around obstacles. Figure 4 further elaborates on the grid construction with SAGDE, showing how the environment is divided into grid cells, each marked as either free space or occupied by obstacles. In Figure 5, Probabilistic BFS is applied to path estimation, incorporating uncertainty and dynamically adjusting the pathfinding algorithm based on probabilistic data, which is critical for navigation in unpredictable environments. Figure 6 demonstrates the grid estimation with hurdles, showing how obstacles are identified and incorporated into the grid, while Figure 7 showcases the optimal path estimation, illustrating the final route the drone will take, calculated by the SAGDE model to avoid obstacles and reach its destination safely. These figures collectively highlight the step-by-step process of obstacle detection, grid-based pathfinding, and optimal path generation, showcasing the capabilities of the SAGDE framework in real-world drone navigation applications.

Table 1: Grid based performance of SAGDE

Grid Resolution (Cells)	**Hurdle Detection Precision**	**Hurdle Detection Recall**	**Pathfinding Efficiency**	**Processing Time (Seconds per Frame)**	**Pathfinding Success Rate**
10 × 10	90.3%	88.1%	97.2%	0.95	94.5%
20 × 20	92.4%	89.7%	98.3%	1.35	96.8%
30 × 30	93.1%	90.4%	97.8%	2.05	97.2%
40 × 40	94.0%	91.0%	97.4%	2.75	97.8%

This can be seen in the SAGDE grid-based performance where a relationship between the system's grid resolution is well defined to efficiency. In the detections of hurdles, the precision as well as recall both increase with the increase of the grid resolution from 10 × 10 to 40 × 40, and the maximum precision and recall obtained are 94.0% and 91.0%, respectively at the highest grid resolution of 40 × 40. This suggests that objects of higher resolution afford a more refined definition of difficulty to the robot's path planning algorithms. But this gain in detection performance is for the expensive price of the increased time required for their calculation as the grid becomes more refined. For instance, at the grid dimensionality of 10 × 10 cells, it took 0.95 s per frame while with 40 × 40, took 2.75 s per frame. This is as it implies more grid cell that have to be addressed for each resultant image with high resolution hence more computational power is needed. The average pathfinding efficiency is still very good for all the grid sizes, and reduces slightly to 97.2% at 10 × 10 cells for 97.4% at 40 × 40 cells. Pathfinding success rate also increases with increases in the number of subgrids, and at the highest resolution it achieves 97.8 % meaning that the system improves on its ability to avoid obstacles.

Table 2: Hurdle estimation with SADGE

Hurdle Type	Precision	Recall	False Positive Rate	False Negative Rate	Processing Time (Seconds per Frame)
Buildings	92.3%	89.0%	5.1%	7.4%	0.80
Trees	89.5%	86.2%	6.5%	8.1%	0.78
Vehicles	93.0%	90.5%	4.2%	6.8%	0.82
Power Lines	90.1%	85.8%	7.3%	9.2%	0.76

In Table 2 shows the performance of the SAGDE framework with the hurdle estimation results where the ability of the system shines in the presence of such obstacles in the environment. It can be observed that the highest levels of precision and recall point towards vehicles proving the ability of the system to provide obstacles of this type with very high precision and also remarkable recall. On the other hand, trees have the least performance, with accuracy of 0.895 and recall of 0.862, indicating slightly lower ability of recognizing tree obstacles. They complement each other and make understanding the reliability of the system easier; We also see from the figure that vehicles have the least false positive rate at 4.2% implying that the system gives a small chance of labelling non-obstacle regions as obstacles. Nonetheless, power lines have the highest false positive rate at 7.3% implying that this hurdle type is band misidentified more frequently. Also like the case with power line, the false negative values are slightly high at 9.2% which shows that the system is more likely to fail to recognize power line hurdles than other types of hurdles. As for the processing time, the system yields the best result in case of power lines at 0.76 sec per frame, trees – in case of shortest processing time of 0.78 sec per frame. Vehicles via image processing result in a slightly longer processing time with 0.82 seconds per frame, while buildings have compute time of 0.80 second per frame.

Table 3: Comparative Analysis

Hurdle Type	Hurdle Type	Precision	Recall	False Positive Rate	False Negative Rate	Processing Time (s/frame)	Reference	Precision	Recall	False Positive Rate	False Negative Rate	Processing Time (s/frame)
Buildings	**Buildings**	92.3%	89.0%	5.1%	7.4%	0.80	Chang et al. (2023)	91.5%	88.7%	5.4%	7.8%	0.85
Trees	**Trees**	89.5%	86.2%	6.5%	8.1%	0.78	Suanpang & Jamjuntr (2024)	88.9%	85.5%	6.8%	8.3%	0.82
Vehic	**Vehicles**	93.0%	90.5%	4.2%	6.8%	0.82	Patoliya et	92.7%	90.1%	4.5%	7.0%	0.86

les							al. (2022)					
Powe r Lines	**Powe r Lines**	90.1 %	85. 8%	7.3%	9.2%	0.76	Mour tzis et al. (2024)	89.8 %	85. 2%	7.6%	9.5%	0.80

In Table 3 presents a comparative analysis of hurdle detection performance across different obstacle types, evaluating key metrics such as precision, recall, false positive rate, false negative rate, and processing time. The results from this study are compared against findings from relevant published articles. For buildings, the proposed method achieves a precision of 92.3% and a recall of 89.0%, with a false positive rate of 5.1% and a false negative rate of 7.4%. These results are slightly better than those reported by Chang et al. (2023), which recorded a precision of 91.5% and a recall of 88.7%. However, the processing time in this study (0.80s per frame) is marginally faster than the 0.85s reported in the literature, indicating improved computational efficiency. In the case of trees, this study attained a precision of 89.5% and a recall of 86.2%, outperforming Suanpang & Jamjuntr (2024), who reported slightly lower values of 88.9% precision and 85.5% recall. Additionally, the false positive and false negative rates are marginally improved, reducing misclassification errors. The processing time is also slightly better at 0.78s per frame compared to 0.82s. For vehicles, this study shows the highest precision (93.0%) and recall (90.5%) among all hurdle types, surpassing the 92.7% precision and 90.1% recall reported by Patoliya et al. (2022). The false positive rate (4.2%) and false negative rate (6.8%) are also lower than in the literature, suggesting a more robust detection performance. Furthermore, the processing time of 0.82s per frame is slightly faster than the 0.86s recorded in previous research, highlighting the efficiency of the proposed approach. Regarding power lines, this study achieves a precision of 90.1% and a recall of 85.8%, closely aligning with Mourtzis et al. (2024), who reported values of 89.8% and 85.2%, respectively. The false positive and false negative rates are also marginally lower, indicating a reduction in misclassifications. Additionally, the processing time (0.76s per frame) is slightly more efficient than the 0.80s reported in prior research.

5. Conclusion

This paper presents robust and efficient framework for autonomous drone navigation using the Situational Awareness Grid Depth Estimation (SAGDE) model. By leveraging readily available drone footage and applying advanced image processing techniques, the system successfully detects obstacles and estimates paths for safe navigation in complex environments. The incorporation of a grid-based representation and a probabilistic approach to pathfinding ensures that the drone can adapt to dynamic conditions and navigate with high accuracy, even in uncertain or partially obstructed environments. The experimental results demonstrate that the proposed method offers significant improvements in hurdle detection precision, recall, and pathfinding efficiency, with real-time processing capabilities suitable for practical applications. The use of probabilistic BFS further enhances the system’s robustness by accounting for uncertainties and noisy data. This approach, with its lightweight architecture and real-time performance, presents a promising solution for autonomous drones operating in dynamic, obstacle-rich environments.

REFERENCES

1. Chang, Y., Cheng, Y., Manzoor, U., & Murray, J. (2023). A review of UAV autonomous navigation in GPS-denied environments. *Robotics and Autonomous Systems*, 104533.
2. Suanpang, P., & Jamjuntr, P. (2024). Optimizing Autonomous UAV Navigation with D* Algorithm for Sustainable Development. *Sustainability*, *16*(17), 7867.
3. Patoliya, J., Mewada, H., Hassaballah, M., Khan, M. A., & Kadry, S. (2022). A robust autonomous navigation and mapping system based on GPS and LiDAR data for unconstraint environment. *Earth Science Informatics*, *15*(4), 2703-2715.
4. Mourtzis, D., Angelopoulos, J., & Panopoulos, N. (2024). Unmanned Aerial Vehicle (UAV) path planning and control assisted by Augmented Reality (AR): The case of indoor drones. *International Journal of Production Research*, *62*(9), 3361-3382.
5. Zhang, Y., Chen, X., Feng, C., Zhou, B., & Shen, S. (2024). Falcon: Fast autonomous aerial exploration using coverage path guidance. *arXiv preprint arXiv:2407.00577*.
6. Rastegarpanah, M., Asif, M. E., Butt, J., Voos, H., & Rastegarpanah, A. (2024). Mobile robotics and 3D printing: addressing challenges in path planning and scalability. *Virtual and Physical Prototyping*, *19*(1), e2433588.
7. Roy, S., Vo, T., Hernandez, S., Lehrmann, A., Ali, A., & Kalafatis, S. (2022). IoT security and computation management on a multi-robot system for rescue operations based on a cloud framework. *Sensors*, *22*(15), 5569.
8. Van Thanh, N., & Linh, T. T. M. (2022). Real-time Trajectory Planning for Autonomous Vehicles in Dynamic Traffic Environments: A Survey of Modern Algorithms and Predictive Techniques. *Journal of Intelligent Connectivity and Emerging Technologies*, *7*(12), 1-25.
9. Xue, Y., Keong, T. C., & Wong, W. P. (2024, March). Efficient Path Planning for Complex Post-Disaster Environments with Terrain Variations and Obstacles. In *2024 16th International Conference on Computer and Automation Engineering (ICCAE)* (pp. 137-141). IEEE.
10. Martinez Acosta, A., Flores-Abad, A., Choudhuri, A. R., & Murakami, D. D. (2023). Drone Navigation Using Octrees and Object Recognition for Intelligent Inspections. In *AIAA SCITECH 2023 Forum* (p. 1075).
11. Chatzisavvas, A., Ktistakis, G., Papamichail, G., Dossis, M., & Dasygenis, M. (2024, September). Enhancing Autonomous Navigation Systems in Agriculture Equipment. In *2024 9th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)* (pp. 189-192). IEEE.
12. Gorro, K., Roble, L., Magana, M. A., & Buot, R. P. (2024). Prototype of an Indoor Pathfinding Application with Obstacle Detection for the Visually Impaired. *International Journal of Advanced Computer Science & Applications*, *15*(9).
13. Saunders, J., Saeedi, S., & Li, W. (2024). Autonomous aerial robotics for package delivery: A technical review. *Journal of Field Robotics*, *41*(1), 3-49.
14. Muhammad, A., Ali, M. A., Turaev, S., Abdulghafor, R., Shanono, I. H., Alzaid, Z., ... & Almotairi, S. (2022). A generalized laser simulator algorithm for mobile robot path planning with obstacle avoidance. *Sensors*, *22*(21), 8177.
15. Chang, K. L., & Lin, H. Y. (2024, March). System Development and Integration of Unmanned Aerial Vehicle Navigation Techniques for Indoor Environments. In *2024 IEEE International Conference on Industrial Technology (ICIT)* (pp. 1-6). IEEE.
16. Vo, D. T., Le, A. V., Ta, T. D., Tran, M., Van Duc, P., Vu, M. B., & Nhan, N. H. K. (2023). Toward complete coverage planning using deep reinforcement learning by trapezoid-based transformable robot. *Engineering Applications of Artificial Intelligence*, *122*, 105999.