PROGRESS IN AEROSPACE AND AVIATION TECHNOLOGY
e-ISSN: 2821-2924

PAAT

Vol. 4 No. 1 (2024) 57-71
https://publisher.uthm.edu.my/ojs/index.php/paat

The Comparative Study Between the Grid Size of Image Frame Using Image Subtraction and Pixel Expansion Cue for Free Obstacle Region Detection

Mohammad Anwaar Badrol[1], Muhammad Faiz Ramli[2]*

[1] *Faculty of Mechanical and Manufacturing Engineering (FKMP),*
Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, MALAYSIA

[2] *Department of Aeronautical Engineering, Faculty of Mechanical and Manufacturing Engineering (FKMP),*
Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, MALAYSIA

*Corresponding Author: faizr@uthm.edu.my

DOI: https://doi.org/10.30880/paat.2024.04.01.007

Article Info

Received: 13 December 2023
Accepted: 8 March 2024
Available online: 30 June 2024

Keywords

Free region detection, grid size, number of grid, computational time

Abstract

This research investigates optimizing grid sizes for obstacle detection in Unmanned Aerial Vehicles (UAVs) using vision-based approaches. There were many proper computer visions algorithm that can be used to detect free obstacle region. One of it is by using grid-based approach. By dividing the image that taken by camera, the algorithm can choose the best path to go through. This study will explore various grid sizes, from 3x4 to 24x32, to determine their effectiveness in detecting free obstacle regions in different environments. The methodology combines image subtraction, pixel expansion cues, and k-means segmentation, with images resized to 640x360 pixels and processed to display only two colors for simplification. The experiments, conducted using an iPhone 12 camera and analyzed with Spyder software and OpenCV, reveal that larger grids are more accurate in complex environments, albeit with higher computational time, whereas smaller grids are more efficient in simpler settings. The research identifies a grid size range of 12x16 to 16x24 as a balanced solution for various scenarios, highlighting the need for adaptive sizing and sophisticated algorithms in UAV obstacle detection.

1. Introduction

The study titled "The Comparative Study Between the Grid Size of Image Frame Using Image Subtraction and Pixel Expansion Cue for Free Obstacle Region Detection" addresses a gap in autonomous vehicle technology. It aims to identify the optimal grid size for obstacle detection in UAVs, a critical aspect for navigating complex environments without pre-built maps. With the increasing focus on Unmanned Vehicles (UVs), ensuring safety through accurate environment perception becomes crucial. Vision-based approaches are gaining popularity over traditional sensor technologies like LiDAR, which are limited in certain scenarios [1] [2]. This study proposes a method using image subtraction and pixel expansion cues to improve obstacle detection in UAVs [3][4].

The research identifies a gap in understanding how different grid sizes impact obstacle detection, especially in varying environments. No comprehensive studies have been conducted to systematically compare the effectiveness of different grid sizes in obstacle detection, leading to challenges in selecting the most suitable size for specific applications. This study aims to fill this gap by comparing various grid sizes' effectiveness in detecting free obstacle regions and investigating factors affecting their performance. The scope of the study includes focusing on UAVs, using Spyder software and OpenCV for image processing, and testing seven different grid sizes.

The significance of this research lies in its potential to enhance autonomous system technologies by providing insights into the most efficient grid sizes for obstacle detection, aiding developers and researchers in the field.

2. Methodology

In this study, the image will be processed in an algorithm conducted through a Spyder software and OpenCV for image processing. The aim is to compare the effectiveness of different grid size, in detecting free obstacle regions and the factor influenced it. To accomplish this, several steps are undertaken in the experimental process.

2.1 Framework Design

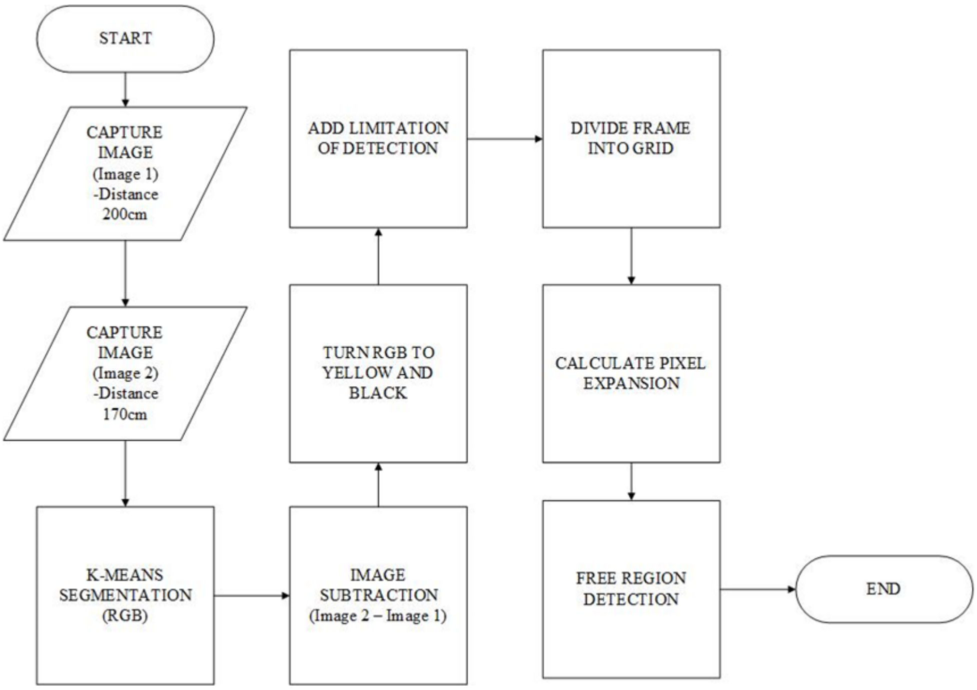

START
CAPTURE IMAGE (Image 1) -Distance 200cm
CAPTURE IMAGE (Image 2) -Distance 170cm
K-MEANS SEGMENTATION (RGB)
IMAGE SUBTRACTION (Image 2 – Image 1)
TURN RGB TO YELLOW AND BLACK
ADD LIMITATION OF DETECTION
DIVIDE FRAME INTO GRID
CALCULATE PIXEL EXPANSION
FREE REGION DETECTION
END

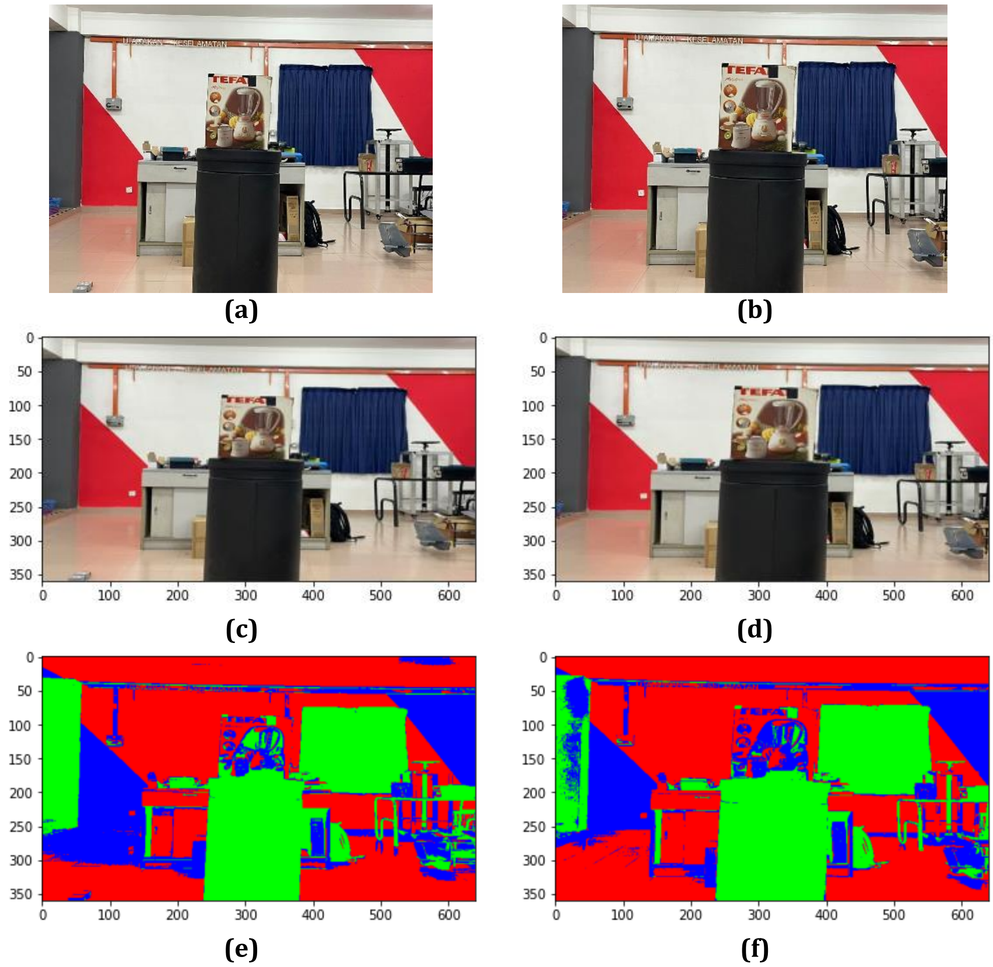
Fig. 1 *Framework design*

The methodology involves a systematic approach to image capture and processing for obstacle detection. The frameworks show the algorithm were used in this study. The process starts by capturing two images from an iPhone 12 camera, positioned at distances of 200 cm and 170 cm from an obstacle. These images undergo several stages of processing using Spyder software and OpenCV in Python. Initially, both images are resized to a consistent resolution of 640×360 pixels. This uniformity is crucial for accurate image subtraction, as it allows for direct pixel-by-pixel comparison. Subsequently, the images are segmented using the k-means algorithm, with a focus on simplifying the images into three primary colors (Red, Green, and Blue). This segmentation is pivotal for tasks like object detection and classification. The images are then subtracted from each other, with this subtraction performed for each color channel, highlighting changes between the two images. To further simplify the process for obstacle detection, the images are converted into a two-color format, yellow and black, which significantly reduces computational requirements and enhances clarity for differentiating obstacles from free regions. Finally, the images are divided into grids of various sizes, ranging from 3x4 to 24x32, to evaluate free region detection.

2.1.1 Capture Image

Since this experiment involves the analysis of any two images, it's worth noting that these parameters can be applied to other cameras as well, not just limited to a drone's camera, like Pi Camera [5]. Even a monocular camera, such as a phone's camera, can be utilized for this purpose.

In this experiment, iPhone 12's camera will be used. The basic lens of the iPhone 12 boasts a wide aperture that allows for excellent low-light performance, making it suitable for various lighting conditions. It has a specific focal length, typically around 26mm (35mm equivalent). When used with a 1.5× zoom, it has a specific focal length, typically around 39mm (35mm equivalent), which provides a standard field of view.



(a) (b)

(c) (d)

(e) (f)

Fig. 2 *Image of (a) Capture image 1; (b) Capture image 2; (c) Resize image 1; (d) Resize image 2; (e)K-means segmentation- image 1; (f) K-means segmentation – image 2*

2.1.2 Resize Image

Resizing images to a consistent resolution of 640×360 pixels in an algorithm installed in a micro-computer is critical for effective image subtraction. This uniformity in size allows for a direct, pixel-by-pixel comparison between images, which is essential for the accuracy of image subtraction techniques. By ensuring that all images have the same dimensions, the algorithm avoids any misalignment issues that could otherwise lead to inaccuracies in the subtraction process.

Having images of the same resolution streamlines the entire process, particularly in systems with limited computational resources, like micro-computers. It not only facilitates a smoother subtraction operation but also aids in resource management. Consistent image sizes consume less memory and require less processing power, which is vital for applications that need rapid processing, such as real-time surveillance or dynamic environmental monitoring.

2.1.3 K-means Segmentation

The k-means algorithm stands as a widely-used method for data segmentation and grouping tasks. Its primary objective is to divide a dataset into k distinct clusters. Each data point is assigned to the cluster whose mean or centroid is nearest to it, ensuring that the data points are grouped based on similarity.

The choice of the k value is critical as it dictates the number of clusters to be formed in the dataset. This value is selected by the researcher based on the specific requirements of the task at hand. For instance, in image segmentation tasks, different values of k can lead to varying levels of granularity in the segmentation.

In this specific application, the k-means algorithm is employed with a k value of 3. This is chosen to segment the image into three primary colors: red, green, and blue (RGB). The algorithm will thus identify and group pixels into three clusters, each corresponding to one of these colors. This approach is particularly useful in simplifying the image for further analysis, where distinguishing between these primary colors could be crucial for tasks such as object detection, image classification, or further image processing steps.

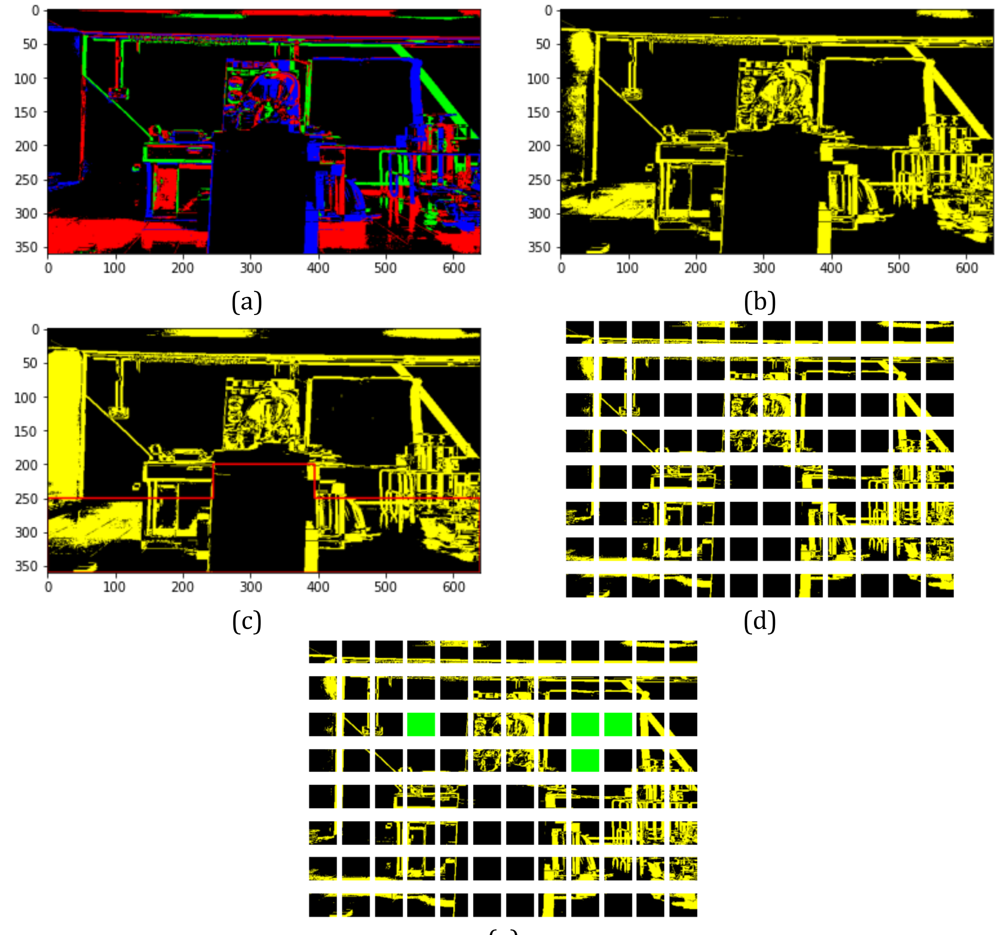


Fig. 3 *Image of (a) Image Subtraction; (b) Turn RGB to 2 colour; (c) Limitation of Detection; (d) Divided Frames into Grid; (e) Obstacle detection*

2.1.4 Image Subtraction

The core of image subtraction lies in its pixel-by-pixel comparison. Each corresponding pixel from two images is subtracted, resulting in a new pixel value that represents the difference. This subtraction is performed for each color channel (Red, Green, Blue) of the pixel. For example, if a pixel in the first image has an RGB intensity of (R, G, B) and the corresponding pixel in the second image has an intensity of (R', G', B'), the subtracted pixel will have an intensity of (R-R', G-G', B-B').

This process effectively brings out the disparities between the two images. Changes in color values can be indicative of various alterations in the scene, such as the movement of objects or changes in lighting. In the context of obstacle detection, for example, the subtracted image can reveal how the distance from obstacles affects the appearance of different areas, thus providing valuable insights into spatial changes or movements within the visual field. This makes image subtraction a powerful tool for analyzing dynamic environments and detecting alterations over time.

2.1.5 Turn RGB to 2 Colour

In the process of simplifying image analysis for obstacle detection, a key step is converting the standard RGB color scheme to just two colors: yellow and black. This method involves a specific color recognition algorithm to identify yellow pixels, a color typically characterized by high red and green values and a lower blue value[4]. When a pixel's RGB values fall within this defined range for yellow, the algorithm assigns it the color yellow. Conversely, if a pixel's RGB values do not match these criteria, it is set to black.

This conversion to a two-color format is primarily driven by the need to reduce computational time. Working with only two colors simplifies the processing requirements, as the algorithm no longer needs to handle the complexities of the full RGB spectrum. This is especially beneficial in real-time applications where speed is crucial.

Additionally, using just yellow and black enhances the clarity in differentiating between obstacles and free regions. Obstacles can be marked in one color and free regions in another, creating a straightforward visual distinction. This binary color coding makes it easier for the algorithm to quickly identify safe passages and potential hazards, streamlining the decision-making process in autonomous navigation systems or similar application.

2.1.6 Limitation of Detection

In simulating a flying drone, setting specific limitations for region detection is crucial to prevent the drone from mistakenly identifying the floor as a free region. Given the image resolution of 640x360 pixels, these limitations are defined using a geometric shape that excludes certain areas from being considered as free regions.

The coordinates for this geometry are strategically chosen to outline an area on the image that represents the floor. This area is delineated by the coordinates (0, 250), (0, 360), (640, 360), (640, 250), (395, 250), (245, 250), and back to (0, 250). These points create a polygonal shape that effectively marks the lower part of the image, typically where the floor would appear in the drone's camera view.

By excluding this geometric area from the free region detection, the simulation ensures that the drone does not consider the floor as a navigable space. This is particularly important to prevent the drone from descending towards the floor, which in real-world scenarios could lead to collisions or unsafe flying conditions. This method of region exclusion is a key safety measure in the drone's navigation algorithm, ensuring that it only identifies and moves towards valid and safe free regions in its flight path. consideration.

2.1.7 Divide Frame into Grids

In this section, the focus is on dividing images into grids, which are essential for free obstacle region detection. Each image is converted into a 2-color format and then segmented into a grid comprising individual cells, with each cell representing a potential area for analysis.

The study investigates a range of grid sizes, varying from 3x4 to 24x32. This variation results in a total number of grids ranging from 12 to 768, allowing for a comprehensive examination of how different grid sizes affect obstacle detection. Specifically, seven grid sizes are considered: (3x4), (4x6), (6x8), (8x12), (12x16), (16x24), and (24x32).

For categorization purposes, these grid sizes are classified into three groups. The (3x4) and (4x6) grids are labeled as larger grids size, offering a broader view of each cell but potentially less detail. The medium grids size, comprising (12x16) and (16x24), provide a balance between the larger and smaller grids. Finally, the (24x32) grid is categorized as a bigger grid size, offering the most detailed analysis at the expense of a narrower view per cell. This diverse range of grid sizes enables the study to assess the impact of grid dimensions on the effectiveness and accuracy of free obstacle region detection in various size.

2.1.8 Free Region Detection

In this context, the environment is represented as a grid, with each cell analyzed for potential obstacles. The algorithm installed in the computer processes the captured image, dividing it into these grid cells[3]. Each cell is then evaluated to determine if it represents a free region. The criteria for a free region in this setup is straightforward: cells composed entirely of black pixels (100% black) are classified as obstacle-free.

This binary approach simplifies the decision-making process for the path planning algorithm. By categorizing each grid cell as either an obstacle or a free region, the algorithm can efficiently map out a safe and navigable route. The reliance on black-pixel grids to represent free regions helps in quick and efficient path planning, making it a practical solution for real-time navigation in autonomous vehicles.

2.2 Experiment Setup

To accurately replicate the conditions of a drone in flight, the camera and obstacle were positioned at a height of one meter above the ground. To avoid introducing additional variables, a phone's camera is used as a substitute for actual drones in order to simulate the behaviour of a 'flying drone'. The camera, which provides the viewpoint of the drone, was initially placed at a distance of 200 cm from the obstacle in order to capture the initial image. Subsequently, the drone was shifted 30 cm in a forward direction to replicate its progress towards the obstacle, with the purpose of capturing the second image.

In the field of engineering and technology, it is essential to carry out experiments under different conditions in order to verify the precision and significance of the results. This study presents two categories of obstacle: objects with texture and objects without texture. These are selected to represent a variety of environments that a drone may encounter. Moreover, the experiments will be conducted in two separate environments: one with a crowded background and another with a less crowded background. But both of it still in same room. The combination of these different elements aims to offer an in-depth understanding of how various grid size impact obstacle detection and navigation efficiency in drones, which ultimately results in safer and more dependable UAV operations.

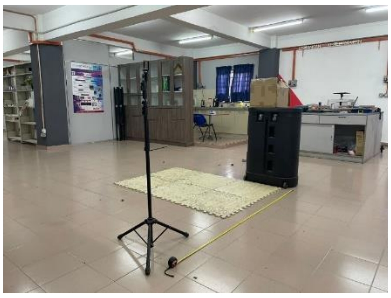
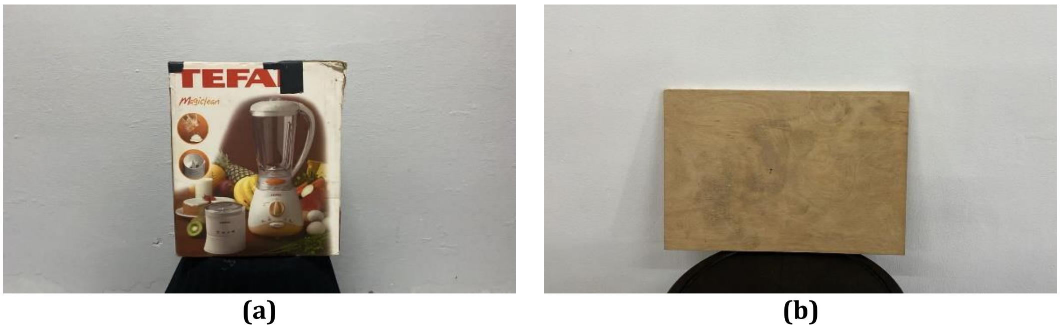
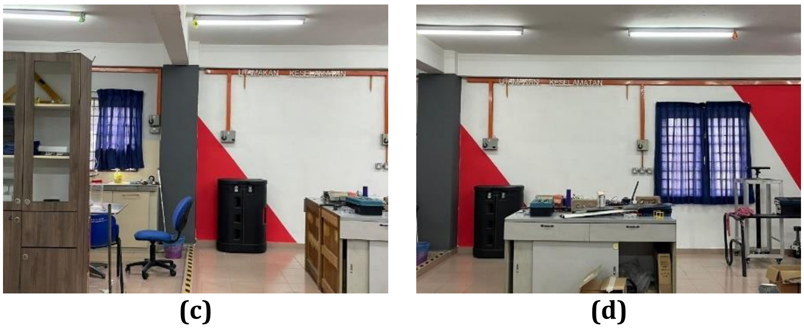


Fig. 4 *Experiment setup*



(a) (b)

Fig. 5 *Types of obstacle (a) Texture object (b) Texture-less object*



(c) (d)

Fig. 6 *Types of environment (c) Complex (d) less-complex*

A texture object (a) refers to an object that has unique patterns or surface variations, which allow it to be visually identifiable from its surroundings. These patterns can appear as variations in colour, repetitive designs, or surface irregularities. Textured objects are typically more easily identifiable by vision-based detection systems due to the presence of unique visual cues that can be captured and analysed by algorithms.

Texture-less objects (b), in contrast, display a lack of visible surface variations and typically have a uniform visual appearance. The lack of texture in their appearance poses a difficulty for vision-based detection systems, as it affects the ability to differentiate them from their surroundings, particularly when their colour or luminance closely resembles that of the background.

A complex environment (c) is a setting that includes various components that may create challenges to the detection system. This includes a range of factors, including different textures, different lighting conditions and various object shapes. Complex environments effectively simulate the real-world conditions in which autonomous vehicles operate. However, they also pose greater challenges for algorithms due to the huge amount of information, as well as the possibility of noise and obstacles.

A less-complex environment (d) is defined by its minimalistic settings and a lack of challenging elements. These factors might involve uniform textures, uniform illumination, and a reduced number of objects. These environments have less requirements on obstacle detection systems, making them suitable for initial testing stages or applications in controlled conditions.

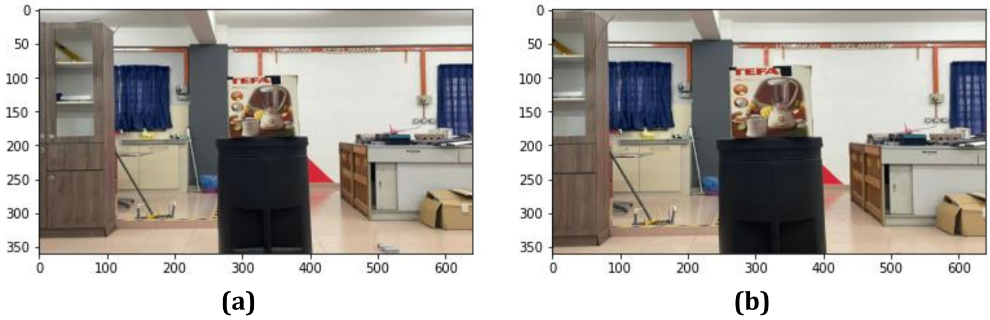
3. Result and Discussion

In order to meet both variables, namely object texture and environment type, then 4 cases need to be created. This experiment divided into 4 cases.

Table 1 *Division of cases*

Case	Type of Obstacle	Type of environment
1	Texture object	Complex
2	Texture object	Less-complex
3	Texture-less object	Complex
4	Texture-less object	Less-complex

3.1 Case 1



(a) **(b)**

Fig. 7 *Case 1 (a) image 1; (b) image 2*

Table 2 *Data for case 1*

Grid Size	Total Number of Grid	Number of Detection	Number of True Detection	Percentage of Detection	Percentage of True Detection	Computational Time	False Positive Rate
3x4	12	0	0	0%	-	2.36	-
4x6	24	0	0	0%	-	3.31	-
6x8	48	1	1	2%	100%	5.1	0%
8x12	96	3	3	3%	100%	7.38	0%
12x16	192	9	9	5%	100%	12.55	0%
16x24	384	27	27	7%	100%	25.78	0%
24x32	768	82	82	11%	100%	56.78	0%

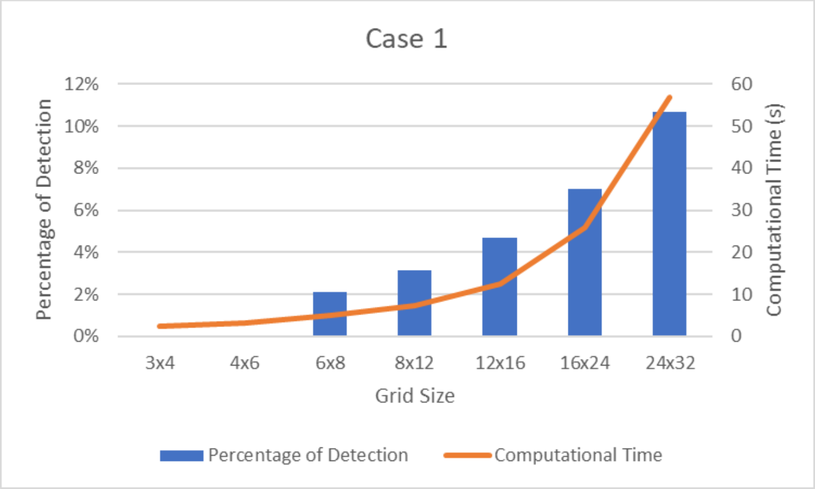


Fig. 8 *Data for case 1*

For Case 1, the study focuses on the detection of textured obstacles within a complex environment, utilizing a range of grid sizes.

Based on Appendices A1, Case 1, For grid 3x4 and grid 4x6, there were no detection of free region. It starts to detect on grid 6x8 with one detection. Then, three regions of free obstacle were detected on grid 8x12. Even as the number of grids increases, the detection still detects free regions in the same area, for grid 12x16, it increased into 9 detections. The detected area was on the right of the obstacle.

When it came with grid 16x24, which is 384 total number of grids, it started to detect the other place. For 1st trial, it detects 26 regions but 1 region were at the other place (still not an obstacle). For 2nd trial, it detects 28 regions with 3 regions were at the other place. For 3rd trial, it detects 27 regions with 2 regions were at the other place. So, with 3 trials, the detected majority regions still precise on a same area with same number of regions which were 25 regions. Based on figure 4.3, for grid 24x32, the size of grids was too small for this environment. It started to detect ceiling as free region.

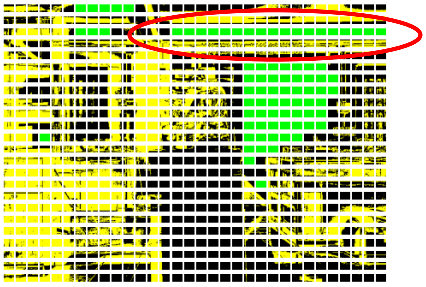
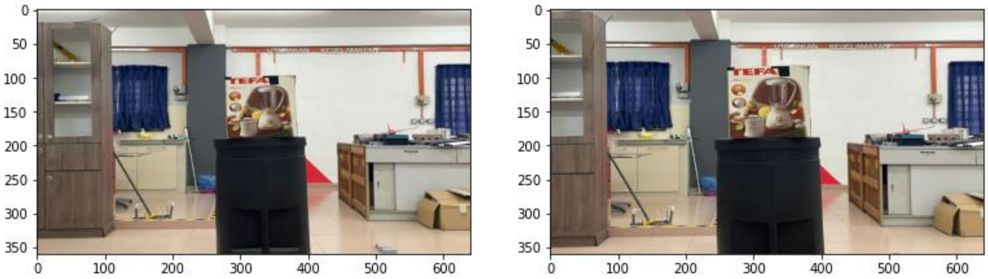


Fig. 9 *The Detection of ceiling*

From the table, we observe that the total number of grids increases with grid size, which is expected since larger grids contain more individual grid elements. Notably, all detections are true, as indicated by the 100% true detection rate, which persists across all grid sizes.

The percentage of detection increases as the grid size becomes larger, peaking at 16% for the 24x32 grid. The computational time also increases with grid size, from 2.26 seconds for the smallest grid to 55.11 seconds for the largest. This indicates that more extensive computations are required for larger grids, which is typical in data processing scenarios where a larger dataset demands more processing time.

3.2 Case 2



(a) **(b)**

Fig. 10 *Case 2 (a) image 1; (b) image 2*

Table 3 *Data for case 2*

Grid Size	Total Number of Grid	Number of Detection	Number of True Detection	Percentage of Detection	Percentage of True Detection	Computational Time	False Positive Rate
3x4	12	0	0	0%	-	2.26	-
4x6	24	0	0	0%	-	2.82	-
6x8	48	3	3	6%	100%	4.27	0%
8x12	96	4	4	4%	100%	7.26	0%
12x16	192	16	16	8%	100%	14.17	0%
16x24	384	44	44	11%	100%	25.73	0%
24x32	768	122	122	16%	100%	55.11	0%

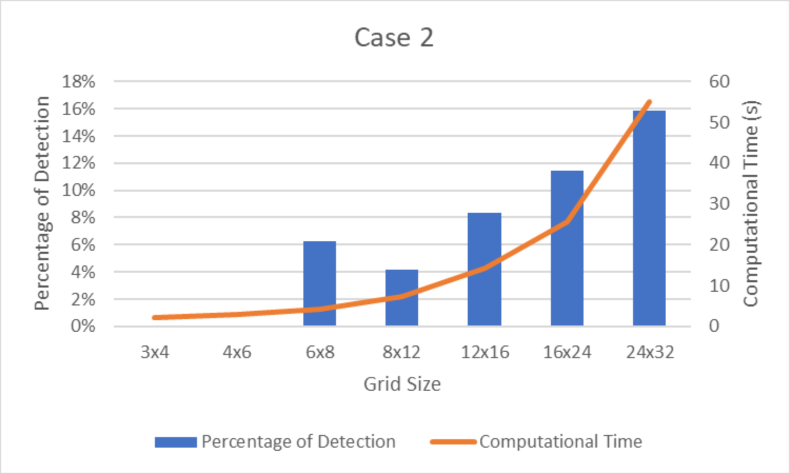


Fig. 11 *Data for case 2*

In Case 2, the study delves into the detection of textured obstacles in less-complex environments, employing various grid sizes to gauge how changes in grid size affect detection accuracy and computational time.

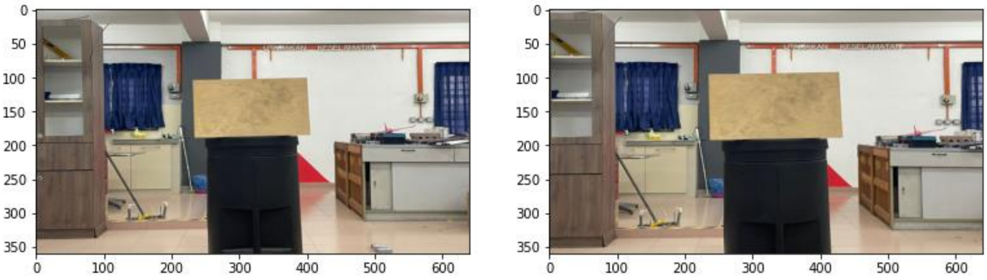
Based on the Appendices A1, same as case 1, the detection starts with grid 6x8, but it detected 3 regions as free obstacle regions. Its logic to be different value because the less-complex environment. 2 regions were side by side on left side of obstacle and 1 region was alone on the right side. For grid 8x12, it still detects the same regions but had 1 extra 1 detected grid which is on ceiling (2nd trial) and at curtain (3rd trial). It shows that every time the algorithm read an image, it can be many results but not meant by having big different, just small.

Grid 12x16 start to detects many areas, which were also true detection. But, the detection not precise on same area. This is because, the environment had textured background (pattern painted on wall and curtain). For reference, refer Figure 4.4 and Figure 4.5. Grid 16x24 still detects the same areas with grid 12x16 with more detection but still on the same area. Refer to Appendices A1, grid 24x32, it started to detect ceiling. Because of the obstacle is textured, there were no detection on the obstacle.

Based on graph 4.2, As the grid size increases from 3x4 to 24x32, there is a clear upward trend in both the percentage of detection and computational time. The smallest grid size, 3x4, begins with a detection rate of 0%, which indicates that no obstacles were detected at this resolution. This trend continues for the 4x6 grid size. However, as we transition to a 6x8 grid size, the obstacle detection rate starts to rise, reaching 6%, and then gradually increases with each subsequent grid size, culminating at 16% for the largest grid size of 24x32. The detection rate appears to grow almost linearly with grid size, suggesting that larger grids, which provide higher resolution, are more effective at identifying obstacles.

Simultaneously, the computational time required to process the images also increases with grid size. The time begins at 2.26 seconds for the 3x4 grid and climbs steadily to 55.11 seconds for the 24x32 grid. This increase is nonlinear, with the curve steepening as the grid size grows, reflecting a more than proportional rise in computational demand for larger grid resolutions. This pattern indicates that while larger grid sizes can improve detection accuracy, they do so at the cost of significantly greater processing time.

3.3 Case 3



(a) **(b)**

Fig. 12 *Case 3 (a) image 1; (b) image 2*

Table 4 *Data for case 3*

Grid Size	Total Number of Grid	Number of Detection	Number of True Detection	Percentage of Detection	Percentage of True Detection	Computational Time	False Positive Rate
3x4	12	0	0	0%	-	2.43	-
4x6	24	0	0	0%	-	3.42	-
6x8	48	0	0	0%	-	4.99	-
8x12	96	1	1	1%	100%	9.21	0%
12x16	192	10	6	3%	60%	14.64	40%
16x24	384	35	30	8%	86%	27.92	14%
24x32	768	133	111	14%	83%	73.45	17%

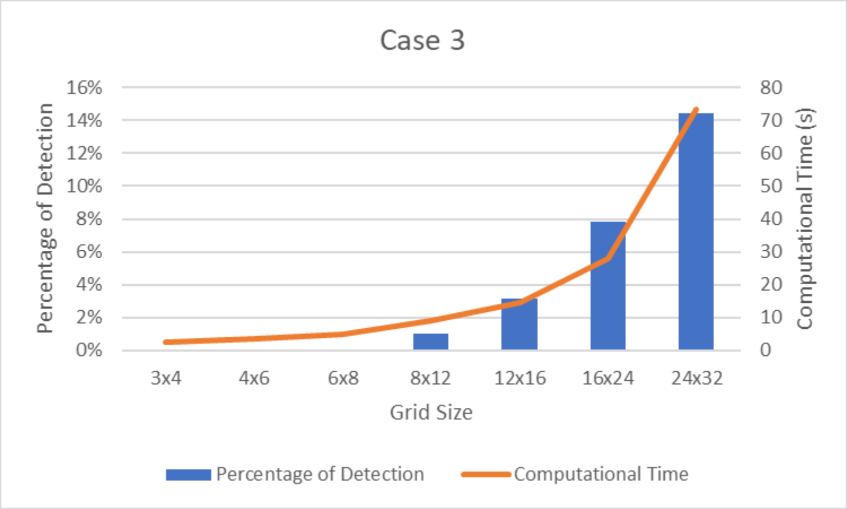


Fig. 13 *Data for case 3*

In Case 3, the study turns its focus to the challenge of detecting texture-less obstacles in complex environments, examining the impact of various grid sizes on detection accuracy, computational time, and particularly the occurrence of false positive rates. This scenario represents a critical area of concern in autonomous navigation, as texture-less obstacles do not provide the distinct visual cues typically relied upon for obstacle detection, necessitating a reliance on shape, size, or other contextual clues.

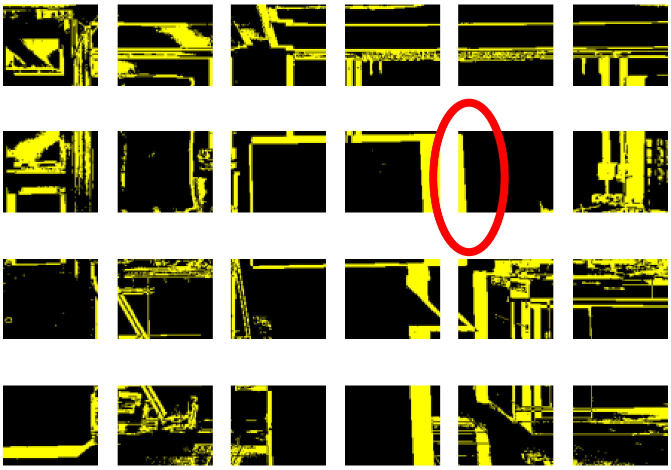
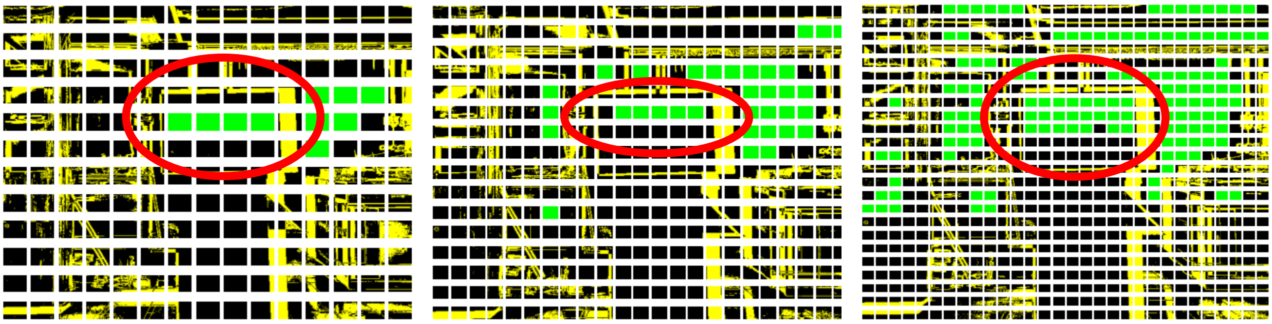


Fig. 14 *Case 3 - grid 4x6 (2nd trial)*

The detection of free region started on grid 8x12. Because it lates to start detecting the free region eventhough having same environment with case 1, it doesn't mean that the detection on texture object better than texture-less. It caused by the angle of the camera or the size of the obstacle. Refers to figure 4.8, there were just little bit of obstacle in the image. Because of that, it not detected as free region.

Next, the detection started on grid 8x12 with 1 detection. Refer figure 4.9, for grid 12x16, like case 1 (same environment), it still detected the same area, but it came with detection on obstacle. 4 regions were detected on obstacle ang make the false positive rate up to 40%. Because its texture-less obstacle, there were only having the frames as texture that can be detect by computer vision and the algorithm of pixel expansion.



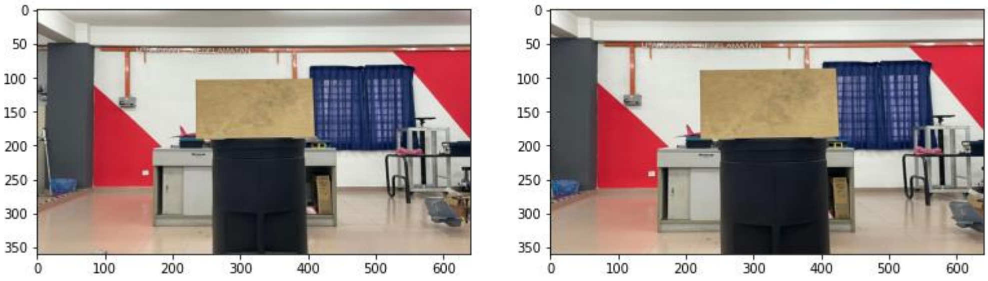
(a) (b) (c)

Fig. 15 *Case 3 (a) Grid 12x16; (b) Grid 16x24; (c) Grid 24x32*

Refer figure 4.10, for grid 16x24, it increased the detection up to 35 regions but still with 4 false detections. It made the false positive rate to 14%. For grid 24x32, it started to detect small area and ceilings. The detection on obstacle increased too. 22 regions of obstacle were chosen as free region and made the false positive rate to 17%. This increase could be attributed to several factors, such as the inherent difficulty for algorithms to detect accurately without distinct visual features, especially against the texture-less object.

Refers to graph 4.3, the computational time follows an ascending curve as grid sizes increase, starting from 2.43 seconds for the smallest grid and steepening considerably to 73.45 seconds for the largest grid size. This steep increase indicates that the processing demand is significantly higher for larger grids, which is a critical factor to consider for real-time applications where decisions need to be made quickly.

3.4 Case 4



(a) (b)

Fig. 16 *Case 4 (a) image 1; (b) image 2*

Table 5 *Data for case 4*

Grid Size	Total Number of Grid	Number of Detection	Number of True Detection	Percentage of Detection	Percentage of True Detection	Computational Time (s)	False Positive Rate
3x4	12	0	0	0%	-	2.83	-
4x6	24	0	0	0%	-	3.66	-
6x8	48	0	0	0%	-	4.31	-
8x12	96	1	1	1%	100%	7.14	0%
12x16	192	7	4	2%	57%	15.52	43%
16x24	384	35	30	8%	86%	27.496	14%
24x32	768	125	101	13%	81%	72.53	19%

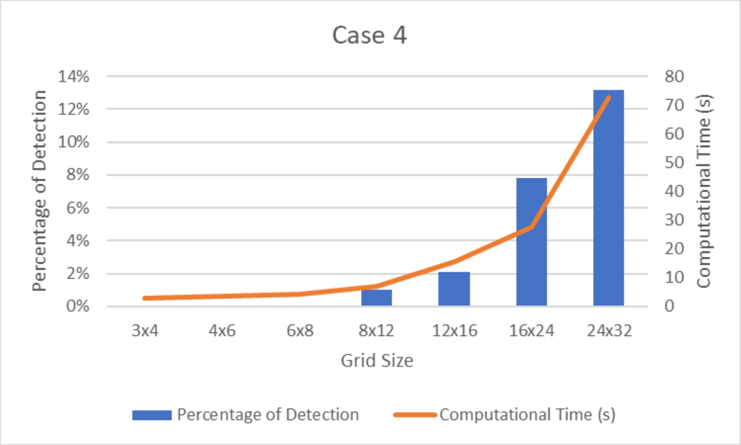


Fig. 17 *Data for case 4*

In Case 4, the study focuses on detecting textured obstacles within a less-complex environment and investigates the impact of various grid sizes on detection accuracy, computational time, and the occurrence of false positive rates. This case is particularly important for understanding the nuances of obstacle detection in environments that may not have the complexities of the earlier cases but still present unique challenges.

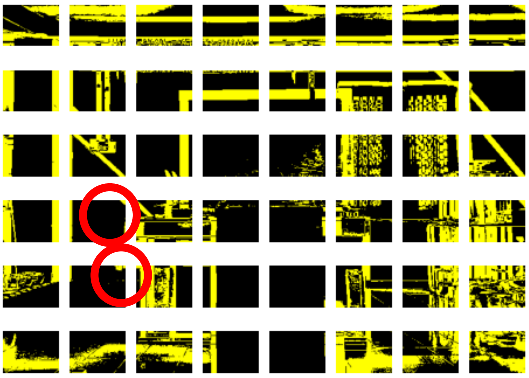


Fig. 18 *Case 4 - grid 6x8*

Same as case 3, because of the angle of the camera, the little bit of obstacle being in the free region grid. So, the detection started to detect the free region on grid 8x12 with 1 detection. On case 2, the 1st detection was in grid 6x8. And it does not mean that the case 2 is better detection variable. It all about the angle and position of the camera.

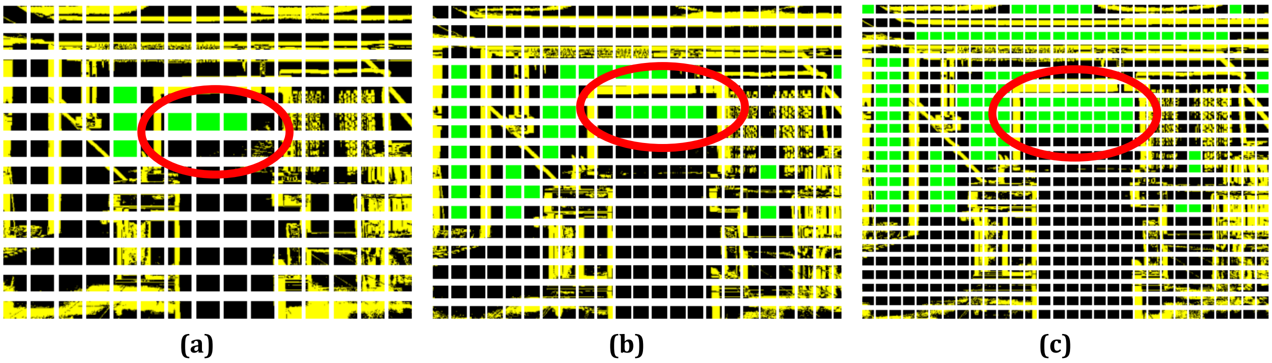


Fig. 19 *Case 4 (a) Grid 12x16; (b) Grid 16x24; (c) Grid 24x32*

Because it used texture-less obstacle like case 3, the grid 12x16 of case 4 also started to detect the obstacle as free region because there was no any texture on the center of the obstacle. It made the false positive rate up to 43% with 3 false detections. On grid 16x24, the false positive rate decreased because the number of detected free region increase, 26 regions, but the false detection just increased by 2 regions. For grid 24x32, it started to detect ceiling and any small space that is 100% black pixel in a grid as free region.

3.5 Findings

Based on all of the cases, there were many findings that can point out from the result.

First, The effectiveness of free obstacle detection on every experiment. The experiments were repeated 3 time for each grid on every case. Natural variability exists in almost all processes and systems. This is especially critical in systems where external factors can influence the results significantly. Repeated measurements increase the reliability of the results. If all the repetitions produce similar outcomes, it shows that the result is accurate. For the computational time, the data has a difference around 0.5s for bigger grid and around 30s for smaller grid. But the data still can be use because it does not interfere too much with the data that needs to be discuss. Based on Appendices A1, the data shows every result that came from repetitive experiment. The detection of every repetitive experiment came out with same value doesn't count either the frames of image were divided by small grids or big grids.

Second, the detections of free region on texture and texture-less object. Textured obstacles were more consistently and accurately identified across various grid sizes. The inherent patterns in textured obstacles provided the necessary visual cues for the image processing algorithms to effectively differentiate between obstacles and free regions. This was evident in both complex and less complex environments, although the detection rates were higher in less-complex environments. The algorithms used in the study, which include image subtraction and pixel expansion cues, were particularly adept at identifying areas where textured obstacles were present. The pixel expansion technique was able to amplify the textural cues, ensuring that the obstacle's presence was not misclassified as a free region. Therefore, textured obstacles resulted in a more reliable performance from the system, ensuring safety by correctly identifying regions where the vehicle should not traverse.

Next, texture-less obstacles posed a significant challenge for the detection system. The lack of distinct textural information on texture-less obstacle led to higher false positive rates, as the algorithm sometimes misidentified these obstacles as free regions. This was especially problematic in complex environments, where the higher density of grid elements in larger grid sizes did not necessarily translate to better detection. Instead, the system was prone to misinterpret texture-less surfaces, especially when they contained uniform colours or minimal contrast with their backgrounds. For texture-less objects, computer vision techniques often rely more heavily on shape, contour, and boundary cues, as well as reflectance properties and shading information.In experiment case 3 and case 4, texture-less obstacles with a significant amount of 100%-black-pixel grids were incorrectly detected as free regions. eventhough it on the center of obstacle. This misclassification presents a potential hazard for autonomous navigation, as it could lead to the system not recognizing an obstacle, thus compromising safety. It is because there was no any texture, contour or shadow on the obstacle.

Third, the detections of free region on type of environment. In complex environments, the research revealed that larger grid sizes contribute to more effective obstacle detection. This observation is supported by the data from Case 1, which shows a positive correlation between grid size and the percentage of detection. The intricate interplay of varied textures, lighting conditions, and the diversity of object shapes in complex settings necessitates finer grids. These detailed grids are essential to capture the nuanced visual information needed for accurate detection. Such environments often contain a plethora of visual information that can confuse detection algorithms if the grids are too coarse. Therefore, larger grids are preferred as they provide a more detailed analysis of the visual scene, allowing the detection system to distinguish between actual obstacles and free regions more effectively.

Futhermore, in less-complex environments, the study's outcomes, particularly from Case 2, indicate that obstacle detection systems can afford to use less granular grids while maintaining a high accuracy level. Despite a less complicated visual scene, the trend of improved detection with increased grid sizes remains apparent. The simplicity of the environment lessens the demand for high-resolution grids, which can help in optimizing computational efficiency without a substantial loss in detection accuracy. In these scenarios, the absence of environmental noise, such as complex textures or varying shadows, allows for the use of smaller grids, easing the computational load and enabling quicker processing times. This rapid processing is advantageous for real-time decision-making, which is critical for autonomous vehicle navigation.

In both complex and less-complex environments, there is an evident increase in detection accuracy with larger grid sizes. However, this increase comes at the computational cost, which can be a critical factor in real-time applications. The implication is that there exists an optimal grid size for different environmental contexts that effectively balances the trinity of detection accuracy, computational efficiency, and the minimization of false positives. This optimal grid size is variable and contingent upon the specific characteristics of the environment and the types of obstacles present. Therefore, adaptive grid sizing could be a significant advancement for autonomous vehicle systems, where the algorithm dynamically adjusts the grid resolution in response to the environmental complexity to ensure both efficient and accurate detection.

Forth, computational time. The findings reveal several critical aspects related to computational time. Firstly, the grid sizes increase, computational time consistently rises. This pattern contradicts the initial assumption that smaller grids might demand more processing due to a higher number of individual grid elements. The increase in computational time is non-linear, with a more pronounced escalation in processing demands as grid sizes become larger.

The study underscores the influential role of object texture and environment complexity on computational time. In cases involving texture-less objects, additional processing is required to differentiate these objects from their surroundings, contributing to increased computational time. Consequently, the computational time factor becomes crucial in real-time applications, such as autonomous vehicle navigation, where timely obstacle detection and response are paramount.

Fifth, False positive rate. One significant finding was the higher occurrence of false positives in cases involving texture-less objects. These objects, lacking distinct visual cues, were often misclassified as free regions, posing potential safety risks. This phenomenon emphasized the need for improved detection algorithms capable of addressing the unique challenges presented by texture-less surfaces.

The changes in grid size did not consistently impact the false positive rate. Like in case 3 and case 4, the false detection occurs on the same place of the obstacle. Therefore, the rate was decreased because the rate calculated with total number of grids. This observation suggests that factors beyond grid size, such as the complexity of the environment and the sensitivity of the algorithms, played crucial roles in false positive occurrences. It further underscores the importance of developing algorithms that can adapt to varying environmental conditions and object textures to avoid false positives effectively.

Sixth, percentage of detection. As the grid size decreases, leading to a higher total number of individual grids, the percentage of detection increases. This suggests that smaller grids, with their finer resolution, are more adept at capturing detailed environmental information, thereby identifying more free regions. However, this higher detection rate with smaller grids comes with a challenge in consistency and precision across different areas. Factors such as the environment's complexity and the presence of textures influence the detection's accuracy. In complex or textured environments, smaller grids may not consistently yield precise detections, highlighting the need for adaptable algorithms that can adjust grid sizes and processing techniques based on the specific environmental context.

Seventh, which was the best grid size that can be used? The choice of the best grid size also relied on the specific application and functional requirements of the drone. In environments with varied textures and complexities or the size of the environment, a one-size-fits-all approach to grid sizing is ineffective. For instance, drones operating in complex urban environments might require different grid sizing compared to those in more uniform and open landscapes. The study's findings suggest that while smaller grids are more effective in detailed environmental detection, they may not always be the practical choice due to computational constraints. In addition, larger grids, though less demanding computationally, might compromise detection precision and increase false positives. Therefore, the grid size must be tailored to the drone's operational environment, balancing the need for detailed detection against the limitations of processing capacity and real-time responsiveness.

Based on the experimental outcomes, a grid size of 16x24 emerges as a practical compromise for a range of scenarios. This grid size represents a middle ground, offering a balance between computational efficiency and detection accuracy. It is large enough to keep computational time within manageable limits while still providing sufficient resolution to detect obstacles accurately. This grid size also shows a lower propensity for false positives, making it a reliable choice for various applications. However, it's important to note that while range 12x16 to 16x24 may be effective in many situations, there is no universally optimal grid size. The specific environmental conditions, types of obstacles, and the intended function of the drone must always be considered when determining the most appropriate grid size for obstacle detection. So, a one-size-fits-all approach to grid sizing is ineffective.

4. Conclusion

The comprehensive study presented in the chapters offers valuable insights into the dynamics of grid size selection for obstacle detection in autonomous vehicle systems. The research thoroughly evaluates the effectiveness of various grid sizes in detecting free obstacle regions across different environments and obstacle textures. Key findings reveal that the efficiency of obstacle detection varies significantly with grid size, object texture, and environmental complexity. Larger grid sizes enhance detection accuracy in complex environments, capturing more detailed visual information essential for precise obstacle detection. However, this comes at the cost of increased computational time, which is a critical factor in real-time applications like autonomous navigation. Conversely, smaller grids, while beneficial in less-complex environments and more efficient in terms of computational resources, may lack the necessary precision in complex settings. The study also highlights the challenges in detecting texture-less objects, where false positive rates are higher, underscoring the need for more sophisticated algorithms capable of effectively discerning obstacles in varied conditions.

Addressing the first objective of comparing the effectiveness of different grid sizes in detecting free obstacle regions, the study conclusively demonstrates that there is no one-size-fits-all solution. The grid size of 16x24 emerges as a practical choice, offering a balanced compromise between computational efficiency and detection accuracy. This size is effective in various scenarios, maintaining computational time within manageable limits while providing sufficient resolution for accurate obstacle detection. However, the optimal grid size is highly dependent on specific environmental conditions and the nature of the obstacles, indicating the necessity for adaptive grid sizing in autonomous systems.

In response to the second objective, investigating the factors that affect the result outcome from different sets of grid sizes, the study identifies several key influencers. The texture of the obstacle plays a crucial role, with textured obstacles being more consistently and accurately identified across various grid sizes. The complexity of the environment significantly impacts detection accuracy, with larger grids preferred in complex settings to capture more detailed visual information. Additionally, the study underscores the importance of considering computational time, particularly for real-time applications where rapid and accurate detection is essential. These findings suggest the need for more sophisticated and adaptable algorithms that can dynamically adjust to varying environmental conditions and object textures to optimize detection accuracy and efficiency.

Acknowledgement

This research was supported by Universiti Tun Hussein Onn Malaysia (UTHM) through GPPS (vot Q210) and Tier 1 (vot H920).

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Mohammad Anwaar Badrol, Muhammad Faiz Ramli;* ***data collection:*** *Mohammad Anwaar Badrol, Muhammad Faiz Ramli;* ***analysis and interpretation of results:*** *Mohammad Anwaar Badrol, Muhammad Faiz Ramli;* ***draft manuscript preparation:*** *Mohammad Anwaar Badrol, Muhammad Faiz Ramli. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] Thakur, S., Sharma, V., & Chhabra, A. (2014). A Review: Obstacle Tracking Using Image Segmentation. A Review: Obstacle Tracking Using Image Segmentation, 2(6), 1–8

[2] Badrloo, S., Varshosaz, M., Pirasteh, S., & Li, J. (2022). Image-Based Obstacle Detection Methods for the Safe Navigation of Unmanned Vehicles: A Review. In Remote Sensing (Vol. 14, Issue 15). MDPI. https://doi.org/10.3390/rs14153824

[3] Bin Abdul Aziz, M. W., & Bin Ramli, M. F. (2023). Monocular Camera Free Region Detection Method of Obstacle Avoidance Using Pixel Volume Expansion for Micro - Sized UAV. 8th International Conference on Software Engineering and Computer Systems, ICSECS 2023, 205–210. https://doi.org/10.1109/ICSECS58457.2023.10256298l

[4] Ramli, M. F. (2020). Development of Obstacle Detection and Avoidance System Based on Integration of Different Based-sized Unmanned Aerial Vehicle Using Cues From Expansion Of Feature Points and Direction of Flow Field Vectors.

[5] Hu, J. wen, Zheng, B. yin, Wang, C., Zhao, C. hui, Hou, X. lei, Pan, Q., & Xu, Z. (2020). A survey on multi-sensor fusion based obstacle detection for intelligent ground vehicles in off-road environments. In *Frontiers of Information Technology and Electronic Engineering* (Vol. 21, Issue 5, pp. 675–692). Zhejiang University. https://doi.org/10.1631/FITEE.1900518