

Article

Grid-Based Low Computation Image Processing Algorithm of Maritime Object Detection for Navigation Aids

Ho-Seok Jeon, Sung-Hyun Park  and Tae-Ho Im * 

Department of Information and Communication Engineering, Hoseo University,
Asan-si 31499, Republic of Korea; wjsghtjr33@naver.com (H.-S.J.); shpark@hoseo.edu (S.-H.P.)

* Correspondence: taehoim@hoseo.edu; Tel.: +82-540-9642

Abstract: Several cameras are mounted on navigation aid buoys and these cameras can be used for accident prevention systems by processing the images captured. The currently existing image processing algorithms were originally designed for accident prevention on land—for example, CCTV (closed-circuit television)—which are performance oriented. However, when it comes to ocean-based images, navigation aids are usually located at sea and the cameras must be battery operated, and consequently, the energy efficiency of image processing is a major concern. Therefore, this paper proposed a novel approach to the detection of images in an ocean environment with a significantly lower computation. The new algorithm clustered pixels to grids and dealt with grids using greyscale rather than the particular color values of each pixel. Simulation-based experiments demonstrated that the grid-based algorithm provided five-times faster image processing in order to detect an object and achieved an up to 2.5 higher detection rate when compared with existing algorithms using ocean images.

Keywords: maritime object detection; pixel clustering; grid-based image processing; navigation aids; low power image processing



Citation: Jeon, H.-S.; Park, S.-H.; Im, T.-H. Grid-Based Low Computation Image Processing Algorithm of Maritime Object Detection for Navigation Aids. *Electronics* **2023**, *12*, 2002. <https://doi.org/10.3390/electronics12092002>

Academic Editor: Boris Andrievsky

Received: 4 April 2023

Revised: 21 April 2023

Accepted: 24 April 2023

Published: 26 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As advanced IT technologies were utilized within various industries, automatic data collection, analysis, and management became possible, and some parts of the infrastructure of these industries were replaced by smart systems [1]. The maritime industry also actively applies advanced technologies; for example, AI technologies are applied and developed in the autonomous operation of maritime transport, which accounts for more than 80% of global trade [2]. Navigation aids which were regarded as signposts for maritime transportation are also combined with digital technologies [3], providing various functions such as collecting marine data or transmitting safety information to vessels. For example, navigation aids can be used to monitor and provide comprehensive weather and ocean environmental information using a variety of IT equipment, which can help the fishing industry or can guide a better route option to maritime vessels. However, accidents with navigation aids frequently occur and primarily IUU (illegal, unreported and unregulated) fishing boats cause the accidents [4]. Damaged navigation aids are costly to repair [5], take a long time and cause inconvenience during the repair. This is a national loss, and therefore, object monitoring around the navigation aids is necessary to prevent accidents. While existing AIS (automatic identification systems) can identify objects around navigation aids, the location of IUU fishing boats may not be reported because they usually turn off the location equipment [6].

Another mean to identify objects around the navigation aids is using cameras on the navigation aid buoys. With AI technologies, the utility and the importance of image processing arise for accident detection and pre/post disaster assessment. Various image processing techniques are applied to images captured from cameras for disaster detection

and analysis. For example, [7] analyzed the massive rock and glacier ice collapse in India which occurred in 2021 and found the three primary causes of the tragic disaster. The satellite images, eyewitness videos and their image processing results primarily contributed to the post disaster assessment. Cameras on navigation aids extend to visually collect [8] and observe marine environment information such as fog, clouds, sea ice and green algae. In addition to environmental information, maritime information (for example, maritime vessels traffic) is also available, and therefore, cameras are an indispensable element of navigation aids, as shown in Figure 1. For example, NDBC (National Data Buoy Center) manages navigation aids using BuoyCAM2 cameras mounted on buoys, which captures 360 degrees every 5 min and transmits six different images on land every hour [8].

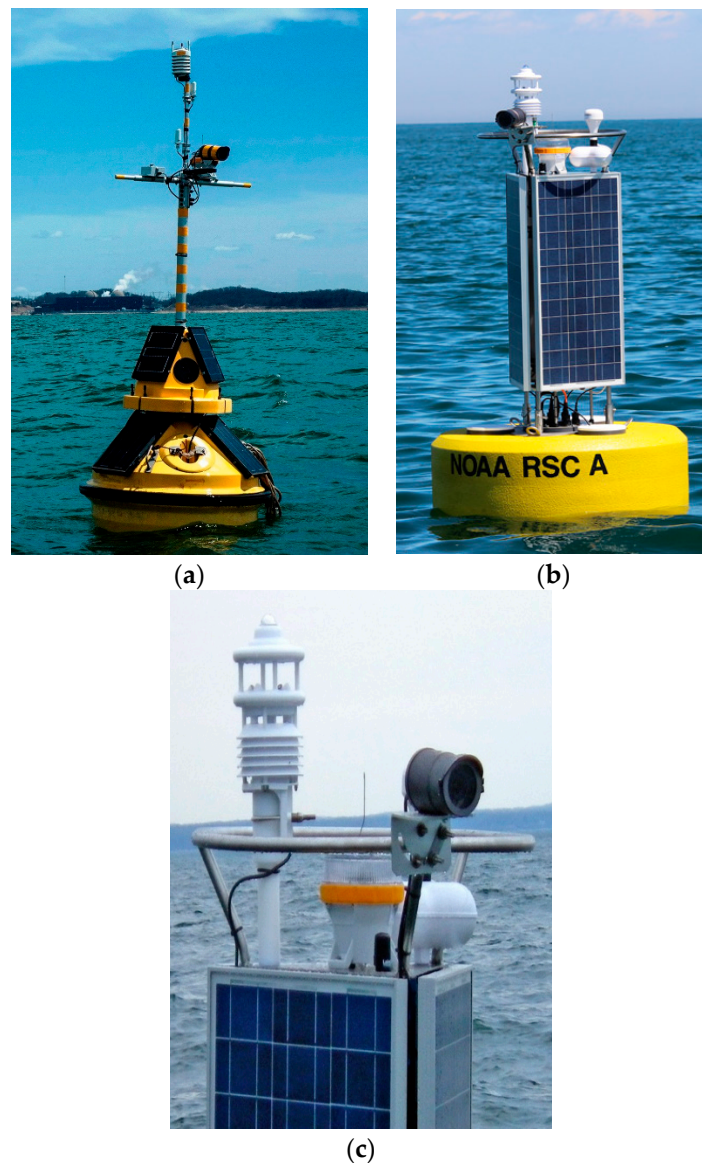


Figure 1. Different types of mounted cameras on buoys of NDBC [8]: (a) A video camera mounted on a weather buoy; (b) Coastal storms nearshore buoy system; (c) Close-up of the coastal storms nearshore buoy system.

Although the transmitted data are checked by the center on land, data are typically used to determine the cause and time of an accident after [9]. Therefore, there are demands to develop a navigation aids accident prevention system using cameras mounted on buoys. Although various method such as AIS, radar and infrared were also considered [10],

cameras are most beneficial because they provide visual information as well as affordable price [11].

Therefore, the scope of this paper is image processing from the visual information captured by cameras for maritime object detection. The object detection using visual information is highly affected by the amount of light in images. Consequently, the performance of object detection algorithms varies according to weather conditions. Therefore, image processing algorithms are typically designed to be specialized to a specific weather condition such as rain [12], intensive cloud [13], fog [14] and night time [15]. This paper focuses on the object detection within the context of a sunny weather conditions from sun-rise to sunset.

2. Literature Review

2.1. Three Types of Object Detection

Maritime object detection commonly consists of three steps: horizontal line detection, maritime object detection and AI applications. The purpose of line detection is to remove areas of the not-interested part based on the detected line [16]. This shortens the ROI (region of interest) in an image, which leads to reducing the complexity and the scope for the object detection [17]. Therefore, the performance of line detection algorithms is important for object detection results [18]. Canny edge detection [19] and Hough transform [20] are commonly used to detect horizontal lines. As Figure 2 shows, Canny edge detection detects contours in the marine image [19] and then, additional correction is performed using filters such as a Gaussian filter in order to remove contour lines which are recognized as noises in a horizontal line detection process. The cloud in the image is not recognized, since the color difference between sky and cloud is not well defined compared with the color difference between sky and ship or sea and sky. Therefore, the edge of cloud is not detected in the process of edge detection algorithms. Finally, among the remaining contours, one or more straight and transverse components are found through Hough transform. The result of Hough transform provides the two coordinates of the start and end points of each detected horizontal line. The straight line found through Hough transform is above an object in Figure 2, not on the actual horizontal line. This is the disadvantage of that the threshold values required for Canny edge detection need to be set manually. The Otsu [21] algorithm can be applied to compensate for this shortcoming and it sets the thresholds to appropriate intermediate values using pixel histogram of an image. By analyzing pixel histogram of an image, the optimized threshold value can be found, which minimizes the distributions of two classes (for example, object class and background class). Then, pixel values of the two classes are presented in a binarized image based on the threshold value. Otsu is relatively simple and effective, so that it is widely applied in image processing [22,23]. Muhuri et al. [22] identified the amount of snowfall of Himalayas in different seasons using SAR (synthetic aperture radar) images of the Himalayas. Otsu is used to detect and decide the snow-covered areas in the images and builds the snow map. With the Otsu algorithm, the study does not need to manually find a threshold, which allows accurate and efficient detection of snowfall in the images. Henley et al. [23] identified an object by using lights and shadows reflected from the hidden space behind the obstacle which hides the object. In this study, Otsu detected shadows and estimated the object by using the shadows. This enabled the separation of shadows and 3D reconfiguration by quickly finding the optimal threshold in images without complex computations.

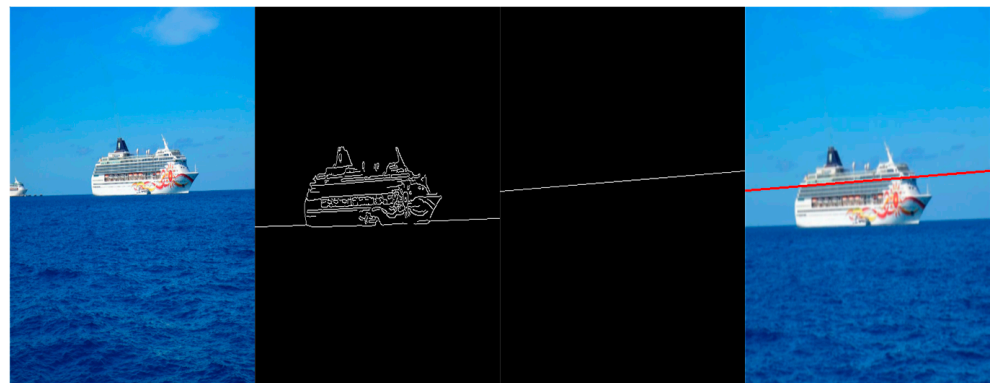


Figure 2. Strait transverse line detection process.

The next step was object detection, and background subtraction method [24] is commonly used. Although background subtraction is able to detect objects without the line detection step, it usually uses the line detection to remove the unnecessary part [25]. Background subtraction differentiates moving parts and constant parts in consequence images and assumes the constant part is a background. Background subtraction requires high computation volume and is difficult to apply if an object does not move [26].

Owing to the rapid development of deep learning image recognition technologies, there are various AI object detection studies for navigation aids using cameras [27–30]. High-performance image processing techniques using deep learning have demonstrated their effectiveness in various research; however, they require significant computation volume and power for it. Existing object detection studies are performance-oriented and they did not sufficiently consider the computation volume. For example, YOLO (You Only Look Once) which uses CNN (convolutional neural network)-based deep learning showed good performance in terms of detection accuracy and fast image processing at the same time [30,31]. However, the performance was measured using the high specification computers with GPUs (graphics processing units) for the deep learning computation. When YOLO was executed using a low performance imbedded board without GPUs, its detection ability dramatically dropped [32].

2.2. Comparison of GLC and Literature Review

Figure 3 summarizes Section 2.1 and includes the reference numbers discussed in the section. The first two steps were Canny edge detection and Hough transform. Hough transform detects one or more lines from the result of Canny edge detection. Next, the position of the detected line is provided to an object detection algorithm. By using the position, the algorithm tries to find an ROI of the object in an image and locates ROI to an AI algorithm. Finally, the AI algorithm identifies the object in the ROI and evaluates the accident risk of the object. If the AI algorithm judges that the object could lead to an accident with the navigation aid, it alerts the datacenter on land.

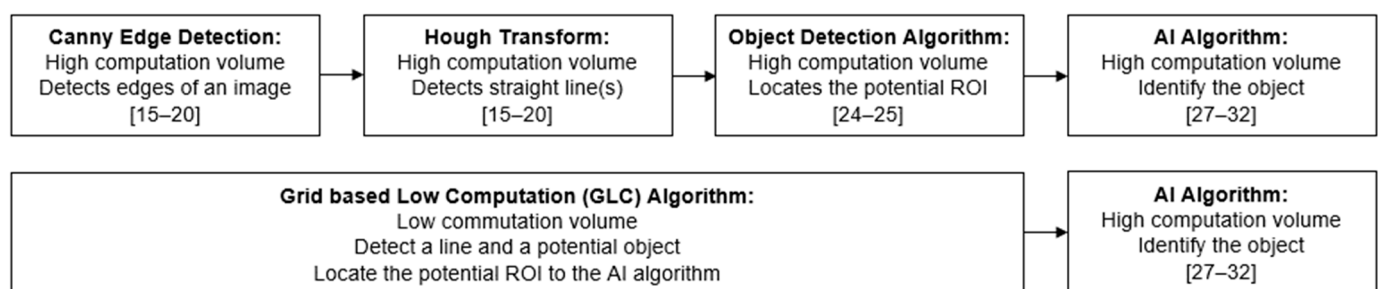


Figure 3. Comparison between GLC and the universal approach.

The literatures reviewed in Section 2.1 were originally designed for the object detection of land images. When compared to the operation of electronic devices through a continuous power supply on land, power supply at sea is significantly limited due to the characteristics of the marine environment. Navigation aids need to maintain the accident prevention system for a long time with limited energy such as solar energy or batteries. Therefore, it is difficult to apply existing high-performance algorithms [33] and so, low energy consumption-oriented research for the ocean object detection is necessary [34,35]. Therefore, this paper proposes and develops a novel image processing algorithm for accident prevention systems mounted on navigation aids. The algorithm is called grid-based low computation (GLC), which detects a maritime object using significantly lower computation. The ultimate goal of the GLC is to locate the ROI of an object in a marine image from the mounted cameras with low complexity. GLC uses a grid-based approach to reduce the computation volume and can also ignore potential errors caused by the moving background of the sea. GLC replaces the first three steps of the existing systems in Figure 3 but not the last step which does not necessarily operate all the time.

This paper consists of five sections. Section 3 explains the original design of the GLC algorithm including three sub-sections describing pixel clustering, greyscale conversion, horizontal line detection and maritime object detection. Section 4 demonstrates the performance of the GLC algorithm by comparing representative algorithms in terms of detection process time as well as detection accuracy. Section 5 provides the conclusion of this paper.

3. Grid-Based Low Computation Algorithm

The aim of the GLC algorithm is to reduce the volume of computation. Therefore, this paper proposes a new approach to simplify object detection processes as much as possible, but to the extent that locating the ROI is available. We propose pixel clustering and greyscale conversion in Section 3.1, simplifying line detection in 3.2, and detecting a maritime object from the reference line and, finally, GLC provides the ROI information.

3.1. Pixel Clustering and Greyscale

In most existing algorithms used to analyze images, the largest amount of computation is gained from reading every pixel value of an image [36,37]. Therefore, GLC converts the pixel-based images to grid-based. We name this process pixel clustering and Figure 4 shows an example of it. Clustering pixels is advantageous in that it reduces the operation of reading all pixels to only the number of grids and, therefore, requires less computing volume. Rather than detecting subtle changes in every single pixel, detecting changes in grids brings about better speed performance in object detection and ignores fine noises from within the moving background. Once an image is received from the cameras, GLC should resize it to 500×500 and then, pixel clustering is conducted using 25×25 grids. The number of grids is chosen by the experiments described in Section 4.3.

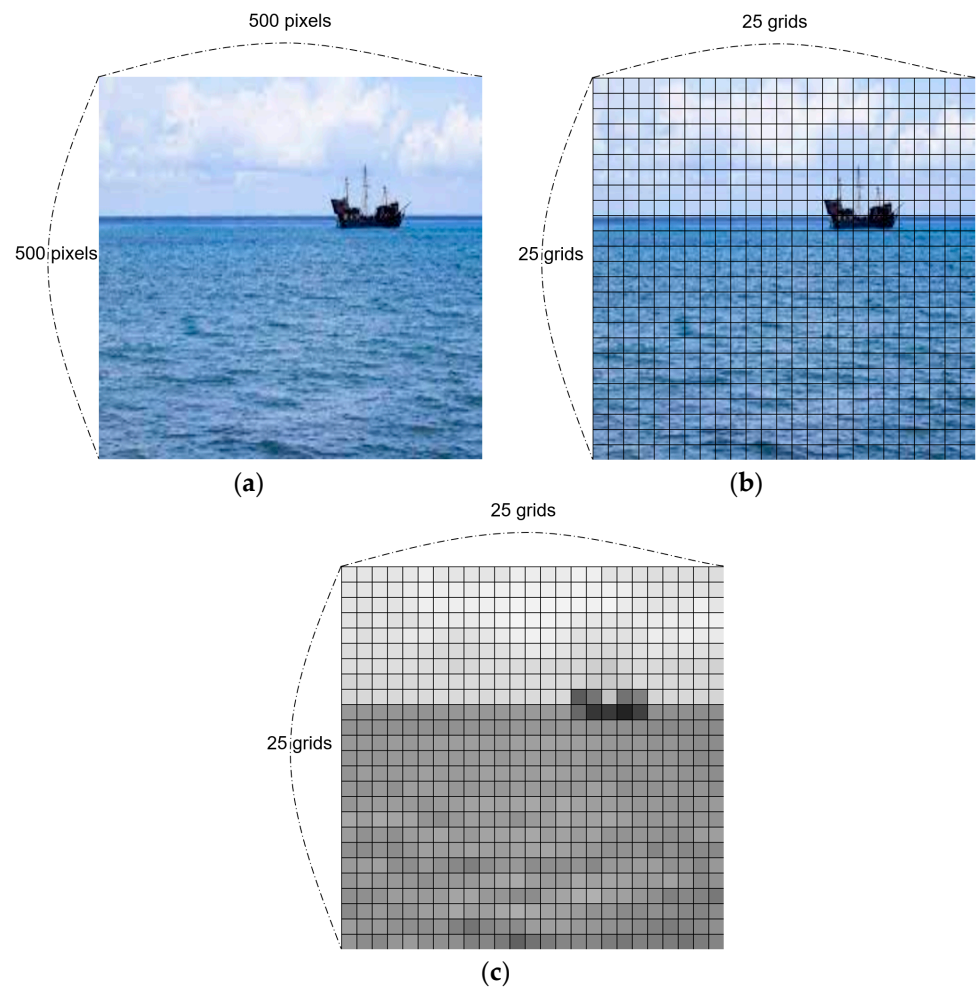


Figure 4. Grids by pixel clustering and greyscale converted image: (a) a resized 500×500 image from a camera; (b) the image after 25×25 pixel clustering; (c) greyscale converted image.

Figure 4a is the resized images and GLC clusters the pixels into grids, as shown in Figure 4b. Next, as shown in Figure 4c, GLC converts three color (i.e., green, red, blue)-based grids to greyscale based through Equation (1). The result of Equation (1) represents a single grid and it is used to compare with other grids in order to detect a horizontal line and an object.

$$\text{GreyScale_Mean} = (\text{Green} + \text{Red} + \text{Blue})/3 \quad (1)$$

Figure 5 illustrates a simple example of pixel clustering. If 4×4 color pixels are clustered into 2×2 grids, calculating the average number of pixels in a greyscale grid is performed using Equation (1). In the area highlighted in green in Figure 5, 218 is an average value of the four pixels ($221 + 220 + 216 + 217$), ignoring anything after the decimal point.

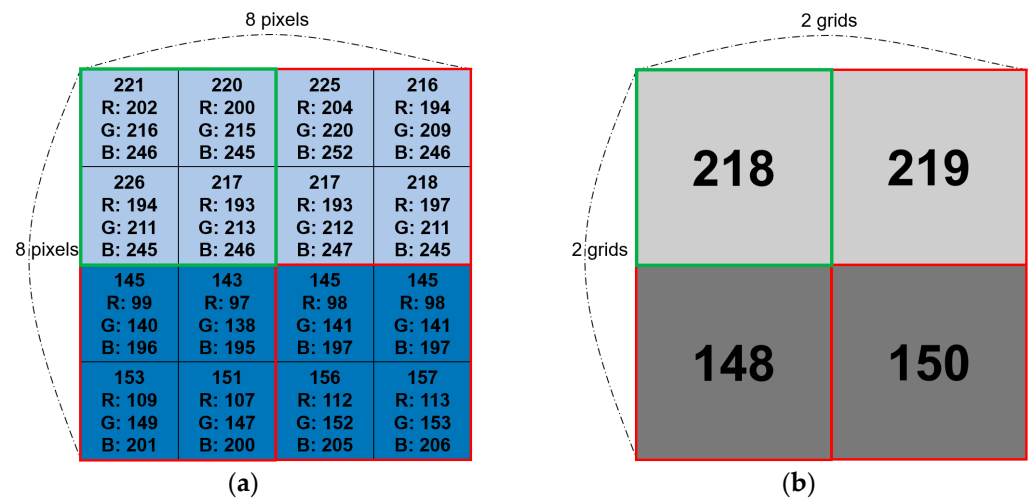


Figure 5. An example of pixel clustering and greyscale process of GLC algorithm: (a) 8×8 pixels before pixel clustering; (b) 2×2 grids after pixel clustering.

Here, it is helpful to mention that GLC is a novel approach: the difference between Figures 2 and 5 shows that GLC uses a totally different concept compared to existing algorithms.

3.2. Horizontal Line Detection

Horizontal line detection is essential to detect maritime objects. GLC approximates the position of the horizon using the greyscale value of grids as shown in Figure 6. After the process described in Section 3.1 (i.e., Figure 6a), the GLC first sums the greyscales of every X-axis grid and generates the average figure for it: for example, one row colored in red in Figure 6b. The grid value of the grids in the X-axis is merged, so that every row has one average value as shown in Figure 6c. The line having the highest variation is assumed to be the point between the sky and the sea and detected as a horizontal line. The average greyscale is quickly calculated because only simple integer values are used. As a result of Equation (2), the horizon colored in red in Figure 6c is detected, since GLP assumes that the point having the largest difference is the position of horizon [38]. The greyscale difference between sky and cloud is less distinct than the difference between sea and sky, so that GLC does not easily detect a horizon line in the sky area.

$$Y \text{ mean pixel difference} = |y_{i+1} - y_i| \quad (2)$$

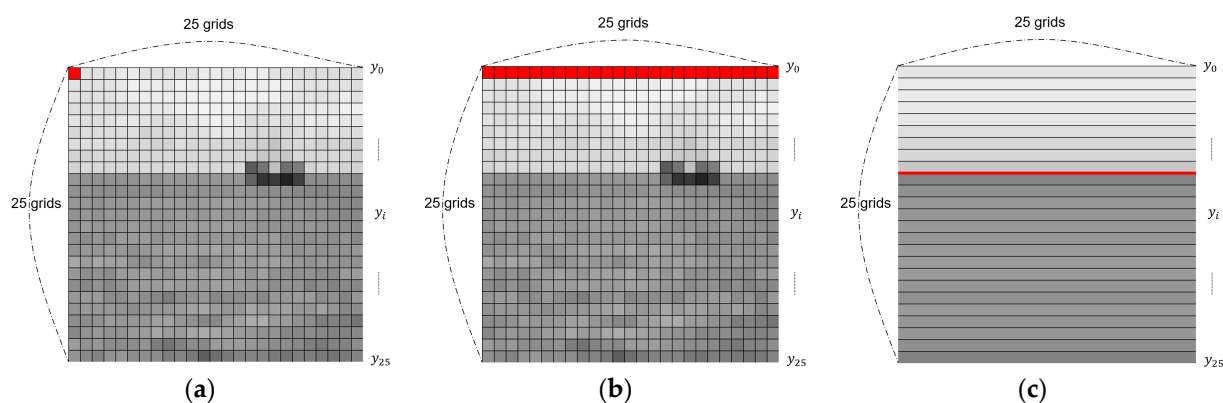


Figure 6. Grid-based horizon line detection process: (a) greyscale converted image; (b) searching the largest difference between two rows; (c) the detected line assumed as a horizon line.

3.3. Maritime Object Detection

For maritime object detection, this paper assumes that:

- GLC is able to detect a vessel when it floats on the horizontal line: this is a practical assumption since the speed of a maritime vessels is typically much slower than vehicles on land;
- Input images from mounted cameras must have a horizon line: the target application of the GLC algorithm is navigation aids accident prevention systems which are assumed to be located in sea and surrounded by horizon lines in 360 degrees.

The maritime object detection is conducted by searching through only two rows: the upper and lower rows of the horizontal line detected in Section 3.2, as shown in Figure 7. The manner of searching is similar in Section 3.2, only the direction of the search is different. In Figure 7, the difference of every adjacent grids in each row is calculated using Equation (3) and the results are added together. For example, the sum of Equation (3) results of the upper row was 87 and that of the lower row was 95. The sum of the lower row was larger than the upper row, which implies that the greyscale value changes in the lower row are more distinct. Therefore, GLC searches an object in the lower row and a point where the greyscale value rapidly changes could be noise or an object. The changes are random and relative; therefore, we refer to the experiment results and set the threshold value to 20.

$$X \text{ mean pixel difference} = |x_{i+1} - x_i| \quad (3)$$

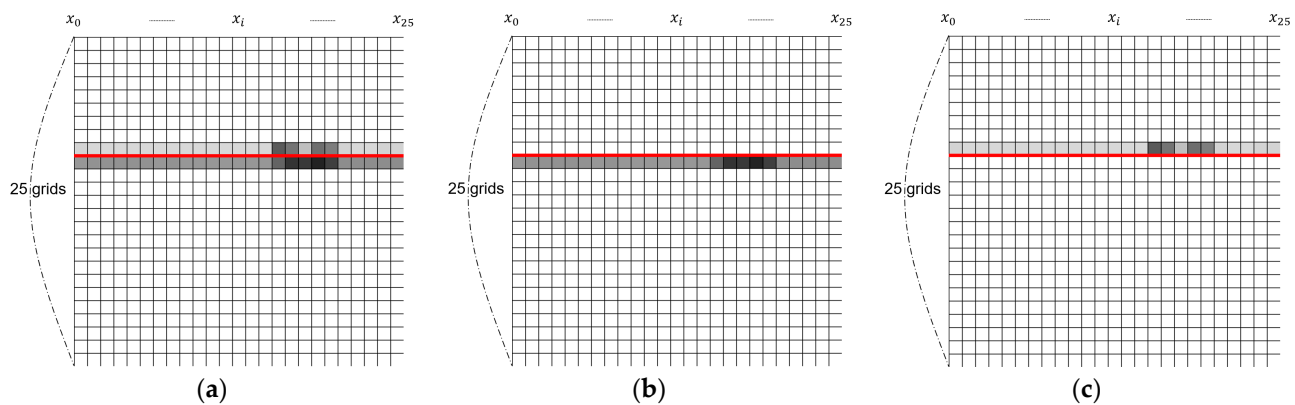


Figure 7. Object detection process in two rows: (a) adjacent rows of the reference line; (b) maximum mean pixel difference variation of the lower row: 95; (c) maximum mean pixel difference variation of the upper row: 87.

The three red vertical lines in Figure 8 indicate where the result of Equation (3) is larger than 20. The ROI is the part of the image and GLC informs an AI algorithm of it and then, the AI algorithm will identify what the object exactly is.

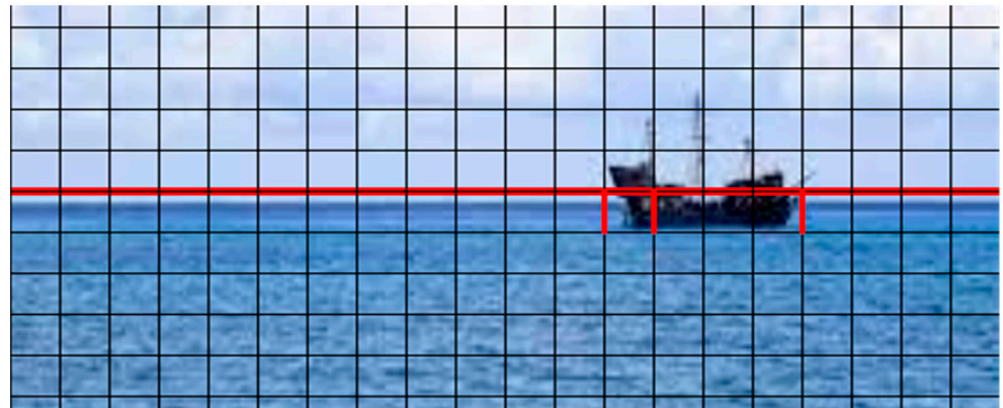


Figure 8. Object detection process of the GLC algorithm.

3.4. Overall Process Description

Figure 9 shows the overall process of GLC. Images from cameras were resized to an optimized size which is determined in each algorithm. Resizing is a typical process of image processing algorithms [39]. For GLC, images are resized to 500×500 pixels and resizing is conducted using the OPENCV 4.1 library which is commonly used in image processing algorithms [40,41]. Next, the image pixels are clustered into 25×25 grids. Optimizing the number of grids is comprehensively demonstrated in Section 4.3. Horizontal detection and object detection are conducted as explained in Sections 3.2 and 3.3. The final outputs of GLC are the (x,y) coordinates of the two end points of a detected vertical line. The number of detected vertical lines can be one or more. AI algorithms can use the information to identify the object and evaluate the accident risks.

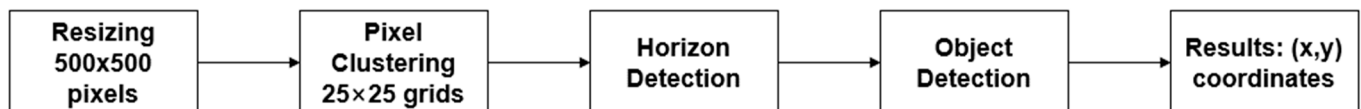


Figure 9. GLC overall process.

4. Experiments

4.1. Experiment Environments

The experiment environment was based on the OPENCV 4.1 library in the Raspbian Buster OS on the Raspberry-Pi4 embedded board. The test data set consisted of 100 images, including 65 images with an object and 35 images without an object, of which some examples are shown in Figure 10. The resolution of every image was 500×500 pixels.

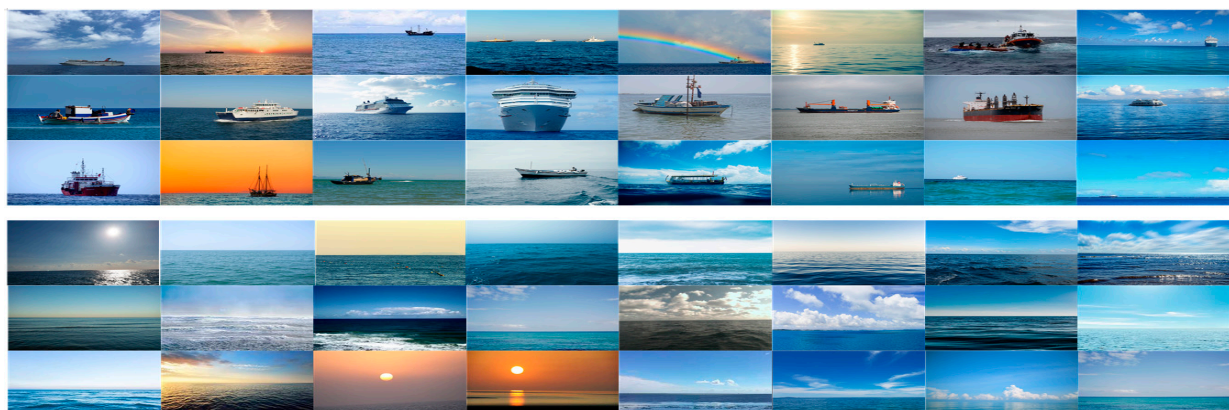


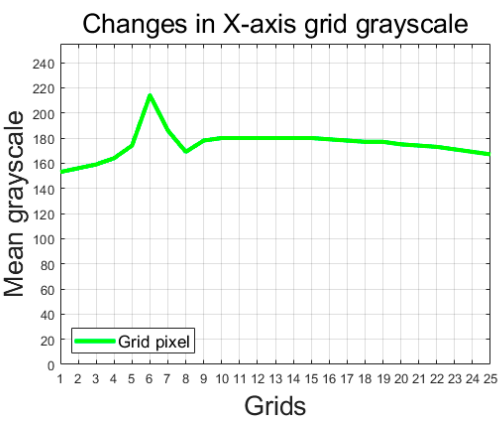
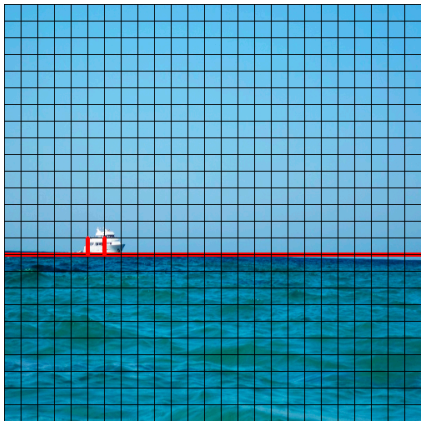
Figure 10. Examples of test data set.

4.2. Experiment Result Definitions

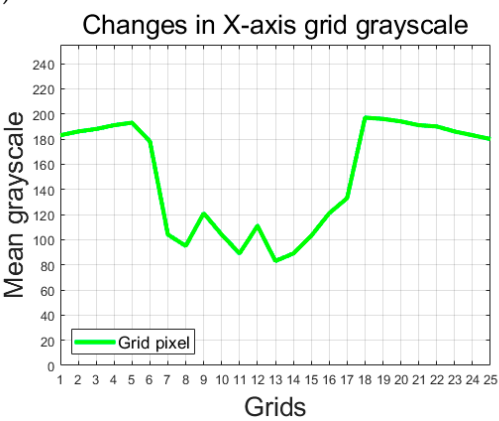
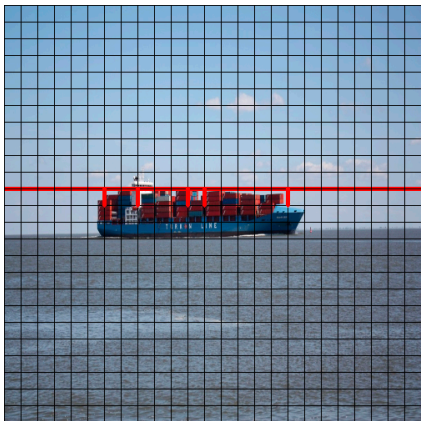
Figure 11 provides four different examples of the experiment results in 25×25 grids. Four graphs in Figure 11 show the greyscale changes of each image in a row using MATLAB. Four images in Figure 11 are greyscale in the process of GLC; however, this paper used color-based pictures for the reader’s better understanding. Table 1 summarizes the experiment result definitions.

Table 1. Experiment result definitions.

Horizon Decision		Object Decision
65 images having an object	Success: GLC decides a line within a grid where a horizon exists	Success: GLC decides one object where the object exists
	Success: GLC decides a line within a grid where an object exists	Fail: any other cases
	Fail: any other cases	Fail
35 images having no object	Success: GLC detects a line within a grid where a horizon exists	Success: GLC decides no object
		Fail: any other cases
	Fail: any other cases	Fail



(a)



(b)

Figure 11. Cont.

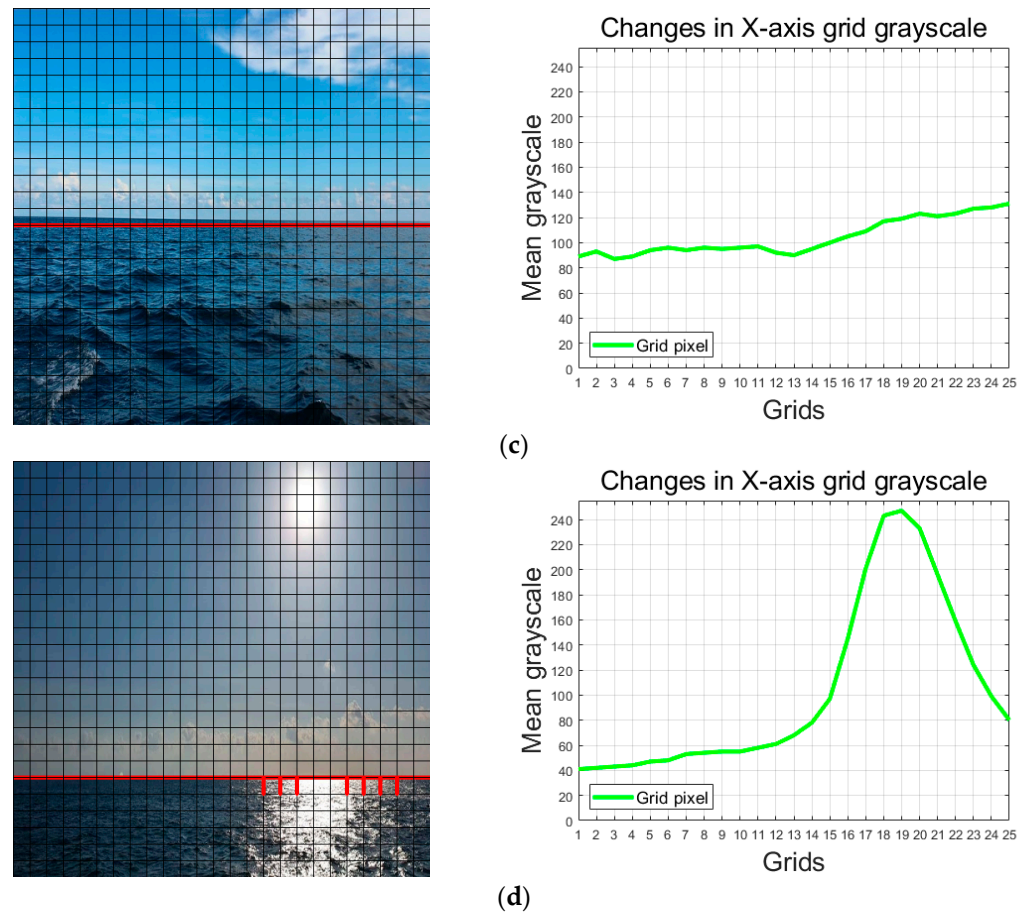


Figure 11. Set Examples of experiment results; (a) horizon decision success and object decision success; (b) horizon decision success and object decision success; (c) horizon decision success and object decision success; (d) horizon decision success and object decision fail.

- Figure 11a: there is an object in an image and GLC detects the horizon within a grid in which the actual horizon exists. GLC decides there is an object in the image because the result of Equation (3) is larger than 20, where the object exists. This paper defines this case as 'horizon decision success' and 'object decision success'.
- Figure 11b: there is an object in an image and GLC detects the horizon within the grid in which the object exists. This error occasionally occurs with GLC as well as existing algorithms when the object size or the hue contrast of the object is large. In this case, however, GLC still has the possibility of successfully detecting an object in an image. The aim of GLC is to locate ROI rather than finding an accurate coordinate of a line; therefore, this paper defines this case as 'horizon decision success'. GLC successfully detects the object where the object exists and so, the object decision of this example image is 'object decision success'.
- Figure 11c: there is no object in an image and the result of Equation (3) is always less than 20. Therefore, this paper defines this case as 'horizon decision success' and 'object decision success'.
- Figure 11d: there is no object in an image and GLC successfully detects a horizon. However, GLC decides there is an object because reflection on the surface makes noises. GLC makes a wrong decision and will misinform the AI algorithm. This paper defines this case as 'horizon decision success' and 'object decision fail'.

4.3. GLC Performanc Evaluataion

In order to evaluate the performance of GLC according to the number of grids, the different number of grids were applied, as shown in Figure 12.

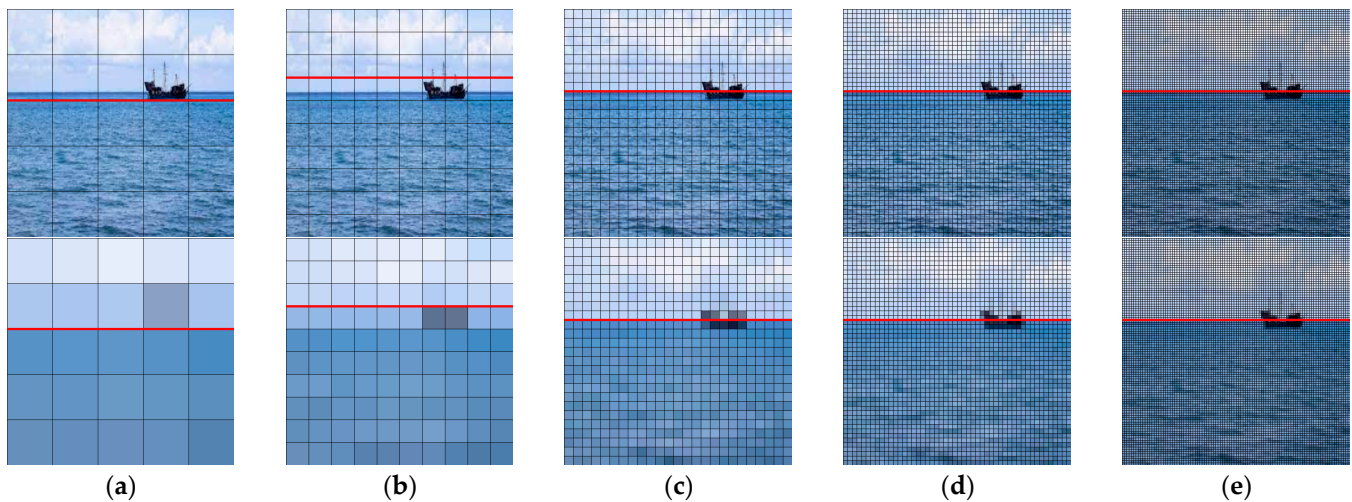


Figure 12. Performance comparison by the number of grids: (a) 5×5 grids; (b) 10×10 grids; (c) 25×25 grids; (d) 50×50 ; (e) 100×100 .

Figure 13 shows the results of detection performance of the GLC algorithm according to the number of grids. The two upper graphs in Figure 13 are the results of 65 images which had an object and two lower graphs are the results of 35 images which had no object. The horizon detection rate of GLC was steady according to the number of grids, which implies that the performance of GLC to detect horizon is reliable. ‘Horizon decision fail’ cases of GLC were primarily caused by light; for example, a sunset or light reflected on the surface. The reasons for horizon detection failures will be discussed in detail in Section 4.4.

In terms of object decision, as the number of grids is smaller, such as 5×5 , the size of the grid becomes larger, and thus, the boundary of a grid may be expressed vaguely, resulting in a low object detection rate. On the other hand, the higher the number of grids, the more precise detection is available, which is advantageous for small object detection; however, misdetection can occur due to ignorable noises. Therefore, the medium number of grids (25×25) shows the best performance of the detection rate.

Most cases of ‘object decision fail’ with 65 images occurred when an object was small because it was far away. However, in this case, the object was expected to be detected when the vessel comes closer to the camera, so it becomes an appropriate size to detect. Another reason for object decision fail was caused by light noise such as sunset or light reflected on the surface. Other noise types caused by wave bubbles, clouds, etc., are mostly temporary, so that they can be alleviated in a short time. In the case when the noises are nonepisodic, object detection failure can be a possibility.

The horizon detection processing speed of GLC is shown in Table 2. The processing speed results are the average values which represent process speed for one frame. As the number of grids increased, the processing speed showed a tiny increase. However, if the lower performance embedded processor than Raspberry-Pi4 is used, the processing time difference becomes larger.

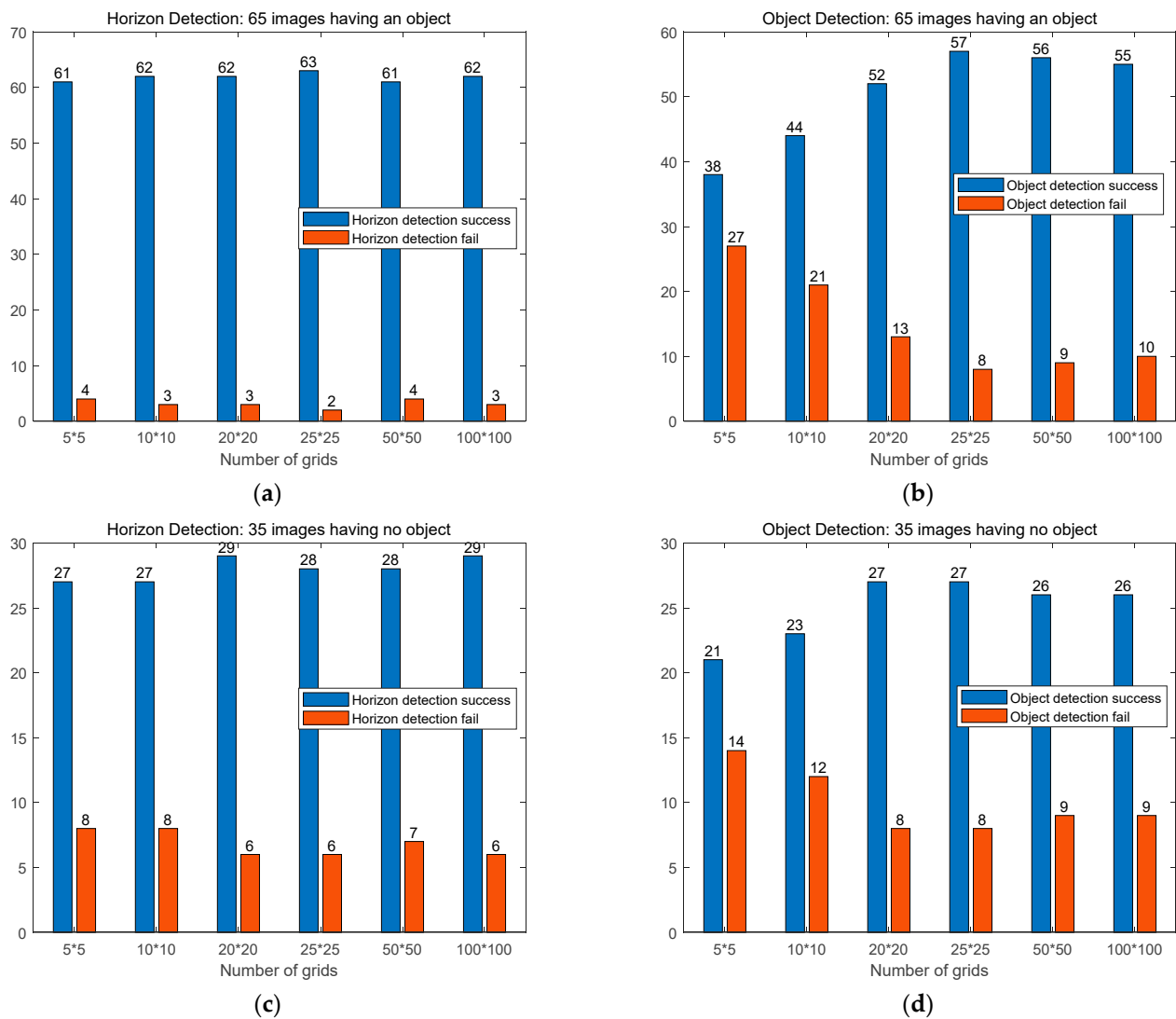


Figure 13. GLC performance comparison with different number of grids: (a) horizon detection results of GLC according to the different number of grids for 65 images having an object; (b) object detection results of GLC according to the different number of grids for 65 image having an object; (c) horizon detection results of GLC according to the different number of grids for 35 images having no object; (d) object detection results of GLC according to the different number of grids for 35 images having no object.

Table 2. GLC processing speed comparison.

The Number of Grids	Horizon Detecting Processing Time (1 Frame)
5 × 5 grids	8.5 ms
10 × 10 grids	9.0 ms
20 × 20 grids	9.3 ms
25 × 25 grids	9.4 ms
50 × 50 grids	9.5 ms
100 × 100 grids	11 ms

4.4. Horizon Detection Performance Comparison

In this section, GLC using 25 × 25 grids is compared with (1) Canny edge detection + Hough transform and (2) Canny edge detection + Hough transform + Otsu. Canny edge

detection and Hough transform are not able to detect an object in an image (please refer to Figure 3); therefore, only horizon detection performance is compared in this section. The same dataset (65 images with an object and 35 images without an object) is used for performance comparison.

For a fair performance comparison, the second horizon decision success case in Table 1 (i.e., Success: GLC decides a line within a grid where an object exists) is excluded in the criteria of success in this section because it is not a suitable criteria for Canny and Hough algorithms. Considering the algorithm features of Canny edge detection + Hough transform, although more than two horizontal lines are detected, if they exist within a grid where the horizon exists, it is defined as 'horizon decision success'.

A threshold pair of Canny edge detection is given to (150, 150), and the comparison result is shown in Figure 14. When Otsu is applied, Otsu chooses a different threshold pair for each image. GLC achieves up to 2.5 times higher object detection performance compared to other existing algorithms.

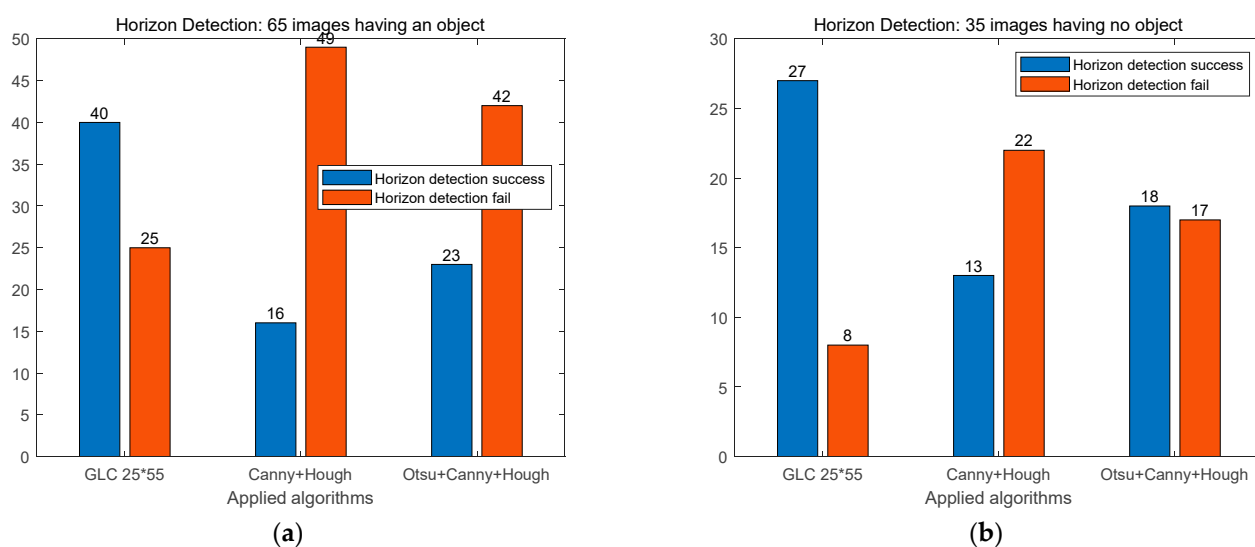


Figure 14. Horizontal line detection results of three different algorithms: GLC outperforms other algorithms in terms of the detection rate: (a) horizon detection results of three different algorithms when images have one object; (b) horizon detection results of three different algorithms when images have no object.

Table 3 describes the detailed reasons for horizon decision failures. The most common reason for other algorithms is that they cannot detect the horizon (Reason 3 in Table 3), which means the result of those algorithms does not have any horizontal line in an image. The reason for this failure is that the algorithms use a fixed threshold pair. To obtain the best performance from those algorithms, the available range of the two variables must be tried and then, the best pair can be found; however, this experiment fixes the pair to (150, 150) for the best processing speed time. This result demonstrates that these algorithms can provide an optimized performance for fixed images on land; however, these are not appropriate in the constantly changing environment. On the other hand, the GLC algorithm is designed to detect one line at the end; therefore, the failure due to no horizon is not something that occurs with GLC.

Table 3. Horizon detection fail reasons.

Reason of Horizon Detection Failure		25 × 25 Grids	Canny Hough	Canny Hough Otsu
With an object	R1: Horizon is detected within a grid where the object exists.	23	2	2
	R2: Horizon is detected within a grid where the object or horizon does not exist.	2	6	4
	R3: No horizon is detected	0	41	36
With no object	R2: Horizon is detected within a grid where the horizon does not exist.	8	0	3
	R3: No horizon is detected	0	22	14

The error that occurs when the horizon is detected on an object (i.e., Reason 1 in Table 3) occurs to every algorithm. For GLC, this error is the primary case because GLC simplifies the image process and the accuracy is lower than other algorithms. However, this case does not necessarily mean that the object decision is not available for GLC. Considering the purpose of GLC, this paper uses the images of Reason 1 for the object detection experiment. Reason 2 was caused by light noise in all algorithms.

Table 4 compares the processing speed of these existing algorithms and GLC. GLC achieves five times faster processing speed than the existing universal line detection algorithms. Canny edge detection and Hough transform are optimized for line detection, which provide more accurate detection of the horizon position than the GLC algorithm but have a slower processing speed due to the high volume of their computation. Although the Otsu algorithm can relieve the problem, the processing time performance GLC is still much faster, primarily owing to the unique grid-based approach by pixel clustering.

Table 4. Horizontal line detection speed result comparison.

Algorithm	Canny + Hough	Canny + Hough + Otsu	GLC
Time	51 ms	47 ms	8.5~11 ms

The experimental results vary depending on the processor performance. If they are compared on low-cost processors, the results will show a larger difference in speed. This implies GLC becomes more efficient when GLC is operated in a very limited and low-cost embedded system which is expected to consume extremely low amount of energy.

4.5. Object Detection Performance Comparison

This section compares the object detection performance of GLC having 25 × 25 grids to an object detection algorithm called CCL (connected component labeling) [42]. CCL aims to detect a maritime object for the purpose of accident prevention and is based on a horizon detection method. CCL is a color-based algorithm and it generates labels and tags them to pixels according to pixel colors and then groups the similar labels based on its pre-defined threshold value. CCL shows a good performance in the maritime object detection rate; however, it requires the high volume of computation.

Figure 15 shows an example of object detection using CCL. From the original image from a camera in Figure 15a, a horizon line is detected in Figure 15b, so the potential ROI in the image can be minimized based on the detected line, as Figure 15c shows. CCL can be applied to detect an object at this step and the object detection result is shown in a box format, as shown in Figure 15d.

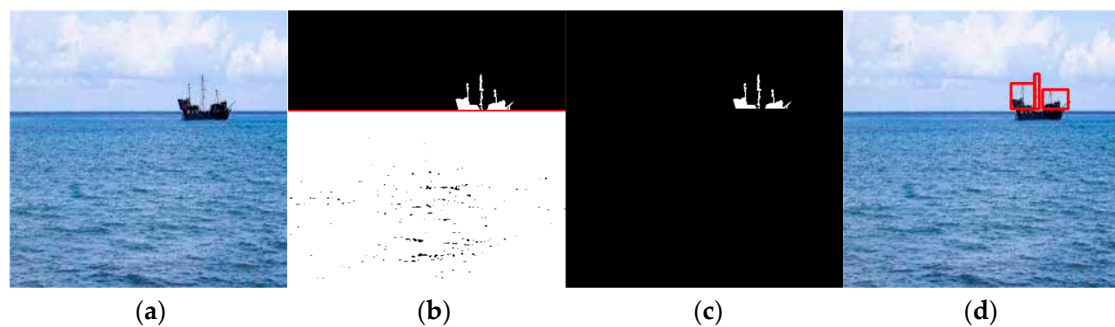


Figure 15. Example of CCL algorithm for the object detection: (a) the original image from a camera and it is resized; (b) Horizon detection result; (c) The process to detect an object based on the detected line; (d) Object detection result which is indicated in boxes.

For the experiment in this section, the horizon detection for CCL was conducted using Canny edge, Hough transform and Otsu, since this combination showed the best performance in Table 4. Table 5 compares the object detection experiment results. The number of images for which GLC successfully makes the object decision was greater than CCL. However, it was caused by the limitation of the horizon detection algorithms rather than the CCL object detection capability. GLC can detect a horizon line and an object at once; however, CCL was not involved in horizon detection. Therefore, it is not fair to state that GLC object detection performance was better according to the experiment results in Table 5.

Table 5. Object detection performance comparison.

	65 Images with an Object		35 Images with No Object	
	GLC	CE + HT + Otsu	GLC	CE + HT + Otsu
Horizon detection success	63 images	25 images	28 images	18 images
Object detection success	GLC	CCL	GLC	CCL
	57 images	18 images	27 images	11 images

In terms of object detection time, the average time after resizing an image to object decision was measured. GLC significantly outperformed CCL because of the large amount of time required for horizon detection before CCL, as Table 6 shows. CCL is based on the horizon line method, which means the algorithms must be applied after the horizon detection. Therefore, the line detection time is included in the result to fairly compare to GLC. Although setting apart the line detection time, the average object detection time of CCL for the 100 images was 2.7 times greater than the time GLC requires for both line and object detection. This result obviously demonstrates that GLC is an extremely energy effective algorithm by reducing the complexity of the detection processes using a novel grid-based method.

Table 6. Object detection speed result comparison.

Algorithm	Canny + Hough + Otsu + CCL	GLC (25 × 25 Grids)
Time	71 ms (47 ms: line detection 24 ms: object detection)	9 ms (line and object detection)

5. Conclusions

As navigation aids were digitalized, they developed beyond a signpost to provide various functions. However, once an accident occurs with navigation aids, it leads to a high cost in terms of time and expense because a human must visit the site to repair any damage and the navigation aid buoy itself is a very expensive piece of equipment. For accident prevention, existing image processing algorithms can be considered; however, they were originally designed for accidents on land. Therefore, such algorithms do not assume a continuously moving background. Moreover, since images recorded on land tend to include various objects, the existing algorithms aim to provide good performance in terms of accuracy, which brings about high computation volumes. Therefore, energy saving is not a major concern because the power supply is easy on land. However, navigation aids are usually located in the sea, often far from land, and the cameras are battery operated; therefore, energy efficiency is a critical challenge. Fortunately, the images recorded at sea tend to simply include a horizon line and an object at a moment, which does not require high accuracy, so there is the potential to reduce the computation volume. Therefore, this paper proposed the GLC algorithm which is optimized for ocean image processing. GLC aims for extremely low energy consumption and, therefore, GLC uses a new grid-based approach when compared to traditional image processing algorithms. GLC receives images from cameras, resizes them and clusters pixels to grids to handle the images by a larger size grid unit. Moreover, GLC uses greyscale rather than color values in order to reduce computation volume further. GLC detects a horizon line by comparing the sum of greyscale in each row. A line having the largest difference between two rows is assumed as the horizon line in an image. Using the line as a reference, GLC tries to find an object in grids which are adjacent to the reference line. If the value of a vertical line between two grids is greater than a defined threshold, GLC provides the coordinates of the two end points to AI algorithms for the further investigation. Simulation-based experiments demonstrated the efficiency of the GLC algorithm compared to universal horizontal line detection algorithms. The simulation used 100 typical ocean images and demonstrated that GLC provided five times faster image processing to detect a horizon line and up to 2.5 times higher detection rate than existing algorithms. The main contributions of this paper are listed below:

- This paper proposed a new image processing approach called GLC aiming for extremely low energy consumption for maritime object detection;
- The grid-based approach was optimized for maritime object detection since grids avoid errors caused by subtle changes in the moving background;
- This paper compared GLC with existing and well-known image processing algorithms and demonstrated that GLC significantly reduces the image processing time also can increase the image detection rate;
- Using the GLC algorithm, navigation aids can extend their functions to long-term accident prevention system using cameras mounted on them.

GLC intensively simplifies the object detection process. However, the ocean images are typically simpler than images on land so that simplified process can have sufficient abilities to detect maritime objects. Applying the GLC algorithm, the accident prevention systems are expected to be operation for a sufficient period of time.

Author Contributions: Conceptualization, T.-H.I.; methodology, H.-S.J.; software, H.-S.J.; validation, H.-S.J.; formal analysis, H.-S.J.; investigation, H.-S.J.; resources, H.-S.J.; data curation, H.-S.J.; writing—original draft preparation, H.-S.J.; writing—review and editing, S.-H.P.; visualization, S.-H.P.; supervision, S.-H.P.; project administration, T.-H.I.; funding acquisition, T.-H.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Korea Institute of Marine Science & Technology Promotion (KIMST) funded by the Ministry of Oceans and Fisheries grant number 202004482. This research was supported by Korea Institute of Marine Science & Technology Promotion (KIMST) funded by the Ministry of Oceans and Fisheries grant number 20210636. This research was supported by

the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program grant number IITP-2023-2018-0-01417 supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). The APC was funded by the ITRC support program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Research data is available at: https://drive.google.com/drive/folders/1ugffarDI46UiLyL0sfCcD1kpW-rR5UfA?fbclid=IwAR2GqwxGZ_CctryGS46hCKENDCe4vYYSQ2pYBQGtMMWptrYtqVAemNOH0VY, accessed on 21 April 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kirimtat, A.; Krejcar, O.; Kertesz, A.; Tasgetiren, M.F. Future trends and current state of smart city concepts: A survey. *IEEE Access* **2020**, *8*, 86448–86467. [\[CrossRef\]](#)
2. Forti, N.; d’Afflisio, E.; Braca, P.; Millefiori, L.M.; Carniel, S.; Willett, P. Next-Gen Intelligent Situational Awareness Systems for Maritime Surveillance and Autonomous Navigation. *Proc. IEEE* **2022**, *110*, 1532–1537. [\[CrossRef\]](#)
3. Babić, A.; Oreč, M.; Mišković, N. Developing the concept of multifunctional smart buoys. In Proceedings of the OCEANS 2021: San Diego—Porto, San Diego, CA, USA, 20–23 September 2021.
4. Ng, Y.; Pereira, J.M.; Garagic, D.; Tarokh, V. Robust Marine Buoy Placement for Ship Detection Using Dropout K-Means. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020.
5. Ramos, M.A.; Utne, I.B.; Mosleh, A. Collision avoidance on maritime autonomous surface ships: Operators’ tasks and human failure events. *Saf. Sci.* **2019**, *116*, 33–44. [\[CrossRef\]](#)
6. Raymond, B.; Christopher, H.; Robert, C.; Helmut, P. Counter-vandalism at NDBC. In Proceedings of the 2014 Oceans—St. John’s, St. John’s, NL, Canada, 14–19 September 2014.
7. Shugar, D.H.; Jacquemart, M.; Shean, M.; Bhushan, S.; Upadhyay, K.; Sattar, A.; Schwanghart, W.; McBride, S.; Van Wyk de Vries, M.; Mergili, M.; et al. A massive rock and ice avalanche caused the 2021 disaster at Chamoli, Indian Himalaya. *Science* **2021**, *373*, 300–306. [\[CrossRef\]](#)
8. O’Neil, K.; LeBlanc, L.; Vázquez, J. Eyes on the Ocean applying operational technology to enable science. In Proceedings of the OCEANS 2015—MTS/IEEE Washington, Washington, DC, USA, 19–22 September 2015.
9. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
10. Del Pizzo, S.; De Martino, A.; De Viti, G.; Testa, R.L.; De Angelis, G. IoT for buoy monitoring system. In Proceedings of the IEEE International Workshop on Metrology for Sea (MetroSea), Bari, Italy, 8–10 October 2018.
11. Prasad, D.K.; Rajan, D.; Rachmawati, L.; Rajabally, E.; Quek, C. Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1993–2016. [\[CrossRef\]](#)
12. Jingling, L.; Dongke, L. Ship target detection based on adverse meteorological conditions. In Proceedings of the Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 14–16 April 2022.
13. Meifang, Y.; Xin, N.; Ryan, W.L. Coarse-to-fine luminance estimation for low-light image enhancement in maritime video surveillance. In Proceedings of the Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019.
14. Yu, G.; Yuxu, L.; Ryan, W.L.; Lizheng, W.; Fenghua, Z. Heterogeneous twin dehazing network for visibility enhancement in maritime video surveillance. In Proceedings of the International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–25 September 2021.
15. Takumi, N.; Etsuro, S. A Preliminary Study on Obstacle Detection System for Night Navigation. In Proceedings of the IEEE/SICE International Symposium on System Integration (SII), Honolulu, HI, USA, 12–15 January 2020.
16. Petković, M.; Vujović, I.; Kuzmanić, I. An overview on horizon detection methods in maritime video surveillance. *Trans. Marit. Sci.* **2020**, *9*, 106–112. [\[CrossRef\]](#)
17. Gershikov, E.; Libe, T.; Kosolapov, S. Horizon line detection in marine images: Which method to choose? *Int. J. Adv. Intell. Syst.* **2013**, *6*, 79–88.
18. Hashmani, M.A.; Umair, M.; Rizvi, S.S.H.; Gilal, A.R. A survey on edge detection based recent marine horizon line detection methods and their applications. In Proceedings of the IEEE International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 29–30 January 2020.
19. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *8*, 679–698. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Hough, V.C. Method and Means for Recognizing Complex Pattern. U.S. Patent 3069654, 1962.
21. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [\[CrossRef\]](#)
22. Muhuri, A.; Ratha, D.; Bhattacharaya, A. Seasonal Snow Cover Change Detection Over the Indian Himalayas Using Pola-ri-metric SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *12*, 2340–2344. [\[CrossRef\]](#)

23. Henley, C.; Maeda, T.; Swedish, T.; Raskar, R. Imaging Behind Occluders Using Two-Bounce Light. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020.
24. Prasad, D.K.; Prasath, C.K.; Rajan, D.; Rachmawati, L.; Rajabally, E.; Quek, C. Object detection in a maritime environment: Performance evaluation of background subtraction methods. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 1787–1802. [\[CrossRef\]](#)
25. Prasad, D.K.; Dong, H.; Rajan, D.; Quek, C. Are object detection assessment criteria ready for maritime computer vision? *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 5295–5304. [\[CrossRef\]](#)
26. Fefilatyev, S.; Goldgof, D.; Shreve, M.; Lembke, C. Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. *Ocean Eng.* **2012**, *54*, 1–12. [\[CrossRef\]](#)
27. Zhenfeng, S.; Linggang, W.; Zhongyuan, W.; Wan, D.; Wenjing, W. Saliency-aware convolution neural network for ship detection in surveillance video. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 781–794.
28. Sung, W.M.; Jiwon, L.; Jungsoo, L.; Dowon, N.; Wonyoung, Y. A Comparative study on the maritime object detection performance of deep learning models. In Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 21–23 October 2020.
29. Safa, M.S.; Manisha, N.L.; Gnana, K.; Vidya, K.M. A review on object detection algorithms for ship detection. In Proceedings of the International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 19–20 March 2021.
30. Hao, L.; Deng, L.; Cheng, Y.; Jianbo, L.; Zhaoquan, G. Enhanced YOLO v3 tiny network for real-time ship detection from visual image. *IEEE Access* **2021**, *9*, 16692–16706.
31. Liu, T.; Zhou, B.; Zhao, Y.; Yan, S. Ship detection algorithm based on improved YOLO V5. In Proceedings of the International Conference on Automation, Control and Robotics Engineering (CACRE), Dalian, China, 15–17 July 2021.
32. Duarte, N.; João, F.; Bruno, D.; Rodrigo, V. Real-time vision based obstacle detection in maritime Environments. In Proceedings of the International Conference on Autonomous Robot Systems and Competitions (ICARSC), Santa Maria de Feira, Portugal, 29–30 April 2022.
33. Hegarty, A.; Westbrook, G.; Glynn, D.; Murray, D.; Omerdic, E.; Toal, D. A low-cost remote solar energy monitoring system for a buoyed IoT ocean observation platform. In Proceedings of the IEEE World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 15–18 April 2019.
34. Micaela, V.; Gianluca, B.; Davide, S.; Mattia, V.; Marco, A.; Francesco, G.; Alessandro, C.; Roberto, C.; Marko, B.; Marco, S. A systematic assessment of embedded neural networks for object detection. In Proceedings of the International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 8–11 September 2020.
35. Kanokwan, R.; Vasaka, V.; Ryousei, T. Evaluating the power efficiency of deep learning inference on embedded GPU systems. In Proceedings of the International Conference on Information Technology (INCIT), Nakhonpathom, Thailand, 2–3 November 2017.
36. Chun, R.H.; Wei, Y.H.; Yi, S.L.; Chien, C.L.; Yu, W.Y. A Content-Adaptive Resizing Framework for Boosting Computation Speed of Background Modeling Methods. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *52*, 1192–1204.
37. Chun, R.H.; Wei, C.W.; Wei, A.W.; Szu, Y.L.; Yen, Y.L. USEAQ: Ultra-Fast Superpixel Extraction via Adaptive Sampling From Quantized Regions. *IEEE Trans. Image Process.* **2018**, *27*, 4916–4931.
38. Gershikov, E. Is color important for horizon line detection? In Proceedings of the IEEE International Conference on Advanced Technologies for Communications (ATC), Hanoi, Vietnam, 15–17 October 2014.
39. Chi, Y.J.; Hyun, S.Y.; Kyeong, D.M. Fast horizon detection in maritime images using region-of-interest. *Int. J. Distrib. Sensor Netw.* **2018**, *14*, 155014771879075.
40. Ferreira, J.C.; Branquinho, J.; Paulo, C.F. Fernando, Computer vision algorithms fishing vessel monitoring—Identification of vessel plate number. In Proceedings of the International Symposium on Ambient Intelligence (ISAmI), Porto, Portugal, 21–23 June 2017.
41. Shu, Z.; Shenggen, H.; Qian, G.; Xindong, L.; Can, C.; Xinzheng, Z. A fusion detection algorithm of motional ship in bridge collision avoidance system. In Proceedings of the International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 8–10 December 2017.
42. Zhan, Y.; Qing, Z.L.; Feng, N.Z. Ship detection for visual maritime surveillance from non-stationary platforms. *Ocean Eng.* **2017**, *141*, 53–63. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.