sensors

Review

The Use of Terrestrial and Maritime Autonomous Vehicles in Nonintrusive Object Inspection

Dmytro Mamchur [1,2,*], Janis Peksa [3], Antons Kolodinskis [1] and Maksims Zigunovs [1]

[1] Information Technologies Department, Turiba University, Graudu Street 68, LV-1058 Riga, Latvia
[2] Computer Engineering and Electronics Department, Kremenchuk Mykhailo Ostrohradskyi National University, Pershotravneva 20, 39600 Kremenchuk, Ukraine
[3] Institute of Information Technology, Riga Technical University, Kalku Street 1, LV-1658 Riga, Latvia
* Correspondence: dgmamchur@gmail.com

Abstract: Traditional nonintrusive object inspection methods are complex or extremely expensive to apply in certain cases, such as inspection of enormous objects, underwater or maritime inspection, an unobtrusive inspection of a crowded place, etc. With the latest advances in robotics, autonomous self-driving vehicles could be applied for this task. The present study is devoted to a review of the existing and novel technologies and methods of using autonomous self-driving vehicles for nonintrusive object inspection. Both terrestrial and maritime self-driving vehicles, their typical construction, sets of sensors, and software algorithms used for implementing self-driving motion were analyzed. The standard types of sensors used for nonintrusive object inspection in security checks at the control points, which could be successfully implemented at self-driving vehicles, along with typical areas of implementation of such vehicles, were reviewed, analyzed, and classified.

Keywords: self-driving vehicle; artificial intelligence; classification; nonintrusive inspection

check for updates

Citation: Mamchur, D.; Peksa, J.; Kolodinskis, A.; Zigunovs, M. The Use of Terrestrial and Maritime Autonomous Vehicles in Nonintrusive Object Inspection. *Sensors* **2022**, *22*, 7914. https://doi.org/10.3390/s22207914

Academic Editors: Markus Lienkamp, Maximilian Geisslinger and Felix Fent

Received: 12 September 2022
Accepted: 13 October 2022
Published: 18 October 2022

1. Introduction

The task of nonintrusive object inspection has become crucial nowadays due to increased goods flow and people flow in recent decades. With this increase, the number of potential threats increases as well, as in crowded places among a number of people, parcels, and baggage it is easier to hide dangerous or illegal items that smugglers could use, as well as terrorists and other criminals. On the other hand, overcontrolling these flows is highly undesirable, creating obstacles and delays in free traveling and delivery. Thus, there is an urgent need to develop nonintrusive unobtrusive methods for passive surveillance to identify potential threats in crowded places or at public checkpoints. Besides traditional nonintrusive methods used at stationary checkpoints, as described in [1], with the advance in robotics, it has become possible to extend inspection systems' equipment to mobile self-driven platforms. Such an approach is beneficial for maritime inspection and might be used for on-land purposes. Nonintrusive object inspection using self-driving vehicles could be active and passive [1–3]. During the functional assessment, different types of control points are established, where people and their belongings should pass through security gates or could be inspected with traditional equipment, such as a manual metal detector, or be searched with the help of specially trained animals (dogs, rats) [4–6]. During a passive inspection, security should be ensured without direct interaction with the object of the search, even without informing them about the searching process [7–9]. With this aim, different object and behavior recognition techniques are employed, typically based on processing the images received from surveillance video cameras installed in public places [10–14].

One relatively novel approach in the last few decades is social media analysis [15–18]. The idea behind this approach is social media profile analysis to identify suspicious persons in the early stages of their appearances in public place [17]. These methods could be automatic, using artificial intelligence and image recognition algorithms to identify specific persons, followed by a search of their social media profiles with activity analysis. These methods are relatively novel and have limited implementation nowadays due to nonsatisfiable search and analysis quality. However, they are constantly improving with the increasing knowledge base and advances in sensors and methods used for this type of analysis [19–23].

In the case of traditional nonintrusive object inspection, in some instances, it is impossible to use fixed control points. Thus, mobile and self-moving vehicles might help [24–28]. As an example, such inspection could be provided for underwater object surveillance during border control, when it is not easy to give a traditional type of nonintrusive inspection with the fixed control point over large vehicles, such as cruise liners, or at a vast area, such as port or a gulf. In this case, smaller remotely controlled or self-driven vehicles equipped with a set of sensors, transducers, and other technologies for nonintrusive control are in use to provide security inspection activities [29–32]. Such vehicles have proven to be effective for underwater inspection, but they still need to be improved with more advanced sensors and signal processing algorithms [33]. Similarly, autonomous self-moving on-land vehicles could be used for nonintrusive inspection in crowded places to increase public safety with early suspicious object detection. To make reviewing more efficient, the process also should be automated. One of the possible solutions is to employ self-moving vehicles carrying inspection equipment so that they could autonomously inspect massive objects, crowds, or object flows with the help of nonintrusive control; analyze information; and send alarms to security officers in case of suspicious situation being detected [34].

2. Typical Structure of Self-Moving Vehicles

As a carrier for nonintrusive object inspection equipment, self-driven autonomous vehicles could be used. Typically, these vehicles contain a set of sensors to analyze an environment detecting obstacles on their preplanned route, and a processing unit to control the vehicle's movement and recalculate the route depending on the presence of obstacles [35,36].

The typical structure of on-land self-driving vehicles that could be used for nonintrusive object inspection is shown in Figure 1. Maritime vehicles have a similar structure, except for the drive system and set of sensors adapted for maritime use.

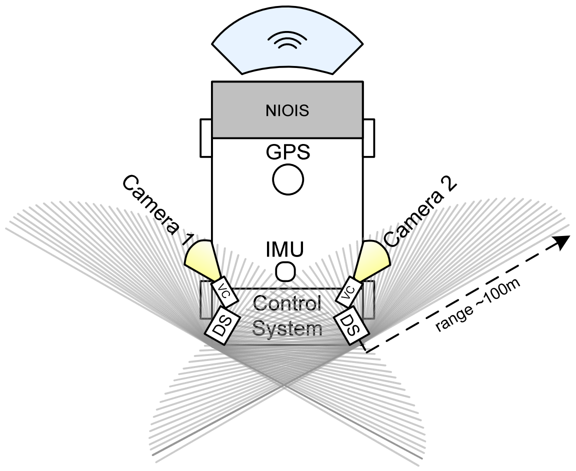

Figure 1. Typical structure of terrestrial self-driving vehicles equipped with nonintrusive object inspection system. Top view.

Such vehicles are typically equipped with distance sensors (DS), used to detect distance to the nearest obstacle, and video cameras (VC), to provide image analysis for obstacle detection and object classification. A GPS detects the current vehicle's location, inertial measurement units (IMU) collect data from gyroscopes, accelerometers, and magnetometers about vehicle motion, and a control system processes all data and implements motion control algorithms. For nonintrusive inspection purposes, the vehicle is additionally equipped with a nonintrusive object inspection system (NIOIS), which could include different types of sensors and data-processing software depending on the environment under surveillance and inspection tasks, such as portable X-ray sensors, infrared cameras, acoustic sensors, odor sensors, etc. [37]. The following chapter reviews sensors used for self-driving motion control.

3. Sensors Used for Self-Driving Vehicles

Interference avoidance is essential in developing motion control systems for self-driven vehicles. This task is typically solved using different types of sensors and control algorithms executed on microcontrollers or microcomputers. These special-purpose computing devices typically contain pins and ports to accept sensor information via standard industrial wired or wireless data-transfer protocols.

In general, the most common method for obstacle detection uses either or both of two types of sensors: active sensors, such as various types of laser and ultrasonic distance sensors; and passive sensors, such as video cameras [38–41]. All sensors vary by their effective range, possibility to detect close and remote objects, operational speed, viewing angle, and overall price. The typical sensor types used for obstacle detection and their main features are presented in Table 1.

Table 1. Comparison of the obstacle detection sensors.

Features	LIDAR	RADAR	Ultrasonic	Video Camera
Detect close objects	low	high	very high	low
Viewing angle	360°	360°	30°	~90°
Effective distance	~100 m	~0.15–250 m	0.03–10 m	~250 m
Operation in darkness	+	+	+	−
Speed detection	+	+	−	−
Device price	USD 70,000	~USD 200	1 USD/pcs	~USD 100
Processing unit price	USD 100+	USD 100+	USD 10	USD 100+

As can be seen from Table 1, different sensors are used with different purposes for object detection. Thus, to detect close objects, ultrasonic sensors are the best choice. However, their effective distance and viewing angle are poor, although the price is very attractive. On the other hand, LIDAR and RADAR sensors have 360° viewing angle, good effective distance, but quite a high price. Thus, the choice of sensor type to implement the interference avoidance algorithm should be made carefully depending on the purpose of the self-driving vehicle. For example, for road vehicles—which themselves cost thousands of dollars, and for which a typical task consists of detecting relatively remote objects, within tens or hundreds meters distance—it is reasonable to implement RADAR and LIDAR sensors. However, for small self-driving drones, where it is quite important to create relatively cheap devices that also should move with relatively slow speed, it is reasonable to consider using a set of ultrasonic sensors. Finally, if the aim is to develop a relatively cheap device that could align with object detection and perform an object recognition algorithm based on its image, the use of a video camera might be considered. In tasks of nonintrusive object inspection with the use of autonomous vehicles, a 360° viewing angle and speed detection are not the most-needed options, while it is important to detect close objects and image recognition technology might be in use. Thus, it is reasonable to use a combination of an ultrasonic sensor and a video camera for this purpose.

3.1. Obstacle Detection with the Video Camera

This method typically employs one or two video cameras to detect obstacles based on image-processing techniques.

In the case of a single camera, the following algorithms are used:

- Algorithms based on known object recognition. They aim to recognize previously known object parameters and evaluate the distance to these objects based on known object dimensions. These algorithms are easy to implement and they are able to effectively recognize previously known objects, but are useless under any uncertainties, e.g., detecting obstacles that were not previously learned;
- Motion-based algorithms. They analyze the sequence of images and compute each pixel offset. Based on the system motion data, it is possible to detect obstacles that appear on a vehicle way without previous information on the type or shape of these obstacles. In most cases, an optical flux algorithm is used to implement this method.

In the case of stereo vision, i.e., combining information from two cameras, the following algorithms are used:

- The stereo comparison method, based on searching for common patterns in two images, computing differences in the binocular discrepancy maps, and estimating the distance to the obstacle based on the horizontal displacement;
- The method of homographic transformation, which aims to transform the image angle from one camera to the image angle of another camera, assuming any significant differences between straight and altered images as an obstacle.

Typically, a combination of the above-mentioned methods is practically used, providing sufficient information about the environment and obstacles. This information could be used in complex route-planning algorithms for self-driving vehicles. Among the significant disadvantages of interference detection methods with video cameras are their high cost and high computational complexity of the models used, which require the use of neural networks and high-performance computing devices, which, in turn, also have a high cost and energy consumption, which are especially important in the development of mobile devices [42,43].

3.2. Interference Detection Using Active Sensors

Active sensors use reflected signal analysis to compute the distance to an obstacle. This group of sensors includes LIDAR (light detection and ranging), RADAR (radio detection and ranging), ultrasonic sensors, infrared sensors, and others. The first three are the most popular in this group:

- LIDAR uses laser radiation to calculate the distance to the target;
- RADAR uses radio waves to calculate the angle, type, distance, and speed of the obstacle;
- Ultrasonic sensors use high-frequency sonic radiation to analyze the time of reflected signal detection to determine the distance to the object.

Compared to video cameras, the advantages of these methods are the possibility to distinguish objects and obstacles under different lighting conditions, higher operational range, specificity, and accuracy of obstacle detection. In addition, less computational capacity is required to calculate obstacle parameters, as these sensors' operational principle does not require complex computation for image processing. Each method has its pros and cons, as in Table 1 [44–52].

3.3. Rationale for the Obstacle Sensor Choice

Data from Table 1 show that using LIDAR sensors for small autonomous vehicles and small robots is not economically reasonable. Thus, most commercial and industrial applications use RADAR or video cameras for obstacle detection. However, even these sensors could be a bit expensive for simple applications requiring cheap solutions, such as nonintrusive surveillance during small public events. In such cases, cheap infrared and ultrasonic sensors are in use. A detailed comparison of the effectiveness of different obstacle detectors is given in [53].

Based on the provided analysis, it was concluded that for nonintrusive and unobtrusive object inspection with the implementation of self-driving vehicles, in most cases

RADAR or a video camera should be implemented as the primary sensor and an ultrasonic sensor as an additional.

Moreover, it should be mentioned that ultrasonic signal processing is another challenge, due to nontrivial data-processing and control algorithms.

3.4. Obstacle Detection Algorithms

The motion of a self-driving vehicle in a changing environment is possible only when the vehicle can adapt its behavior following the changing information about this environment. In most cases, based on the preliminary description of the environment, the route-planning module generates possible paths to a given destination. Using data from sensors in real time, the self-driving algorithm should adjust the vehicle's motion to avoid collisions via recalculating basic path parameters. The use of ultrasonic distance sensors as the main obstacle detectors is quite problematic in this case, because the raw distance data do not provide information on obstacle location. Thus, complete information on obstacles could only be obtained via examining an obstacle from different angles. As an on-duty self-driving inspection vehicle performs specific tasks, it is impossible to spend time collecting information from each possible angle. Instead, the system should obtain the maximum possible information from the data collected during the vehicle's movements. The possible solution is collecting and merging data from previous measurements and creating a continuous environment map based on these measurements. However, such an approach, with the use of environment simulation using graphical primitives (lines, polygons, circles, etc.), requires high-resolution information and a considerable data preprocessing capacity. Even after meeting all these requirements, the control system could provide false results due to information noise, errors during data collection, sensor faults, etc.

Alternatively, an environment map could be created based on different types of fullness matrices. There are two types of fullness matrices: raster matrix (fixed-cell matrix) and adaptive cell size matrix. Each of these types has its pros and cons. Using a raster matrix also requires using a large but fixed memory volume. The use of an adaptive cell size algorithm reduces the memory capacity needed in cases of vast obstacles or huge unfilled environment areas. However, it requires a lot of mighty computational power, along with frequent data updates, which also harden the use of probabilistic algorithms.

Similar approaches are used with other obstacle detection sensors, but they vary in distance range for sensor effect, computational capacity, and sensor price, as it was mentioned previously.

4. Route Planning Algorithms for Self-Driving Vehicles

Another essential task for self-driving vehicles used for nonintrusive object inspection is an implementation of route-planning and motion control algorithms to cover the whole search area detecting obstacles on their way either to inspect or avoid them, followed by preplanned route recalculation. Route planning for self-driving vehicles is searching for a geometrical interpretation of the vehicle route from starting position to the target using a typically incomplete obstacle map to avoid collision with these obstacles [54].

Depending on the vehicle type, environment, and movement control algorithm, different assumptions might be applied to build an environment simulation model and make the path-searching process more efficient.

Currently, several different algorithms are developed for obstacle avoidance and route planning, starting with the simple vehicle stop when an obstacle is detected and finishing with the vehicle adaptive behavior change depending on the type, dimensions, and other features of the detected block. These algorithms differ in the data amount needed to process, number of sensors, operational speed, spatial complexity, efficiency, and control strategy [55].

The most typical algorithms used in self-driving vehicle control systems could be divided into the following groups:

- Bug algorithms;

- Naïve, or the simplest algorithms;
- Algorithms with the use of distance sensors;
- Potential field-based algorithms;
- Graph-based algorithms;
- Formal algorithms (Dijkstra's algorithm, Floyd–Warschall algorithm);
- Heuristic algorithms (e.g., width search);
- Hybrid search algorithms (for example, A*, D*).

4.1. Bug Algorithms—Bug-1 Algorithm

Bug algorithms implement the most straightforward naïve approach aimed to move directly to the target until encountering an obstacle. In case of collision, the shape of a block is calculated, followed by a recomputation of the path to the target. These algorithms do not save previously collected data and do not use once-created environment maps. Instead, they use only directly measured data [56,57]. The first algorithms from this class were proposed in 1986 by V. Lumelskiy and O. Stepanov and were called Bug-1 and Bug-2. The difference between these algorithms is the condition for stopping the motion alongside the obstacle and continuing movement to the target.

In the Bug-1 algorithm, in the case of obstacle detection, a self-driving vehicle keeps its motion around an obstacle starting from the collision point, determining its shape. After this, vehicle calculates the obstacle contour point closest to the target, called the "starting point". Then, the vehicle moves alongside the obstacle contour to the "starting point". Reaching this point, the vehicle moves to the target with a newly recalculated route. Such an algorithm is hugely inefficient, but it guarantees to obtain any target point if possible [58,59]. An example of the movement with the Bug-1 algorithm is presented in Figure 2.

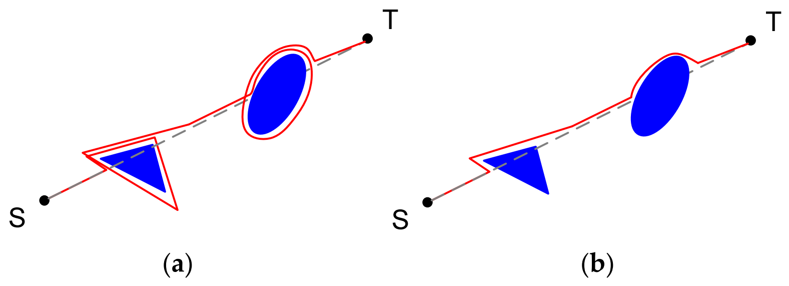

Figure 2. Bug-1 (**a**) and Bug-2 (**b**) algorithms comparison.

To increase Bug-1 efficiency, it was modernized to Bug-2, where a self-moving vehicle stops its motion around an obstacle once it crosses an M-line or a course-line—a conditional line between initial vehicle position and a target. A comparison of the Bug-1 and Bug-2 algorithms is presented in Figure 2.

The simplicity of these algorithms has several significant drawbacks. First of all, vehicle paths will not be optimal. Secondly, these algorithms do not consider the vehicle's mechanics [60]. Mainly, such a type of movement is impossible for the car.

To solve the first problem, in 1997 I. Kammon and E. Rivlin proposed the DistBug algorithm. Differently to Bug-2, the "starting point" considers the point with the distance to the target less than the distance from the next obstacle point. In addition, in contrast to the Bug-1 and Bug-2 algorithms, the direction of motion alongside the obstacle is calculated depending on the angle with which the vehicle approaches the barrier. Although these modifications significantly reduce the distance to be covered in most cases, this algorithm could lead to opposite results in several instances. An example of both cases is given in Figure 3 [61].

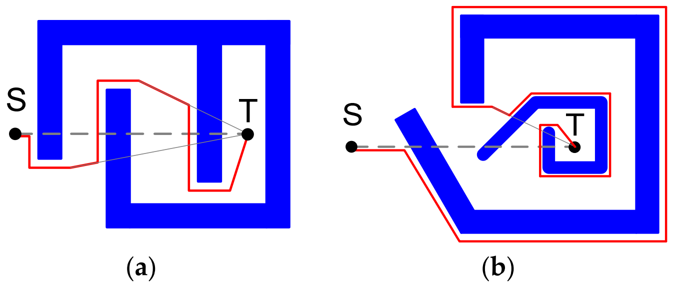

Figure 3. Improvement (**a**) and deterioration (**b**) of the vehicle movement based on the DistBug algorithm.

4.2. Bug Algorithms with the Distance Sensor

All the algorithms listed above are based on the assumption that the self-moving vehicle can detect the presence of an obstacle only at a close distance to it. However, most modern sensors used in robotics can detect obstacles at a much longer range. Based on these assumptions, Lumelsky and Skwis proposed the VisBug-21 [62] and VisBug-22 [63] modifications of Bug algorithms in 1988 and 1990, respectively. These algorithms are based on Bug-2, i.e., the vehicle follows the "M-line" but significantly reduces the trajectory. Figure 4a shows an example of the algorithm.

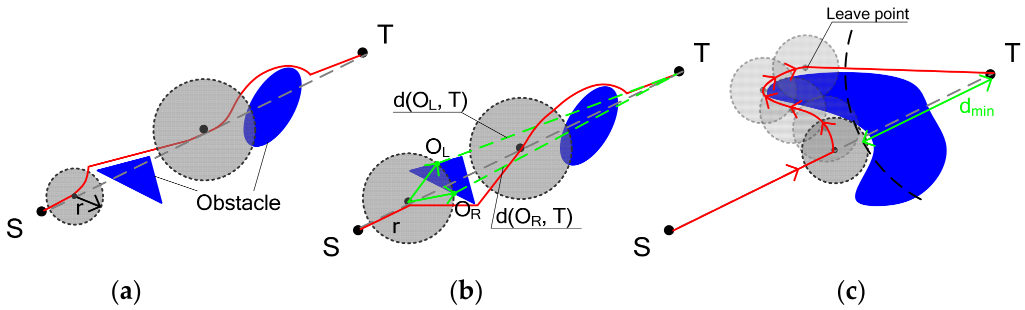

Figure 4. VisBug (**a**) and TangentBug (**b**,**c**) algorithms.

Another more successful approach was the TangentBug algorithm, developed by I. Kamon, E. Rivlin, and E. Rimon in 1997 [64] based on the DistBug algorithm. This algorithm is shown in Figure 4b,c. LTG states for 'local tangent graph', which is constructed within the distance sensor range. Next, the algorithm searches for obstacle avoidance point Oi, which minimizes the expected sum of distances d(x, O_i) + d(O_i, T), where x is the vehicle's current position and T is the target point. Figure 4c shows when the initially optimal path, supplied with new data, became much longer. In this case, the algorithm saved the d(O_i, T) value estimation, then continued its move along the obstacle until the distance to the target became shorter than this estimate. After that, the vehicle left the obstacle contour trajectory and continued its move to the target.

This algorithm proved to be the most efficient among the Bug family algorithms, and many modifications were created based on it. In particular, the WedgeBug algorithm considers the distance sensor's limited viewing angle, and the InsertBug algorithm thinks about the safety of the vehicle's move radius around the obstacle. The general classification of the Bug-type algorithms is shown in Figure 5 [58].

The basic principles of Bug algorithms, as well as Bug1, Bug2 and COM algorithms were presented by Lumelsky and Stepanov in [59,60], and later developed by Kamon and Rivlin in [61]. Insert bug and VisBug were proposed by Lumelsky and Skewis in [62] and developed in [63]. DistBug was proposed by Kamon, Rivlin and Rimon in [64]. Sankaranarayanan and Vidyasagar proposed Alg1 and Alg2 in [65]. Rev1 and Rev2 algorithms were described by Noborio, Maeda and Urakawa in [66]. Horiuchi and Noborio proposed HB-I & Ave algorithm in [67]. Lee, Adams and yeol Ryoo developed FuzzyBug algorithm and presented it in [68]. Finally, Kamon, Rimon and Rivlin proposed Tangent bug algorithm in [69].

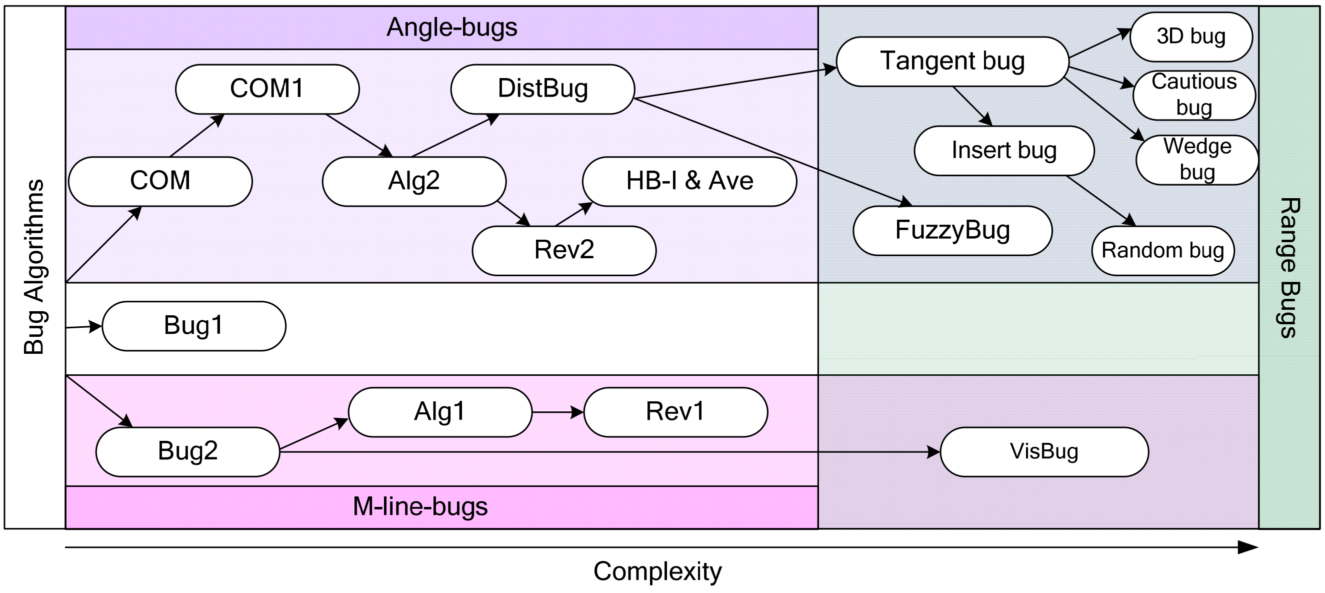

Figure 5. Bug-type algorithms classification.

4.3. Potential Field Algorithms

O. Khatib proposed a navigation algorithm based on the use of the artificial potential field. According to this concept, a self-driving vehicle is considered a particle moving in a possible area generated by a destination point and obstacles. The destination point creates the potential for attraction, while blocks generate the potential for repulsion. The vehicle moves in this field under the influence of a force that attracts it to the destination point while repulsing it from obstacles on the route [70]. This potential field is called the vector force field (VFF). The possible area could be constructed both from a previously known obstacles map and based on sensor data obtained during the movement. In the second case, if the information about the field has been changed, a recalculation of the forces influencing the area should be made. This approach has some limitations that make it hardly usable without additional modifications. For example, Figure 6 shows the configuration of obstacles where a self-driving vehicle never reaches its destination.

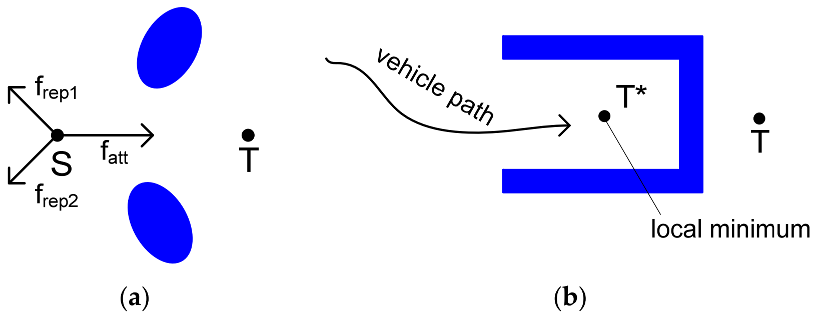

Figure 6. Examples of obstacles moving according to potential field algorithms and stopping before reaching the target point: sum of repulsive vectors balances attraction vector and vehicle stops (**a**); obstacle geometry forces move to local minimum and stop at local target point T* (**b**).

To avoid the situations shown in Figure 6b, it was proposed to construct a potential field as the sum of two components: the target field, i.e., the direction of movement to the target point; and the barrier field, consisting of repulsive forces. The sum of these components is the navigation field. Such a field can be represented as a slope with mountains, where the potential gradient equals the field inclination. An example of such field representation is shown in Figure 7 [71].

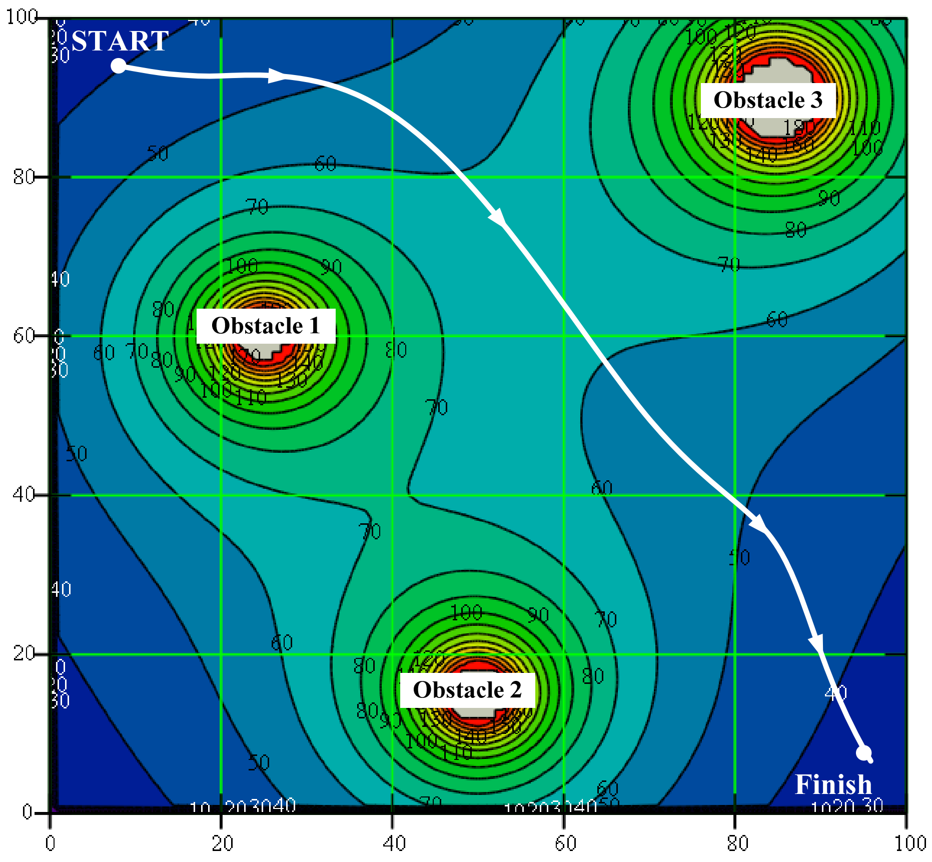

Figure 7. Graphical representation of a potential field.

The general idea of the methods is to move along the vector lines of the vector field, the potential function of which reflects the configuration of obstacles and their shape, and also the target point. This approach is suitable both in two-dimensional and three-dimensional environments. Among the potential field methods, the most popular one is the artificial potential field (APF). This algorithm is simple, with low complexity and high implementation efficiency [71]. The vector field is divided into two components: the target point, represented by an attractive vector field; and obstacles, represented by a repulsive vector field. The addition of these two vector fields along the vehicle's trajectory allows one to solve two tasks: moving to a given target point and avoiding obstacles. The technology of path planning using the artificial potential method is simple, and thus allows one to control the movement in real time. However, the method has a significant drawback: the existence of local minima is possible. A number of approaches have been proposed to eliminate this shortcoming, but there is still no completely satisfactory solution, and these approaches could be used only in certain situations under specified conditions. The method also has an uncertainty associated with the choice of the function coefficients for construction the potential functions. These two factors limit the wide use of the artificial potential method for the solution of practical problems. For example, these algorithms do not allow one to avoid unsolvable situations similar to presented in Figure 6a.

Various approaches were used to improve this type of algorithm, including the implementation of neural networks, high-level control algorithms, and advanced algorithms where the potential field is calculated by taking into account not only the absolute position of the self-driving vehicle and obstacles, but also the vehicle position relative to the obstacle and ignoring the blocks behind the vehicle. A significant disadvantage of these algorithms is that calculating the potential field requires information about most of the obstacle's positions within the area, albeit inaccurately. This could lead to erroneous results in a significant incompleteness of the information, and in some cases, may prevent the vehicle from reaching the target point. Another disadvantage is the representation of the vehicle as a particle that does not consider its kinematic parameters (for example, the vehicle cannot make a turn at a significant angle without passing a certain distance).

4.4. Algorithms on Graphs

Graph-based algorithms assume dividing space into cells, considered nodes, and creating a graph by connecting adjacent cells with edges of different weights. The boundaries that reach nodes with the obstacles usually indicate infinitely immense importance, which allows one to implement any search algorithm for the graph. The use of standard algorithms such as the Dijkstra or Floyd–Worschel algorithm does not consider the desired motion or other preknown environment parameters, leading to a large number of nodes that need to be processed. For example, the Dijkstra algorithm searches for a destination evenly in all directions, and the Floyd–Worschel algorithm handles all available edges, leading to the algorithm's quadratic complexity. A heuristic breath-first search algorithm can be used in a predefined heuristic (the simplest case is the distance to the target). The result of such an algorithm will be a vehicle route similar to Bug-type algorithms.

4.5. Hybrid Search Algorithms

The idea of the hybrid search algorithm A* is to apply the existing heuristics to the Dijkstra algorithm. Algorithm A* uses heuristics to determine the nodes that should be analyzed first. This would help reduce the Dijkstra algorithm's running time by reducing the number of nodes under analysis. This preserves the feature of the Dijkstra algorithm concerning the guaranteed pathfinding, and if the heuristic is admissible, the finding of the shortest path is also guaranteed. The most popular are:

(1) Dijkstra's algorithm;
(2) Width search algorithm;
(3) Algorithm A*.

All the above-listed algorithms can work only with an a priori (preliminary) created environment map and must be restarted once the map changes. To improve the efficiency of these algorithms in dynamically changing environment, two groups of improvements were developed for the A* algorithm, which allow one to update the previously calculated path in case new information appeared:

- D* algorithms: D * and Focused D *;
- LPA* algorithms: LPA* and D* Lite algorithms.

Both options implement the A* algorithm, saving all its parameters. If the information about previously known parts of the environment changes and it is not updated, such preservation causes additional challenges. A significant disadvantage of these algorithms is their spatial complexity, b^2, where b is the number of nodes or cells of the obstacle's matrix. Another downside is the necessity to process a significant number of cells each time the data are updated.

4.6. Rationale for the Choice of Algorithm

The pros and cons of the reviewed algorithms are summarized in Table 2.

Table 2. Comparison of route-planning algorithms.

Algorithm	Computational Complexity	Guarantees Goal Achievement	Takes into Account Kinematics	Algorithm Speed	Works in Dynamically Changed Environment
Bug-1	Simple	Yes	No	Slow	Yes
Bug algorithms with the distance sensor	Medium	Yes	Yes/No	Medium	Yes
Tangent-bug	Medium	Yes	No	Medium	Yes
InsertBug	Medium	Yes	Yes	Medium	Yes
Potential field algorithms	Complex	No	No	Fast	No
Algorithms on graph	Complex	No	No	Fast	No
Hybrid search algorithms	Very Complex	Yes	Yes	Fast	Yes

Based on the provided analysis, it could be concluded that algorithms based on the potential field have several disadvantages:

- They require information on the existence of all obstacles, even those that are beyond the scope of the sensors;
- They have great computational complexity, as they provide a complete recalculation of the potential field with each new measurement;
- They do not guarantee the achievement of the goal with specific configurations of obstacles or incomplete initial data;
- They do not take into account the kinematics of the self-driving vehicle.

Thus, the most typical algorithms used nowadays for self-driving vehicles are ones of the improvements A* or TangentBug-based algorithm. Moreover, these algorithms should consider the vehicle's kinematic model.

As the paper discusses the possibility of developing a small, relatively cheap self-moving vehicle for nonintrusive object inspection, it is important to employ a not-high-performance computing device for initial data processing, fed data by a necessary set of sensors. Thus, one of the most suitable solutions is implementing the TangentBug algorithm family with its modifications for route planning. For safety radius computation, it is suggested to implement the InsertBug algorithm and its improvements, which take into account vehicle kinematics and sensor parameters.

5. Self-Driving Vehicles Positioning Principles and Navigation Control

5.1. General Principles of Global Positioning

One of the most critical questions in uncrewed-vehicle motion control is detecting its geographical position. Nowadays, most solutions use a Global Positioning System for this purpose, which implements satellite signal analysis to detect an object's position.

The idea of satellite navigation takes its roots in the 1950s. When the first artificial Earth satellite was launched, American scientists led by Richard Kershner observed satellite signals and found that thanks to the Doppler effect, the frequency of the received signal rises when the satellite approaches and drops as it moves away. The essence of the discovery is that knowing the observer's coordinates makes it possible to detect the satellite's position and speed, and knowing the satellite's placement makes it possible to see the observer's speed and coordinates [72,73].

In 1973, the United States started the DNSS program, renamed later into NavStar, which aimed to launch satellites into medium Earth orbit, receive satellite signals using Earth-based equipment, and detect objects' geographical coordinates with the use of specialized software. The program was renamed to its modern name, Global Positioning System (GPS), in December 1973.

With the spread of cellular communication, it became possible to produce various devices such as devices for geographic coordinate determination and data transmission devices used to transmit the coordinates of different transportation objects, such as cars, ships, aircraft, etc. These devices are called trackers. Gradually, with the development of microelectronics and software advances, as well as with the increase in cellular communication coverage, it became possible not only to transmit an object's geographical coordinates but to perform other options, such as:

- Calculating the object's location, speed, and movement direction based on the GPS satellites' signals;
- Connecting external sensors to analog or digital inputs of a tracker;
- Reading data from the vehicle's onboard equipment via either serial port or CAN interface;
- Storing a certain amount of data in the internal memory during communication breaks;
- Transferring received data to the central server for the following processing.

Received data can either be stored at a local storage device and transferred later to a centralized database or transmitted to the centralized database in real time via cellular communication channels.

All modern tracking systems are identical. They have the same functionality, and the main differences are related to using components from different manufacturers and the functionality of particular techniques.

Therefore, for nonintrusive object inspection tasks with self-driving vehicles, the GPS module is responsible for detecting the current vehicle position and adjusting its motion algorithm if needed. The most typical implementation is in large areas, for example, at a gulf for maritime border control vehicles.

5.2. *Software for Autonomous Moving Control*

A vital component for any autonomous moving vehicle is software with built-in navigation maps and a report system. This software typically consists of two parts: the onboard vehicle's software to control its motion, and remote monitoring software, to analyze and store tracking and other data. The onboard part is typically implemented using microcontrollers or single-board computers, while the remote monitoring part could be implemented either with the WEB-based software or as a desktop solution.

The WEB service is typically used to control a small number of devices, while the main focus is operational GPS location monitoring. The WEB application is available from any computer independently of the operating system installed. This application typically has limited resources to control and analyze received data; thus, customized desktop solutions are used for more advanced and robust solutions. Furthermore, it should be considered that data in WEB-based applications typically are not aimed to be stored for a long time; as a rule, about one month. Thus, a report generated with such a system could be generated only for a limited period. With WEB-based applications, it is hardly possible to customize a program to specific needs and problems. Therefore, the undoubted advantage of WEB-based software is its ability to be started from any computer, tablet, or smartphone, neglecting the operating system installed. However, it is limited in the possibility of adjusting it to specific needs and the amount of data that could be processed.

In the case of a special-purpose software system created, it could be tailored to specific task demands, including controlling the vehicle's autonomous movements and analyzing data received from its additional sensors. In this case, all the data received from the vehicle are stored on remote storage, and they are constantly available and can be further processed with more powerful computers. Thus, it is possible to provide complex analyses on received data, including video image processing; build different analytical reports for any time; and work independently on Internet connections within local networks.

Thus, for the aims of nonintrusive object inspection with the use of self-moving vehicles, the proper solution is the use of simple microcontroller-based software to control vehicle moving and send sensor data to the remote device, and custom advanced software within the powerful remote computer to store and analyze both vehicle's movement and position data and additional nonintrusive inspection sensor data.

6. Maritime-Specific Sensors and Nonintrusive Object Inspection

Even though general principles of self-driving vehicles' motion control are similar, maritime devices have their peculiarities. During an extended period, navigational safety control was not the priority task. All changed after 11 September 2001 [74]. Since then, different technologies and instruments have been implemented for maritime border control, especially underwater, to detect possible threats, such as military objects, divers, mines, smuggling, etc. Most of these technologies and instruments require the use of inspection equipment underwater, so underwater self-driving vehicles are in use to solve this task [75–77]. Although the general principles of route-planning algorithms are similar to terrestrial ones but with an additional degree of freedom, object recognition and nonintrusive inspection principles differ because of the different natures of water and air media. Thus, one of the most critical tasks here is object recognition, aimed at detecting nonconventionalities in common shapes or the sea floor and seeing the number of manmade objects compared to natural ones. Marine specifics make it not always possible to implement the same techniques as on-land inspection, which mainly relates to the different nature of aquatic media. Thus, the most popular methods for many years were those based on the use of acoustic waves for long-distance underwater information transmission [75]. These techniques employ different types of sound and sonar sensors [75–77]. They proved efficient in gathering underwater acoustic images and initially involved operators in recognizing suspicious objects. With the latest advances in digital technologies, different types of AI-based techniques are employed to make the recognition process automatic.

Traditional methods typically used for underwater inspection use acoustic image analysis with image representation in grey-scale sonar or acoustic camera images and implement formal numerical analysis for object recognition [78]. A voting-based approach may be applied to enhance low-quality photos.

To enhance the quality of acoustic methods, additional analysis in the time–frequency domain proved efficient [75]. Such research provides faster and more reliable results than time-domain signal analysis, which allows for the detection and recognition of small targets in underwater control.

The limitation of active sonar sensors, namely their relatively high cost, adverse side effects on marine wildlife, and easy visibility to intruders, led to the intensification of research on passive object detection methods [79–82].

Novel approaches implement AI elements to solve image recognition tasks [83,84]. The downside of underwater acoustic image analysis methods is the lack of datasets to train CNNs in the classification huge range of different object types, as most such techniques are developed to detect and classify military objects. In [85], the authors created massive datasets for typical underwater targets, such as drowning victims, airplanes, mines, wrecks, and seafloor images. They implemented an approach that involves semisynthetic data generation aimed at artificially creating additional photos to enlarge the dataset, followed by training the deep convolutional neural network with the whole dataset and fine-tuning it with 70% of the authentic images dataset, implementing the transferal of deep learning results obtained on semisynthetic data to natural images.

In [86], a novel approach was proposed without using machine learning techniques to reliably detect underwater threats from sonar images. The method implements two steps: object area extraction and shape representation. Object area extraction represents image pixels as neural oscillators synchronizing with neighbors when their rhymes are similar. Shape representation detects conditions with the help of multiple resolution contours, which allows one to eliminate noises and—by implementing geometrical parameters—clearly detect object contours.

Another approach is grounded on underwater optical image processing. These techniques typically require cheaper equipment, and nowadays, pretty accurate solutions provide reliable underwater image analysis. However, in the early stages of underwater inspection, optical image-processing techniques were not that effective due to low-quality images taken [87]. Thus, additional strategies for image quality enhancement were applied. Traditional image-processing techniques assume grayscaled image processing in order to eliminate processing complexity and enhance object detection accuracy [88,89]. With the latest advances in video and image acquisition technologies, this group of methods became more popular due to cheaper hardware and the possibility to implement novel data-processing algorithms, providing reliable results even with low-quality optical images.

Even with advances in optical camera technologies, there is still an urgent need to enhance underwater images to improve the object recognition accuracy rate. In [90], a framework for underwater visual image recognition was proposed, which consists of three steps: a color correction algorithm, a super-resolution generative adversarial network, and an improved object recognition algorithm. The color correction algorithm allows one to eliminate unnecessary noise during image preprocessing. The proposed resolution improvement algorithm allows one to enhance image quality by deblurring and sharpening the image. A similar technique is widely used to enhance underwater image-processing quality [91]. Finally, the proposed object recognition algorithm allows one to improve object recognition accuracy.

Another framework [92] proposes to process underwater optical camera images in the following three stages: in the first stage, an advanced median filter is used to eliminate noise from the image; in the second stage, a convolutional neural network is employed to image recognition with the samples from ImageNet—the world's most extensive database for image recognition projects; in the third stage, preprocessed improved median filter images were used to tune a pretrained convolutional neural network (CNN) and classify objects in images. This approach demonstrated promising results in object recognition with small datasets used to train CNNs.

Another framework [93] proposes to provide underwater image recognition based on deep learning and transfer learning methods. The idea is to label and train a deep learning neural network based on in-air optical image recognition for manmade objects, followed by moving acquired CNN knowledge to underwater detection of the same objects.

Threat detection accuracy could be increased with enhanced sensor capability, which typically leads to a significant increase in sensor price, and with the implementation of advanced signal processing approaches to multiple sensor signals [94].

Finally, genetic algorithms could be implemented to build a reliable maritime border inspection system. In [95], it was proposed to build system-of-system architecture (SoSA) for maritime border control, choosing all necessary components based on genetic algorithm-based model simulation. Such an approach allows one to select the SoSA optimal configuration depending on marine surveillance needs and constraints: low price, high precision, best area coverage, etc.

In [96], the authors developed a web-map visualization portal to track and receive data from remote self-driving autonomous underwater vehicles, which might be used for remote control and management of underwater vehicles.

Another essential task in maritime border control with autonomous self-moving vehicles is the choice of self-driving motion control and route-planning algorithms [97]. Taking into account the specifics of the environment, it is not always possible to create an area map for vehicle orientation, and in most cases, there is a need to implement autonomous motion in an unknown environment. To solve this problem, a method that builds environment maps using underwater acoustic markers (also known as localization and mapping problems) is used [98].

In [99], a robust algorithm was proposed for underwater target recognition using optical camera images based on box partitioning. Such an approach overcomes problems related to noise or obstacles overlapping part of the image containing target points. However, this method, developed to be implemented for autonomous underwater vehicle docking, might be modified to detect different objects based on image processing.

Except for object recognition, a material-type detection also could be implemented in maritime border control. Thus, in [100], the authors use a neutron sensor for threat material analysis—laboratory experiments to distinguish rocks from explosives.

Thus, the provided analysis showed two possible lines in underwater object inspection: the use of high-precision expensive acoustic sensors or high-resolution video cameras or creating cheaper multisensor systems with intelligent analysis algorithms to enhance aquatic control with the help of autonomous vehicles.

7. Nonintrusive Control with Self-Driving Vehicles

To implement nonintrusive control with the use of self-driving vehicles, a set of additional equipment might be installed. X-ray machines are one of the most used traditional techniques for nonintrusive object inspection. However, such a system is impossible to imagine in crowded places as a passive nondisturbing scanning system. Thus, other techniques should be implemented, such as odor recognition, spectrography methods, ultrasonic and ultraviolet signal analysis, etc., as described in [1].

8. Conclusions

In this paper, we have presented a review on modern advances in border control with the use of self-driving autonomous vehicles, starting from the typical structure of self-driving autonomous vehicles, analyzing specific sensors used for motion control, and typical algorithms used to implement autonomous movement according to preplanned route avoiding obstacles. Standard sensor types and motion control algorithms were reviewed and analyzed for their appropriate use in different tasks. Special attention was paid to maritime object inspection with the use of self-driving autonomous vehicles as one of the most promising fields for research with many open problems. The provided analysis showed the necessity of improving motion control algorithms and nonintrusive object inspection methods and equipment to increase accuracy and reliability and eliminate unnecessary disturbance to ordinary people while ensuring high security.

Author Contributions: Conceptualization, D.M., J.P., A.K. and M.Z.; methodology, D.M., J.P., A.K. and M.Z.; formal analysis, D.M., J.P., A.K. and M.Z.; investigation, D.M., J.P., A.K. and M.Z.; resources, D.M., J.P., A.K. and M.Z.; data curation, D.M., J.P., A.K. and M.Z.; writing—original draft preparation, D.M.; writing—review and editing, J.P., A.K. and M.Z.; visualization, D.M.; supervision, D.M. and J.P.; project administration, D.M. and J.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Mamchur, D.; Peksa, J.; Le Clainche, S.; Vinuesa, R. Application and Advances in Radiographic and Novel Technologies Used for Non-Intrusive Object Inspection. *Sensors* **2022**, *22*, 2121. [CrossRef] [PubMed]
2. Tholen, B. The changing border: Developments and risks in border control management of Western countries. *Int. Rev. Adm. Sci.* **2010**, *76*, 259–278. [CrossRef]
3. Trauner, F.; Ripoll Servent, A. The Communitarization of the Area of Freedom, Security and Justice: Why Institutional Change does not Translate into Policy Change. *JCMS J. Common Mark. Stud.* **2016**, *54*, 1417–1432. [CrossRef]
4. Wasilewski, T.; Szulczynski, B.; Wojciechowski, M.; Kamysz, W.; Gebicki, J. A highly selective biosensor based on peptide directly derived from the HarmOBP7 aldehyde binding site. *Sensors* **2019**, *19*, 4284. [CrossRef]
5. Di-Poï, N.; Milinkovitch, M.C. Crocodylians evolved scattered multi-sensory micro-organs. *Evodevo* **2013**, *4*, 19. [CrossRef] [PubMed]

6. Qi, P.F.; Meng, Q.H.; Zeng, M. A CNN-based simplified data processing method for electronic noses. In Proceedings of the ISOCS/IEEE International Symposium on Olfaction and Electronic Nose, Montreal, QC, Canada, 28–31 May 2017; pp. 1–3. [CrossRef]
7. Polner, M. Customs and Illegal Trade: Old Game–New Rules. *J. Borderl. Stud.* **2015**, *30*, 329–344. [CrossRef]
8. Nguyen, H.D.; Cai, R.; Zhao, H.; Kot, A.C.; Wen, B. Towards More Efficient Security Inspection via Deep Learning: A Task-Driven X-ray Image Cropping Scheme. *Micromachines* **2022**, *13*, 565. [CrossRef]
9. Yang, H.; Zhang, D.; Qin, S.; Cui, T.J.; Miao, J. Real-Time Detection of Concealed Threats with Passive Millimeter Wave and Visible Images via Deep Neural Networks. *Sensors* **2021**, *21*, 8456. [CrossRef]
10. Wang, J.; Jiang, K.; Zhang, T.; Gu, X.; Liu, G.; Lu, X. Visible–Infrared Person Re-Identification via Global Feature Constraints Led by Local Features. *Electronics* **2022**, *11*, 2645. [CrossRef]
11. Capasso, P.; Cimmino, L.; Abate, A.F.; Bruno, A.; Cattaneo, G. A PNU-Based Methodology to Improve the Reliability of Biometric Systems. *Sensors* **2022**, *22*, 6074. [CrossRef]
12. Elordi, U.; Lunerti, C.; Unzueta, L.; Goenetxea, J.; Aranjuelo, N.; Bertelsen, A.; Arganda-Carreras, I. Designing Automated Deployment Strategies of Face Recognition Solutions in Heterogeneous IoT Platforms. *Information* **2021**, *12*, 532. [CrossRef]
13. Huszár, V.D.; Adhikarla, V.K. Live Spoofing Detection for Automatic Human Activity Recognition Applications. *Sensors* **2021**, *21*, 7339. [CrossRef] [PubMed]
14. Montaño-Serrano, V.M.; Jacinto-Villegas, J.M.; Vilchis-González, A.H.; Portillo-Rodríguez, O. Artificial Vision Algorithms for Socially Assistive Robot Applications: A Review of the Literature. *Sensors* **2021**, *21*, 5728. [CrossRef] [PubMed]
15. Palomino, M.A.; Aider, F. Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis. *Appl. Sci.* **2022**, *12*, 8765. [CrossRef]
16. Slabchenko, O.; Sydorenko, V.; Siebert, X. Development of models for imputation of data from social networks on the basis of an extended matrix of attributes. *East.-Eur. J. Enterp. Technol.* **2016**, *4*, 24–34. [CrossRef]
17. Sydorenko, V.; Kravchenko, S.; Rychok, Y.; Zeman, K. Method of Classification of Tonal Estimations Time Series in Problems of Intellectual Analysis of Text Content. *Transp. Res. Procedia* **2020**, *44*, 102–109. [CrossRef]
18. Romanovs, A.; Bikovska, J.; Peksa, J.; Vartiainen, T.; Kotsampopoulos, P.; Eltahawy, B.; Lehnhoff, S.; Brand, M.; Strebko, J. State of the Art in Cybersecurity and Smart Grid Education. In Proceedings of the 2021 IEEE 19th International Conference on Smart Technologies (EUROCON), Lviv, Ukraine, 6–8 July 2021. [CrossRef]
19. Williams, J.M. The safety/security nexus and the humanitarianisation of border enforcement. *Geogr. J.* **2016**, *182*, 27–37. [CrossRef]
20. Iqbal, A.; Amin, R.; Iqbal, J.; Alroobaea, R.; Binmahfoudh, A.; Hussain, M. Sentiment Analysis of Consumer Reviews Using Deep Learning. *Sustainability* **2022**, *14*, 10844. [CrossRef]
21. Al Naqbi, N.; Al Momani, N.; Davies, A. The Influence of Social Media on Perceived Levels of National Security and Crisis: A Case Study of Youth in the United Arab Emirates. *Sustainability* **2022**, *14*, 10785. [CrossRef]
22. Tesfagergish, S.G.; Kapočiūtė-Dzikienė, J.; Damaševičius, R. Zero-Shot Emotion Detection for Semi-Supervised Sentiment Analysis Using Sentence Transformers and Ensemble Learning. *Appl. Sci.* **2022**, *12*, 8662. [CrossRef]
23. Shahzalal, M.; Adnan, H.M. Attitude, Self-Control, and Prosocial Norm to Predict Intention to Use Social Media Responsibly: From Scale to Model Fit towards a Modified Theory of Planned Behavior. *Sustainability* **2022**, *14*, 9822. [CrossRef]
24. Sandino, J.; Vanegas, F.; Maire, F.; Caccetta, P.; Sanderson, C.; Gonzalez, F. UAV Framework for Autonomous Onboard Navigation and People/Object Detection in Cluttered Indoor Environments. *Remote Sens.* **2020**, *12*, 3386. [CrossRef]
25. Recalde, L.F.; Guevara, B.S.; Carvajal, C.P.; Andaluz, V.H.; Varela-Aldás, J.; Gandolfo, D.C. System Identification and Nonlinear Model Predictive Control with Collision Avoidance Applied in Hexacopters UAVs. *Sensors* **2022**, *22*, 4712. [CrossRef] [PubMed]
26. Khan, M.F.; Yau, K.-L.A.; Ling, M.H.; Imran, M.A.; Chong, Y.-W. An Intelligent Cluster-Based Routing Scheme in 5G Flying Ad Hoc Networks. *Appl. Sci.* **2022**, *12*, 3665. [CrossRef]
27. Ming, Z.; Huang, H. A 3D Vision Cone Based Method for Collision Free Navigation of a Quadcopter UAV among Moving Obstacles. *Drones* **2021**, *5*, 134. [CrossRef]
28. Cui, X.; Zhang, X.; Zhao, Z. Real-Time Safety Decision-Making Method for Multirotor Flight Strategies Based on TOPSIS Model. *Appl. Sci.* **2022**, *12*, 6696. [CrossRef]
29. Nikolakopoulos, K.; Kyriou, A.; Koukouvelas, I.; Zygouri, V.; Apostolopoulos, D. Combination of Aerial, Satellite, and UAV Photogrammetry for Mapping the Diachronic Coastline Evolution: The Case of Lefkada Island. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 489. [CrossRef]
30. Avola, D.; Cinque, L.; Di Mambro, A.; Diko, A.; Fagioli, A.; Foresti, G.L.; Marini, M.R.; Mecca, A.; Pannone, D. Low-Altitude Aerial Video Surveillance via One-Class SVM Anomaly Detection from Textural Features in UAV Images. *Information* **2022**, *13*, 2. [CrossRef]
31. Marzoughi, A.; Savkin, A.V. Autonomous Navigation of a Team of Unmanned Surface Vehicles for Intercepting Intruders on a Region Boundary. *Sensors* **2021**, *21*, 297. [CrossRef]
32. Tomar, I.; Sreedevi, I.; Pandey, N. State-of-Art Review of Traffic Light Synchronization for Intelligent Vehicles: Current Status, Challenges, and Emerging Trends. *Electronics* **2022**, *11*, 465. [CrossRef]
33. Zhao, Z.; Hu, Q.; Feng, H.; Feng, X.; Su, W. A Cooperative Hunting Method for Multi-AUV Swarm in Underwater Weak Information Environment with Obstacles. *J. Mar. Sci. Eng.* **2022**, *10*, 1266. [CrossRef]

34. Cetin, K.; Tugal, H.; Petillot, Y.; Dunnigan, M.; Newbrook, L.; Erden, M.S. A Robotic Experimental Setup with a Stewart Platform to Emulate Underwater Vehicle-Manipulator Systems. *Sensors* **2022**, *22*, 5827. [CrossRef] [PubMed]
35. Coppola, M.; McGuire, K.N.; Scheper, K.Y.W.; de Croon, G.C.H.E. Onboard communication-based relative localization for collision avoidance in micro air vehicle teams. *Auton. Robot.* **2018**, *42*, 1787–1805. [CrossRef] [PubMed]
36. Bresson, G.; Alsayed, Z.; Yu, L.; Glaser, S. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Trans. Intell. Veh.* **2017**, *2*, 194–220. [CrossRef]
37. Ranyal, E.; Sadhu, A.; Jain, K. Road Condition Monitoring Using Smart Sensing and Artificial Intelligence: A Review. *Sensors* **2022**, *22*, 3044. [CrossRef]
38. Fraundorfer, F.; Engels, C.; Nister, D. Topological mapping, localization and navigation using image collections. In Proceedings of the International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October 2007–2 November 2007; pp. 3872–3877. [CrossRef]
39. Goedemé, T.; Nuttin, M.; Tuytelaars, T.; Van Gool, L. Omnidirectional vision based topological navigation. *Int. J. Comput. Vis.* **2007**, *74*, 219–236. [CrossRef]
40. Kim, D.-H.; Shin, K.; Han, C.-S.; Lee, J.Y. Sensor-based navigation of a car-like robot based on bug family algorithms. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2013**, *227*, 1224–1241. [CrossRef]
41. Mueller, M.W.; Hamer, M.; D'Andrea, R. Fusing ultrawide band range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1730–1736. [CrossRef]
42. Budiharto, W. Intelligent Surveillance Robot with Obstacle Avoidance Capabilities Using Neural Network. *Comput. Intell. Neurosci.* **2015**, *2015*, 745823. [CrossRef]
43. Badrloo, S.; Varshosaz, M.; Pirasteh, S.; Li, J. Image-Based Obstacle Detection Methods for the Safe Navigation of Unmanned Vehicles: A Review. *Remote Sens.* **2022**, *14*, 3824. [CrossRef]
44. Zhang, X.; Zhou, M.; Qiu, P.; Huang, Y.; Li, J. Radar and vision fusion for the real-time obstacle detection and identification. *Ind. Robot.* **2019**, *46*, 391–395. [CrossRef]
45. Ci, W.; Xu, T.; Lin, R.; Lu, S. A Novel Method for Unexpected Obstacle Detection in the Traffic Environment Based on Computer Vision. *Appl. Sci.* **2022**, *12*, 8937. [CrossRef]
46. Neelam Jaikishore, C.; Podaturpet Arunkumar, G.; Jagannathan Srinath, A.; Vamsi, H.; Srinivasan, K.; Ramesh, R.K.; Jayaraman, K.; Ramachandran, P. Implementation of Deep Learning Algorithm on a Custom Dataset for Advanced Driver Assistance Systems Applications. *Appl. Sci.* **2022**, *12*, 8927. [CrossRef]
47. Hachaj, T. Potential Obstacle Detection Using RGB to Depth Image Encoder–Decoder Network: Application to Unmanned Aerial Vehicles. *Sensors* **2022**, *22*, 6703. [CrossRef] [PubMed]
48. Hussain, M.; Ali, N.; Hong, J.-E. Vision beyond the Field-of-View: A Collaborative Perception System to Improve Safety of Intelligent Cyber-Physical Systems. *Sensors* **2022**, *22*, 6610. [CrossRef]
49. Buckman, N.; Hansen, A.; Karaman, S.; Rus, D. Evaluating Autonomous Urban Perception and Planning in a 1/10th Scale MiniCity. *Sensors* **2022**, *22*, 6793. [CrossRef]
50. Elamin, A.; El-Rabbany, A. UAV-Based Multi-Sensor Data Fusion for Urban Land Cover Mapping Using a Deep Convolutional Neural Network. *Remote Sens.* **2022**, *14*, 4298. [CrossRef]
51. Prochowski, L.; Szwajkowski, P.; Ziubiński, M. Research Scenarios of Autonomous Vehicles, the Sensors and Measurement Systems Used in Experiments. *Sensors* **2022**, *22*, 6586. [CrossRef]
52. Kelly, C.; Wilkinson, B.; Abd-Elrahman, A.; Cordero, O.; Lassiter, H.A. Accuracy Assessment of Low-Cost Lidar Scanners: An Analysis of the Velodyne HDL–32E and Livox Mid–40's Temporal Stability. *Remote Sens.* **2022**, *14*, 4220. [CrossRef]
53. Yeong, D.J.; Velasco-Hernandez, G.; Barry, J.; Walsh, J. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. *Sensors* **2021**, *21*, 2140. [CrossRef]
54. Pires, M.; Couto, P.; Santos, A.; Filipe, V. Obstacle Detection for Autonomous Guided Vehicles through Point Cloud Clustering Using Depth Data. *Machines* **2022**, *10*, 332. [CrossRef]
55. Taylor, K.; LaValle, S.M. Intensity-based navigation with global guarantees. *Auton. Robot.* **2014**, *36*, 349. [CrossRef]
56. Xu, Q.-L.; Tang, G.-Y. Vectorization path planning for autonomous mobile agent in unknown environment. *Neural Comput. Appl.* **2013**, *23*, 2129. [CrossRef]
57. Zhu, Y.; Zhang, T.; Song, J.; Li, X. A new bug-type navigation algorithm considering practical implementation issues for mobile robots. In Proceedings of the 2010 IEEE International Conference on Robotics and Biomimetics, Tianjin, China, 14–18 December 2010; pp. 531–536. [CrossRef]
58. McGuire, K.N.; de Croon, G.C.H.E.; Tuyls, K. A comparative study of bug algorithms for robot navigation. *Robot. Auton. Syst.* **2019**, *121*, 103261. [CrossRef]
59. Lumelsky, V.; Stepanov, A. Dynamic path planning for a mobile automaton with limited information on the environment. *IEEE Trans. Autom. Control.* **1986**, *31*, 1058–1063. [CrossRef]
60. Lumelsky, V.J.; Stepanov, A.A. Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica* **1987**, *2*, 403–430. [CrossRef]
61. Kamon, I.; Rivlin, E. Sensory-based motion planning with global proofs. *IEEE Trans. Robot. Autom.* **1997**, *13*, 814–822. [CrossRef]

62. Lumelsky, V.; Skewis, T. A paradigm for incorporating vision in the robot navigation function. In Proceedings of the 1988 IEEE International Conference on Robotics and Automation, Philadelphia, PA, USA, 24–29 April 1988; Volume 2, pp. 734–739. [CrossRef]
63. Lumelsky, V.J.; Skewis, T. Incorporating range sensing in the robot navigation function and Cybernetics. *IEEE Trans. Syst.* **1990**, *20*, 1058–1069. [CrossRef]
64. Kamon, I.; Rivlin, E.; Rimon, E. A new range-sensor based globally convergent navigation algorithm for mobile robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, 22–28 April 1996; Volume 1, pp. 429–435. [CrossRef]
65. Sankaranarayanan, A.; Vidyasagar, M. A new path planning algorithm for moving a point object amidst unknown obstacles in a plane. In Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 13–18 May 1990; Volume 3, pp. 1930–1936. [CrossRef]
66. Noborio, H.; Maeda, Y.; Urakawa, K. Three or more dimensional sensor-based path-planning algorithm hd-i. In Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289), Kyongju, Korea, 17–21 October 1999; Volume 3, pp. 1699–1706.
67. Horiuchi, Y.; Noborio, H. Evaluation of path length made in sensor-based path-planning with the alternative following. In Proceedings of the 2001 ICRA IEEE International Conference on Robotics and Automation, (Cat. No.01CH37164), Seoul, Korea, 21–26 May 2001; Volume 2, pp. 1728–1735.
68. Lee, S.; Adams, T.M.; Ryoo, B.Y. A fuzzy navigation system for mobile construction robots. *Autom. Constr.* **1997**, *6*, 97–107. [CrossRef]
69. Kamon, I.; Rimon, E.; Rivlin, E. Tangentbug: A range-sensorbased navigation algorithm. *Int. J. Robot. Res.* **1998**, *17*, 934–953. [CrossRef]
70. Khatib, O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In *Autonomous Robot Vehicles*; Springer: Berlin/Heidelberg, Germany, 1986; pp. 396–404. [CrossRef]
71. Pêtrès, C.; Romero-Ramirez, M.-A.; Plumet, F. A potential field approach for reactive navigation of autonomous sailboats. *Robot. Auton. Syst.* **2012**, *60*, 1520–1527. [CrossRef]
72. Cullen, A.; Mazhar, M.K.A.; Smith, M.D.; Lithander, F.E.; Ó Breasail, M.; Henderson, E.J. Wearable and Portable GPS Solutions for Monitoring Mobility in Dementia: A Systematic Review. *Sensors* **2022**, *22*, 3336. [CrossRef] [PubMed]
73. Caporali, A.; Zurutuza, J. Broadcast Ephemeris with Centimetric Accuracy: Test Results for GPS, Galileo, Beidou and Glonass. *Remote Sens.* **2021**, *13*, 4185. [CrossRef]
74. Andritsos, F.; Mosconi, M. Port security in EU: A systemic approach. In Proceedings of the 2010 International WaterSide Security Conference, Carrara, Italy, 3–5 November 2010. [CrossRef]
75. Li, S.; Jin, X.; Yao, S.; Yang, S. Underwater Small Target Recognition Based on Convolutional Neural Network. In Proceedings of the Global Oceans 2020: Singapore–U.S. Gulf Coast, Biloxi, MS, USA, 5–30 October 2020. [CrossRef]
76. Cho, H.; Pyo, J.; Yu, S.-C. Drift error reduction based on the sonar image prediction and matching for underwater hovering. *IEEE Sens. J.* **2016**, *16*, 8566–8577. [CrossRef]
77. Pyo, J.; Cho, H.; Joe, H.; Ura, T.; Yu, S.-C. Development of hovering type AUV "Cyclops" and its performance evaluation using image mosaicing. *Ocean. Eng.* **2015**, *109*, 517–530. [CrossRef]
78. Foresti, G.L.; Murino, V.; Regazzoni, C.S.; Trucco, A. A voting-based approach for fast object recognition in underwater acoustic images. *IEEE J. Ocean. Eng.* **1997**, *22*, 57–65. [CrossRef]
79. Nie, D.; Sun, Z.; Qiao, G.; Liu, S.; Yin, Y. Kite-type passive acoustic detection system for underwater small targets. In Proceedings of the 2014 Oceans-St. John's, St. John's, NL, Canada, 14–19 September 2014. [CrossRef]
80. Felber, F. Extended Intruder Detection to Counter Advanced Underwater Threats in Ports and Harbors. In Proceedings of the 2018 IEEE International Symposium on Technologies for Homeland Security (HST), Woburn, MA, USA, 23–24 October 2018. [CrossRef]
81. Percival, A.M.; Crowe, D.V.; Crawford, A. CUwPS: An integrated system for the detection, localization, and classification of underwater threats. In Proceedings of the 2010 International WaterSide Security Conference, Carrara, Italy, 3–5 November 2010. [CrossRef]
82. Sutin, A.; Salloum, H.; DeLorme, M.; Sedunov, N.; Sedunov, A.; Tsionskiy, M. Stevens Passive Acoustic system for surface and underwater threat detection. In Proceedings of the 2013 IEEE International Conference on Technologies for Homeland Security (HST), Waltham, MA, USA, 12–14 November 2013. [CrossRef]
83. Balashova, E.A.; Zabolotskikh, E.V.; Azarov, S.M.; Khvorostovsky, K.; Chapron, B. Arctic Ocean Surface Type Classification Using SAR Images and Machine Learning Algorithms. In Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019. [CrossRef]
84. Lopera Tellez, O. Underwater threat recognition: Are automatic target classification algorithms going to replace expert human operators in the near future? In Proceedings of the OCEANS 2019-Marseille, Marseille, France, 17–20 June 2019. [CrossRef]
85. Huo, G.; Wu, Z.; Li, J. Underwater Object Classification in Sidescan Sonar Images Using Deep Transfer Learning and Semisynthetic Training Data. *IEEE Access* **2020**, *8*, 47407–47418. [CrossRef]
86. Matsuda, Y.; Ogawa, M.; Yano, M. System of detecting underwater threats in side scan sonar images. In Proceedings of the OCEANS 2015-Genova, Genova, Italy, 18–21 May 2015. [CrossRef]

87. Mallet, D.; Pelletier, D. Underwater video techniques for observing coastal marine biodiversity: A review of sixty years of publications (1952–2012). *Fish. Res.* **2014**, *154*, 44–62. [CrossRef]
88. Chen, B.; Li, R.; Bai, W.; Zhang, X.; Li, J.; Guo, R. Research on Recognition Method of Optical Detection Image of Underwater Robot for Submarine Cable. In Proceedings of the 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China, 11–13 October 2019. [CrossRef]
89. Pavin, A. Underwater object recognition in photo images. In Proceedings of the OCEANS 2015-MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015. [CrossRef]
90. Chen, Z.; Zhao, T.; Cheng, N.; Sun, X.; Fu, X. Towards Underwater Object Recognition Based on Supervised Learning. In Proceedings of the 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO), Kobe, Japan, 28–31 May 2018. [CrossRef]
91. Sun, X.; Shi, J.; Dong, J.; Wang, X. Fish recognition from low-resolution underwater images. In Proceedings of the 2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Datong, China, 15–17 October 2016. [CrossRef]
92. Jin, L.; Liang, H. Deep learning for underwater image recognition in small sample size situations. In Proceedings of the OCEANS 2017-Aberdeen, Aberdeen, UK, 19–22 June 2017. [CrossRef]
93. Yu, X.; Xing, X.; Zheng, H.; Fu, X.; Huang, Y.; Ding, X. Man-Made Object Recognition from Underwater Optical Images Using Deep Learning and Transfer Learning. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018. [CrossRef]
94. Meecham, A.; Acker, T. Underwater threat detection and tracking using multiple sensors and advanced processing. In Proceedings of the 2016 IEEE International Carnahan Conference on Security Technology (ICCST), Orlando, FL, USA, 24–27 October 2016. [CrossRef]
95. Melgar, I.; Mas, C.; Sanchis, M.; Gomez, M. Optimization of System-of-Systems architectures for maritime border control using genetic algorithms. In Proceedings of the IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society, Glendale, AZ, USA, 7–10 November 2010. [CrossRef]
96. Ordonez, C.E.; Potesta, J.J.; Malinoski, M.; Halpin, S.M. Autonomous underwater vehicle observation, real-time MetOcean, field, asset, and project execution data. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016. [CrossRef]
97. Pyo, J.; Yu, S.-C. Development of radial layout underwater acoustic marker using forward scan sonar for AUV. In Proceedings of the 2019 IEEE Underwater Technology (UT), Kaohsiung, Taiwan, 16–19 April 2019. [CrossRef]
98. Maki, T.; Shiroku, R.; Sato, Y.; Matsuda, T.; Sakamaki, T.; Ura, T. Docking method for hovering type AUVs by acoustic and visual positioning. In Proceedings of the 2013 IEEE International Underwater Technology Symposium (UT), Tokyo, Japan, 5–8 March 2013. [CrossRef]
99. Yahya, M.F.; Arshad, M.R. Robust recognition of targets for underwater docking of autonomous underwater vehicle. In Proceedings of the 2016 IEEE/OES Autonomous Underwater Vehicles (AUV), Tokyo, Japan, 6–9 November 2016. [CrossRef]
100. Obhodas, J.; Sudac, D.; Valkovic, V. Matrix Characterization of the Sea Floor in the Threat Material Detection Processes. *IEEE Trans. Nucl. Sci.* **2010**, *57*, 2762–2767. [CrossRef]