

Article

Placement of Optical Sensors in 3D Terrain Using a Bacterial Evolutionary Algorithm

Szilárd Kovács ^{1,*} , Balázs Bolemányi ¹  and János Botzheim ² 

¹ Department of Mechatronics Optics and Mechanical Engineering Informatics, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, 4-6 Bertalan Lajos Street, 1111 Budapest, Hungary; bolemanyibalazs@edu.bme.hu

² Department of Artificial Intelligence, Faculty of Informatics, Eötvös Loránd University, Pázmány P. Sétány 1/A, 1117 Budapest, Hungary; botzheim@inf.elte.hu

* Correspondence: kovacs.szilard@mogi.bme.hu

Abstract: This paper proposes an optimization framework for terrain large scale optical sensor placement to improve border protection. Compared to the often used, maximal coverage of an area approach, this method minimizes the undetected passages in the monitored area. Border protection is one of the most critical areas for sensor placement. Unlike traditional border protection solutions, we do not optimize for 2D but for 3D to prevent transit. Additionally, we consider both natural and built environmental coverings. The applied environmental model creates a highly inhomogeneous sensing area for sensors instead of the previously used homogeneous one. The detection of each sensor was provided by a line-of-sight model supplemented with inhomogeneous probabilities. The optimization was performed using a bacterial evolutionary algorithm. In addition to maximizing detection, minimizing the number of the applied sensors played a crucial role in design. These two cost components are built on each other hierarchically. The developed simulation framework based on ray tracing provided an excellent opportunity to optimize large areas. The presented simulation results prove the efficiency of this method. The results were evaluated by testing on a large number of intruders. Using sensors with different quantities and layouts in the tested $1 \times 1 \times 1$ km environment, we reduced the probability of undetected intrusion to below 0.1% and increased the probability of acceptable classification to 99%.

Keywords: sensor placement; route detection; evolutionary computing



Citation: Kovács, S.; Bolemányi, B.; Botzheim, J. Placement of Optical Sensors in 3D Terrain Using a Bacterial Evolutionary Algorithm. *Sensors* **2022**, *22*, 1161. <https://doi.org/10.3390/s22031161>

Academic Editor: Natividad Duro Carralero

Received: 5 January 2022

Accepted: 1 February 2022

Published: 3 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

This study aims at an optimization framework for terrain large scale optical sensor placement. Appropriate sensors are crucial for the operation of any automated system. The right number and position of sensors are essential for a sensor system's efficient and reliable operation. The camera is one of the essential sensors in terms of surveillance. It was chosen for this study since it is reasonably priced and provides an information-rich and easy-to-understand signal. It is also crucial to consider the environment in which sensors are placed. The uniqueness of this article is not to cover the entire area but to detect any intruders. It is not necessary to cover all points of the area for detection, but it is enough to find the targets at one point in their route.

The real-life importance of this topic is protecting critical areas such as power plants, military facilities, and country borders. To be more efficient, it is high time to reconsider the previously used 2D models. Hence, these models focused on people, objects located on the ground, and land vehicles. In the case of flying targets, the 3D environment study is inevitable. Due to their relatively easy availability, flying targets such as drones or kites are more common as potential intruders. They can both be used for smuggling goods and gathering confidential information [1–3]. Equipped with a camera, they can be sent as an outpost to avoid the border patrols and find the proper timeframe for crosses

without being caught. In addition to drone threat, summary studies were conducted on the applied sensors for drone monitoring and countermeasures against drones [3]. In line with the European Horizon projects [4], border protection is a priority for future years. One possible way to protect borders is through aerial reconnaissance. Long-term aerial reconnaissance can be achieved using airships and solar panels. Wind loading and easy visibility are the main disadvantages of airships, but their operation in a multimodal system can provide significant advantages. Sensor placement plays an important role in several other areas. Well-equipped sensor nodes with different sensors to monitor the environment and structure can be effectively used for social distancing and emergency management in a sensor network. Such a system has been tested in a park to manage visitors and create a favorable route for them to avoid crowding and plan an escape route in the event of an emergency [5].

This article has developed a new simulation-based approach to border management. The simulation uses ray tracing for effective application in highly inhomogeneous environments. Optimization was performed in 3D to prevent undetected ground and air crossings.

In Section 2, topic-related literature is reviewed. In Section 3, the simulation model is detailed. Section 4 describes the optimization method. Experimental results are presented in Section 5. Lastly, the conclusions and further works are summarized.

2. Related Literature

There are several sub-areas of border management, the two largest being physical border protection and psychological border management. The second's purpose is the study of people's behavior and psychological profiling [6]. The first is for the detection, localization, and prevention of illegal border crossings. In addition to high-altitude air traffic and ground crossings, the threat posed by low-altitude drones has increased over the last decade. The literature review can be divided into four main parts. The first part presents the sensors applicable to the boundary and area protection task and the related drone detection literature. The second topic is dynamic border protection methods. The third topic is static sensor placement. The last part contains sensor placement methods for more specialized tasks. Several studies have addressed the surveillance of drones [7]. The applied sensors for drone surveillance: camera (mono, RGB [8–14], multi-, hyperspectral, short-/longwave infrared [14]), radar [15], radio direction finder [16–18], acoustic (single [14,19,20], array, matrix) and laser detection and ranging. Various fusion techniques [14,21,22] have also appeared, mainly with visual, infrared, and acoustic sensors. In the case of ground transit, the use of geophones is also common [23,24]. There are static, dynamic solutions for boundary surveillance. Dynamic solutions include different patrol mechanisms [25,26]. Generally, short—a few hours—flight times are typical for drones [27], but developments aiming at flight times of several hours [28] also exist.

The border surveillance is often concentrated only on ground intruders, so 1D line arrangements are common in theoretical sensor placement methods. In most cases, sensors had uniform disk-shaped sensibility decreasing towards the edges [29–33]. Radars are well suited for area protection due to their large field of view [31]. Two dimensional (2D) coverage studies are more common in the literature than articles on border protection. Akbarzadeh et al. investigated an optimal sensor coverage in 3D elevation terrain and the built environment. The optimal 2D coordinates and horizontal and vertical angular positions of each sensor were optimized by simulated annealing, the Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm, and the Covariance matrix adaptation evolution strategy [34]. Unlike in the natural homogenous non-flat environments in the built-in environment, the sensor placing goal was to cover inhomogeneous flat surfaces. Altahir et al. developed a weighted coverage model for installing camera surveillance systems. The placement was based on a 2D risk map in 3D space. Inversely, the sensors were placed based on 2D weighted coverage demand [35]. As a continuation of their work, they used dynamic programming as a discrete optimization for 2D generated urban

layout [36]. Various aspects, such as power supply, energy efficiency, sensor lifetime, reliability, greedy coverage and the placement of the controllers in the sensor network, can be considered for the sensors' placement [32,37,38]. An obvious solution for a wireless sensor network is to use renewable energy with a rechargeable battery [32]. The energy efficiency can also be maintained by the timing of the sensors [39]. The goal is to ensure adequate coverage even in the event of outages. In the case of a fail, relocating nodes can provide an excellent solution to hold the uniformity of 2D coverage [40]. Energy supply is also an important aspect of border surveillance. Dong et al. implemented a boundary monitoring procedure with solar-powered sensors. In addition to the surface coverage, time optimization was also applied due to the limited energy of the sensor's battery [29,30]. The method was later expanded to include adaptive sensing range adjustment for energy-efficient, time-aligned alignment of sensors [41]. Another aspect of sensor placement is localization, which requires signal strength, time, or direction data from different sensors. Xu et al. investigated ideal sensor placement for single target localization based on circular time of arrival. The optimality criterion is to minimize the trace of the inverse Fisher information matrix [42]. Xu's hybrid localization procedure study is for single static target localization using the hybrid received-signal-strength, angle-of-arrival, and time-of-arrival measurements on the 2D plane [43]. Akbarzadeh et al. examined a new optimization approach for temporal coverage. The essence of temporal coverage is to cover the area around the most probable position of the target point with the available sensors. It was concluded that individual control of each sensor in series works better all at once [44]. After detection, target tracking and localization is the next important task [45–47]. Another exciting research area is the replacement of a temporarily failed sensor for localization. Pedrollo et al. trained a neural network to be a virtual sensor, replace unavailable sensors, and generate synthetic but still realistic data [48]. Another similar task is to observe 3D objects. De Rainville et al. created a framework for mobile robotic sensor placement with covariance matrix adaptation evolution strategy optimization. The mobile robots were equipped with optical sensors. The optimization goal was maximization of the pixel density on the area [49]. Herguedas et al. examined the optimal sensor placement for deformable bodies [50]. The procedure was later improved using RGB-D cameras [51]. An important area of the Optimal Sensor Placement Problem is the vibration measurements in various structures such as bridges [52]. The problem examines small dimensionally discrete sensor placement. Zhang et al. examined the coverage-based optimization of different bodies with different evolutionary algorithms [53]. Spielberg et al. performed a sensor placement task during the soft robotics simulation to monitor the inside of the soft robot [54].

3. Modeling

Simulation-based optimizations are becoming more common. Creating a suitable simulation environment has a competitive speed compared to a complex analytical solution. The studies in the simulation are very flexible and illustrative. In the field of sensor placement, simulation-based solutions are less common and are not applied for border protection. Nevertheless, the simulation-based approach has many advantages compared to the traditional analytical methods. Simulation is much more flexible, making it easy to examine even dynamic environments, and is easier to expand and apply for new tasks.

During the simulation, the signal's path between the object and the sensor was emphasized. Reflections were not considered during ray tracing because, in the studied natural environment and the use of optical sensors, this is not significant. Absorption and transparency were calculated for signal propagation. Algorithm 1 contains the process of applied signal propagation.

Algorithm 1 Ray Tracing.

```

function RAY TRACING(Sensors, Intruders, Environment)
  for Intruders' Route do
    function FIELD OF VIEW CHECK(Sensors, Intruders)
      return true/false, Ray strengths, Distance and Angular differences
    if Intruders in the sensors' field of view then
      function GROUND CROSSING AND BACKGROUND(Positions, Ground)
        return true/false, Closest ground backgrounds for sensors and intruders
      if Signal not cross the ground then
        function BUILT CROSSING AND BACKGROUND(Positions, Built elements)
          return true/false, Closest Built element backgrounds for sensors and intruders
        if Signal not cross built elements then
          function VEGETATION CROSSING AND BACKGROUND(Positions, Vegetation, Ray strength)
            return Ray strengths, Closest Vegetation backgrounds for sensors and intruders
          if Signal greater than the cut value then
            function CLOUD CROSSING AND BACKGROUND(Positions, Clouds, Ray strength)
              return Ray strengths, Closest Cloud backgrounds for sensors and intruders
        function SELECT CLOSES BACKGROUND(backgrounds)
          return Closest background
      return Ray strengths, Backgrounds, Distance and Angular differences

```

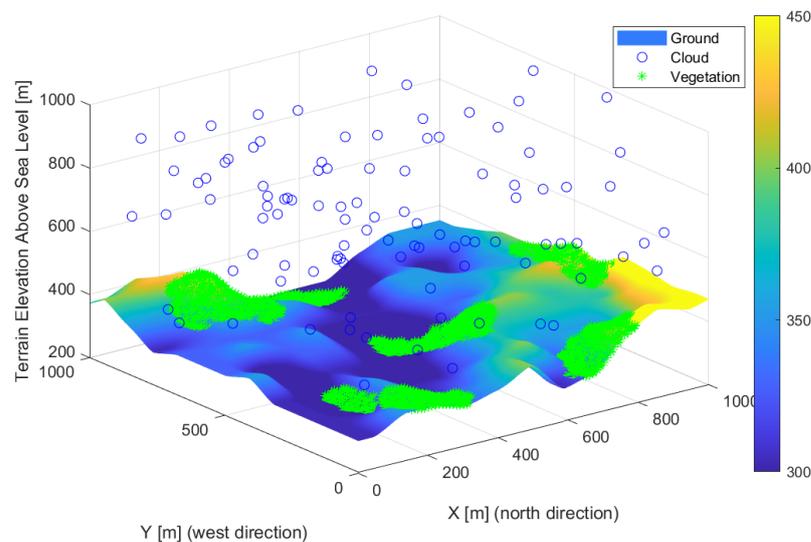
The disadvantages of simulation are that it approximates reality. Simulation neglects tend to produce better results than reality, so neglects must be considered in evaluating the results. The main difficulties in object detection are the visibility of the object, the background, the weather conditions, and the influence of the sun. In this article, the visibility and background attenuation of the object has been taken into account. Weather can significantly degrade detection for some sensors [55,56]. Three different effects can occur: reduction of visibility, particles appearing in the image, and, in the case of optical sensors, particles on the detector can obscure or blur regions. The first is the inevitable decrease in most optical sensors' detection, which can be considered as distance decreasing [55]. Particles can be filtered [57]. It is difficult to consider particles' blur and covering effect. One possibility for optimization is to punish the high angular position of the sensors. Another solution is simulating weather conditions [58]. The sun degrades the detection to different degrees depending on the quality of the optics. The sun's position can be calculated according to its geographic location [59]. In most cases, only the distance from the sensor is used for goodness calculation. In the prepared simulation, the ray passaged through the medium in the environment weakens the signal further. In addition to the strength of the signal, the signal-to-noise ratio from a detection perspective is more important. No model has been developed for optical sensors to test the signal-to-noise ratio for detection. This area is the best developed in the case of radar [60]. Signal-to-noise ratio based detection can be divided into four main parts: transmitter noise, receiver noise, the signal of the object to be observed, and background noise. For optical sensors in the field, the transmitter noise can be considered as the slow variation of sunlight. The receiver noise is mostly considered as the optical signal-to-noise ratio, containing dark current noise and spatial frequency transmission of optics. The signal is the visibility of the object. Background noise can be taken into account by the background environment. In this paper, the detection model has been compiled with the background in addition to object visibility, and the transmitter and receiver noise is not discussed. The receiver noise is sensor-specific and can be added to the discrete properties of the sensors. Based on the implemented ray tracing, the transmitter noise can be estimated by adding sun and other light sources. Based on the type of background, the estimated detection probability decreased. Table 1 shows the effect of background on the signal strength.

Table 1. Background impacts on the signal.

Environmental Element	Sign Decrease [%]
Clear sky	0
Clouds	$20 \cdot \text{cloud's density}$
Ground	[0..50] predefined
Walls	[0..50] predefined
Vegetation	$50 \cdot \text{vegetation's density}$

3.1. Environment Model

Terrain, clouds, vegetation and artificial built elements formed the modeled environment. Accurate elevation data were loaded during the design of the environment. The vegetation was loaded from a random vegetation map. The clouds and buildings were randomized. Automated loading of the entire environment is not yet implemented. The elements of the environment have different properties that are required for the placement of optical sensors. Clouds were shaped as orbs with position, size, absorption, and transmission properties. The vegetation was modeled with orbs shapes with position, size, absorption, and transparency. The constructed elements were considered rectangles with the property of position, orientation, and size. The simple shapes allow a quick parallel calculation, and more complex elements can be built. Figure 1 shows how the examined elements can construct the environment. A random vegetation map was added to the real elevation map. Clouds were generated at random locations and transparency in each iteration. Changing environmental elements increase robustness and are more realistic.

**Figure 1.** Model of the environment used for the studies, with elevation map, vegetation, clouds.

3.2. Sensor Model

Pinhole camera models were used with focus distance. The field of view and distance were calculated based on the focal length and sensor-specific typical detector size and resolutions. The object's size and its minimal pixels representation have an essential role in calculating visibility distance. Maximal signal strength is considered half of the visibility distance, and then it decreases linearly. The value calculated from the distance gave the initial value of the signal propagation. A crop value can be set for the detectable signal minima. The sensors have a position, orientation, and focal length properties for optimization. The detection range of a sensor plotted black is shown in Figure 2 illustrating the optimization result when only one sensor is used. A red line indicates a possible intruder route. The maximum visibility distance was determined during the optimization up to two-pixel imaging of a one-meter target. The real visibility is much smaller visibility due to the environment and background. The four-pixel projections are plotted in figures,

which is better related to the visibility. A larger pixel representation is recommended for a basic classification.

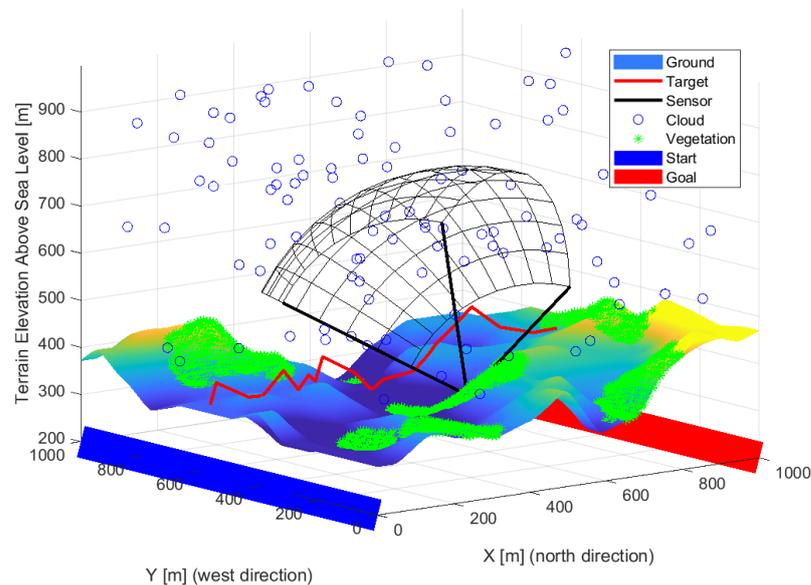


Figure 2. The detection field of a sensor is plotted black. The red line exemplifies a possible route of an object.

3.3. Target Model

Flying objects were defined in the simulation, so it was necessary to examine the 3D space instead of the usual 2D. These objects move from one edge of the simulation space ($x = 0$) to the other ($x = x_{max}$). Any point above the surface of the starting plane of the space ($x = 0$) was a possible starting point. During the route planning, the area was divided evenly along the transit direction. For the other coordinates, the maximum deflection from the previous position was calculated based on the object's top speed. The object moved randomly above the surface, between the maximum deflection and the straight direction. Objects have a size, minimal pixel representation, initial position, maximum speed, and a random path between the two edges of the study range properties. Based on the initial properties, a random path was calculated and added to the properties. Straight diagonal and mixed routes were generated in each iteration, a sample is plotted in Figures 3–5. The maximum observability is sought on each route's evaluation. The different paths (red lines) of the objects quasi evenly filled the study area. Due to the gaps, previous observations were also weighted, giving momentum to the optimization and smoothing the cost function. Some of the routes were complex despite the simple generation. Lateral, curved movements have also appeared, in addition to straight and near-straight passing attempts. Using multiple intruders in simulations with random patches results in some complex patches. The cost function considers the undetected paths with greater weight, so optimization better secures "quasi" intelligent routes. An excellent way to reduce the number of simulations is to use smarter intruders. Pinball and flood-fill algorithms can be a good solution for intelligent intruders [61].

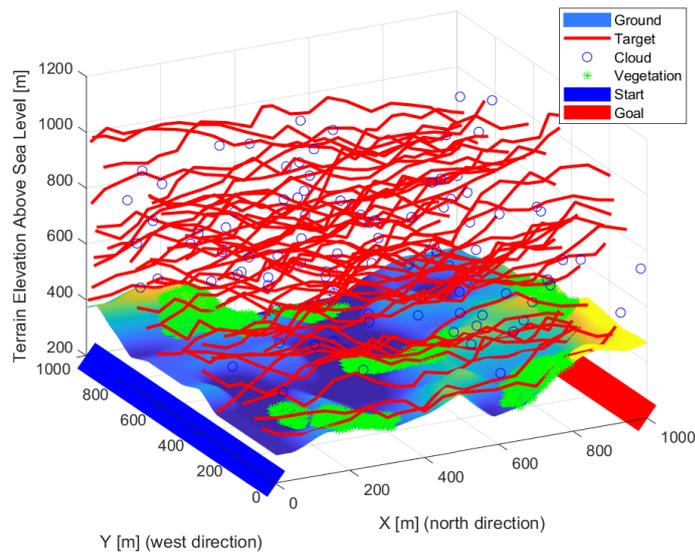


Figure 3. New routes (red lines) were generated at each iteration step.

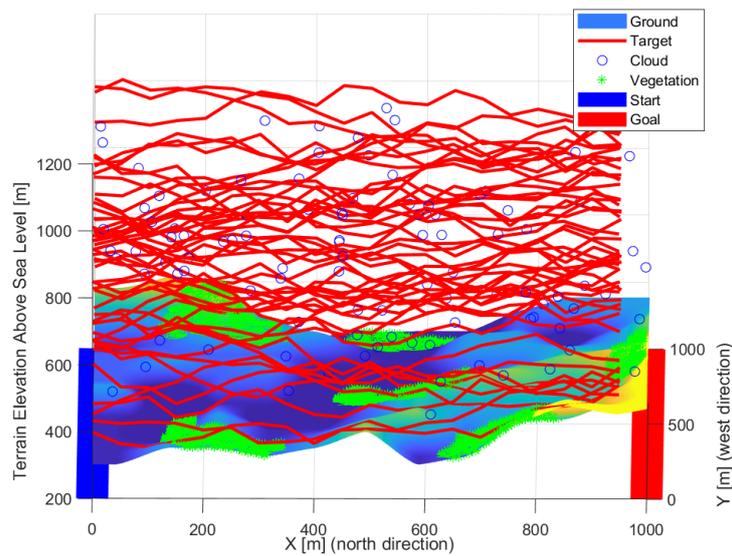


Figure 4. The different paths (red lines) of the objects quasi evenly filled the study area.

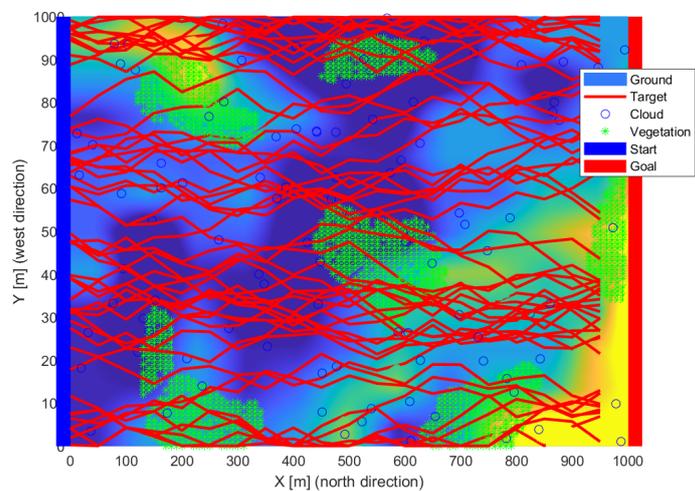


Figure 5. Some of the routes were complex despite the simple generation.

4. Optimization

4.1. Objective

The goal is to prevent unnoticed passage with minimal sensor use. It is necessary to estimate the total detection probability for the entire sensor system to recognize the hidden paths. The resulting detection is difficult to determine in sensor systems. The strongest single detection was considered the detection probability of the system. For different targets and environments, different correlation tensors can be applied. This tensor can be determined from measurements. A simple and obvious solution is to use the maximum single detection probability when combining detections of multiple sensors. This case is close to the worst case, which increases the system's reliability, but there may be negative values in the correlation tensor so that it can be better than the actual worst case. The objective can be written as the minimum observations taken from each route's maximum observation in Equation (1), where p_d is a single detection probability.

$$O_{min} = 1 - \min_{|object}(\max_{|route}(\max_{|sensor}(p_d))). \quad (1)$$

Applying the objective as a cost function resulted in highly variable costs and frequent population exchange during optimization. The cost function has been modified to improve the quality of the optimization. The number of intruders was increased, and as a standard solution in the low batch sample proportion teachings, the principle of momentum was applied. In Equation (2), the average detection of the objects was calculated. The minimum detection (Equation (1)) and the mean detection (Equation (2)) were weighted (w_o) in Equation (3). The previously computed values have also been considered in the resulting cost (C) in Equation (4). Thus, the momentum principle was realized with the weight factor (w_c). In Equation (4) n denotes the number of simulations. Compared to fixed-structure optimization tasks, it is not practical to consider historical values with greater or uniform weighting. By changing the structure, some sensors can be replaced or combined for a more optimal result despite their excellent performance.

$$O_{mean} = 1 - \text{mean}_{|object}(\max_{|route}(\max_{|sensor}(p_d))) \quad (2)$$

$$O_{ext} = w_o \cdot O_{min}(p_d) + (1 - w_o) \cdot (O_{mean}(p_d)) \quad (3)$$

$$C(n) = w_c \cdot C(n - 1) + (1 - w_c) \cdot O_{ext}(n). \quad (4)$$

A secondary goal of the hierarchical task is to minimize the number of sensors used. Based on the hierarchy, two levels of cost function were applied. The first level of the cost function is to secure the hidden passage, and the second level minimizes the number of sensors. The two-level design was implemented using two cost levels. Any arrangement that can permanently minimize passage will be one step lower in cost, and its goals will be expanded by reducing the number of sensors applied.

4.2. Individuals

Individuals consisted of varying amounts of sensors. Each sensor had fixed and variable parameters. Fixed parameters were detector size, resolution, and the range of the variable parameters. Variable parameters are position, vertical and horizontal orientation, and focal length. During optimization, the specified parameters could only take discrete values, and the variable parameters were continuous. The structure of the individuals is shown in Figure 6.

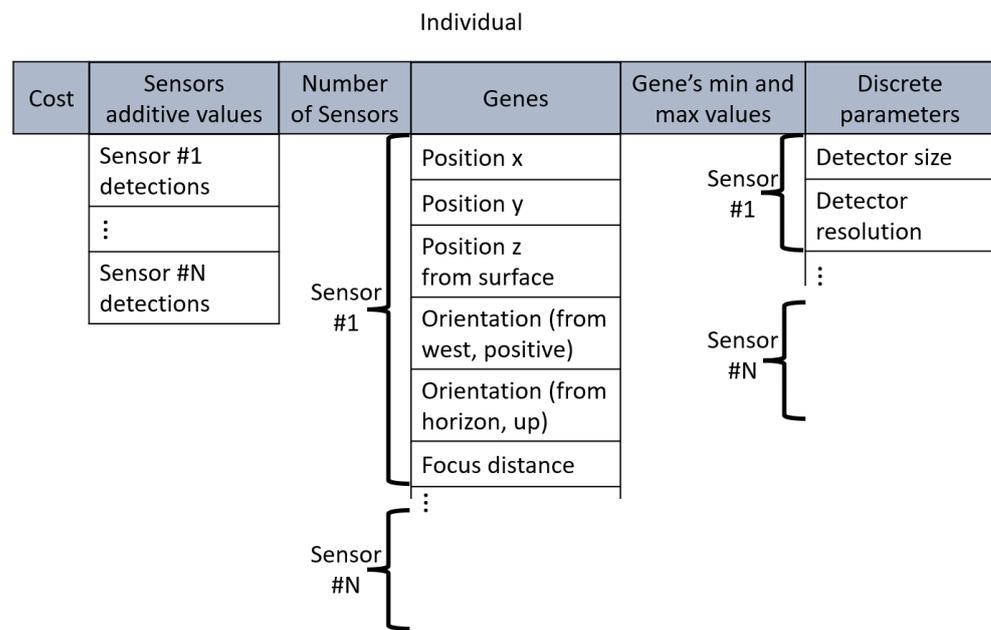


Figure 6. Structure of an individual.

4.3. Optimization Method

The Bacterial Evolutionary Algorithm (BEA) was used for optimization [62]. The BEA consists of bacterial mutation and gene transfer, shown in Algorithm 2. BEA has been applied to a wide range of problems, for instance, optimizing the fuzzy rule bases [62,63], feature selection [64], data clustering [65], and combinatorial optimization problems [66].

Algorithm 2 Bacterial Evolutionary Algorithm.

```

function PARAMETER_INITIALIZATION(params)
   $N_{pop} \leftarrow$  Number of individuals
   $criteria \leftarrow$  Stop criteria
   $opt_{pi} \leftarrow$  Options of population initialization
   $opt_{bm} \leftarrow$  Options of bacterial mutation
   $opt_{gt} \leftarrow$  Options of gene transfer
function POPULATION_INITIALIZATION( $opt_{pi}, N_{pop}$ )
  parallel( $N_{pop}$ ) create and evaluate population
while  $criteria$  do
  function BACTERIAL_MUTATION ( $opt_{bm}, Population$ )
  function GENE_TRANSFER ( $opt_{gt}, Population$ )
return Best individual

```

During bacterial mutation, random sensors or sensors with the lowest added value were chosen. Sensors may have left the individual, new sensors may have joined the individual, or sensors may have been replaced. During gene transfer, one or more sensor from the better-performing individual was transferred to a lower-performing individual or replaced with a sensor from the better individual. The replaced sensors were with the most negligible additive value or selected randomly. The algorithm and the operators used are shown in Figure 7. The optimization process is shown in Figure 8.

A sensor's additive value is calculated as the sum difference between the maximum detection of each intruder in the whole sensor system and the detection without that sensor. The total additive value of a sensor cannot be calculated due to the infinite possible path and the change of the sensor placement. Still, it can be estimated and summed with the previous values in each simulation.

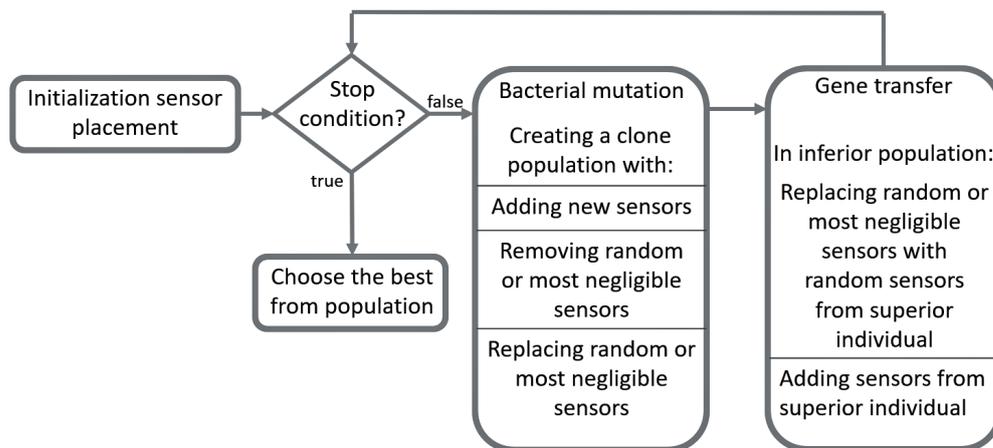


Figure 7. Bacterial Evolutionary Algorithm used for sensor placement.

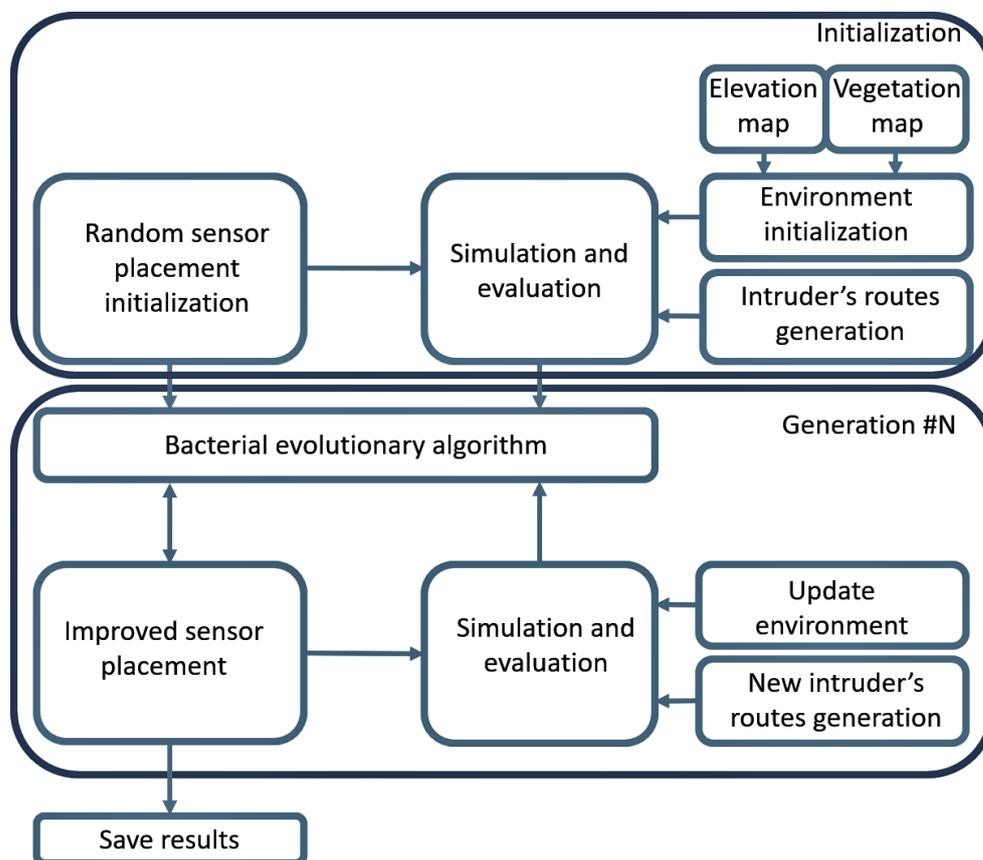


Figure 8. Optimization flow chart.

5. Experimental Results

During the experiments, a $1 \times 1 \times 1$ km area was simulated. In each iteration, the cloud map has been updated, and new paths have been initialized. 20%–80% division of random sensors or the ones with the estimated lowest added values were modified. The spontaneous mutation has weighed less due to the lack of local search. First, the sensors were selected for clone creation, and then the listed mutations occurred on them, in 30%–30%–40% of cases, a new sensor was added, removed, or replaced. The sensor's numbers were ranged and considered in selecting the mutation operation. During gene transfer, the chance of sensor transfer and sensor replacement was 50%–50%. A sensors' additive

value was calculated based on a 50%–50% weighting of the previously accumulated and current additive values. Thus, the current value got more significant weight. In general, high-resolution sensors came to the fore during optimization. Several acceptable solutions have emerged for different amounts of sensors. Low-resolution sensors can be helpful in some cases. Due to the mutation operator used, sensors with minimal added value were included temporarily and for more iteration in the case of a larger number of sensors. This error can be fixed by using a local search method. The Bacterial Memetic Algorithm (BMA) [67] complements BEA with local search. This algorithm has several variants with different local search techniques such as the Levenberg–Marquardt algorithm [67], Simulated Annealing [68], Hill climbing, and discrete local search [69,70]. In depicting the results, the sensors' detection distance was plotted based on the 4-pixel projection of a one-meter target. In general, the sensors had difficulty detecting objects at high altitudes.

Figures 9–11 show the solution for three sensors. Two sensors facing crossed, the fields of views meet this to cover most of the space. The third sensor is located independently and covers a path at the edge. The layout is able to cover high altitudes with low detection probability. It is not always possible to detect intruders flying low near the ground. Each sensor looks slightly upwards, but not so much that it is greatly affected by the weather. The effects of the sun can degrade detection based on the orientation of the map and the travel direction of the intruder.

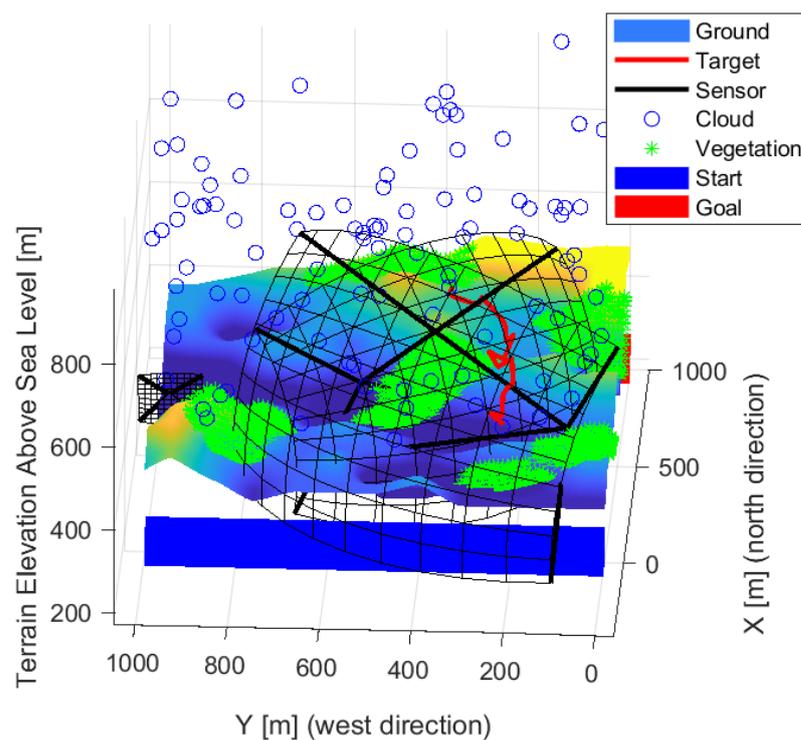


Figure 9. The solution in the case of the three sensors front-wise.

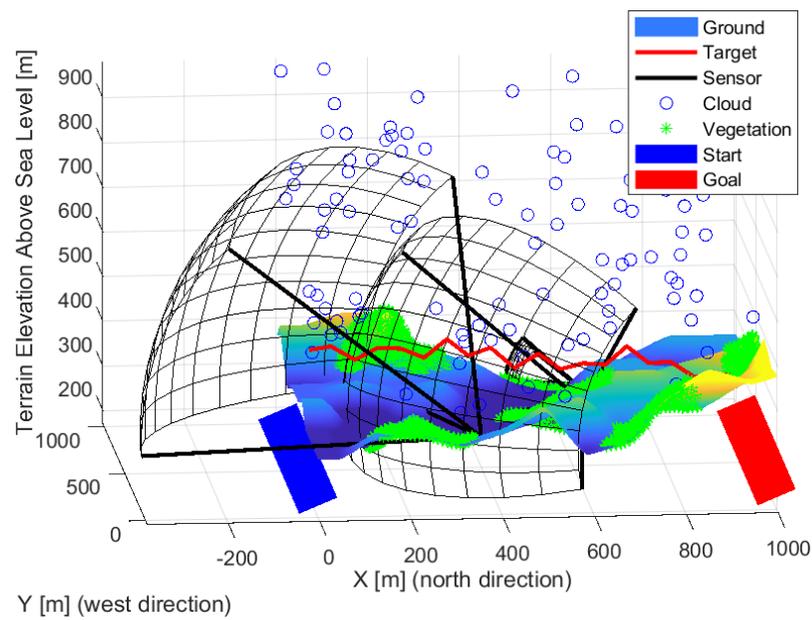


Figure 10. The solution in the case of the three sensors from the side view.

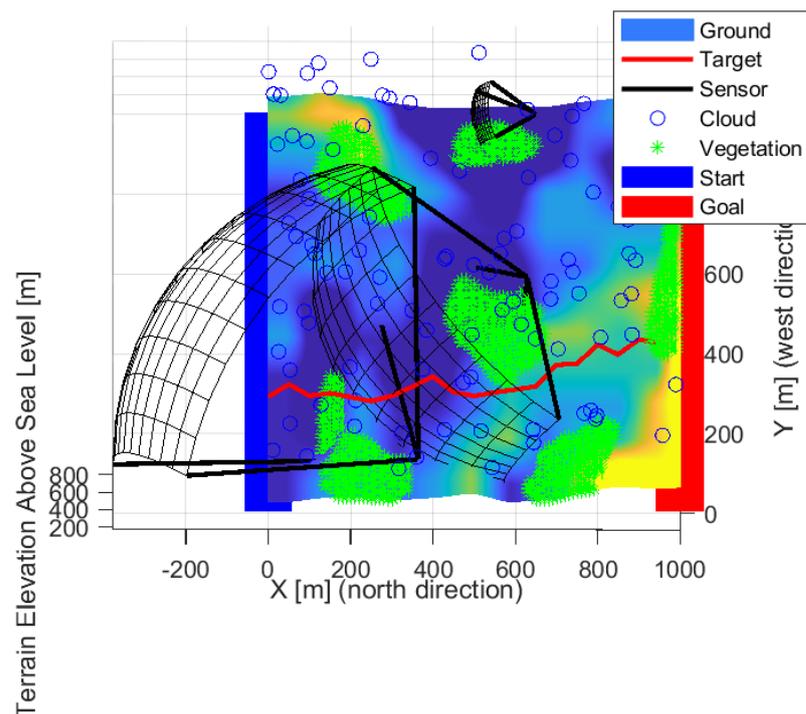


Figure 11. The solution in the case of the three sensors from above.

Figures 12–14 show the solution for four sensors. Three sensors were facing up and one forward. The sensors in the valley facing upwards are optimal as the sky background provides better detection. Upward-facing sensors can cover a large area, and the background gives them better detection. Their disadvantage is greater exposure to the weather. The forward-facing sensor is on the edge of the test area, but it covers the edge of the site and as far as possible at the edge of the forest/vegetation to see objects at low altitudes.

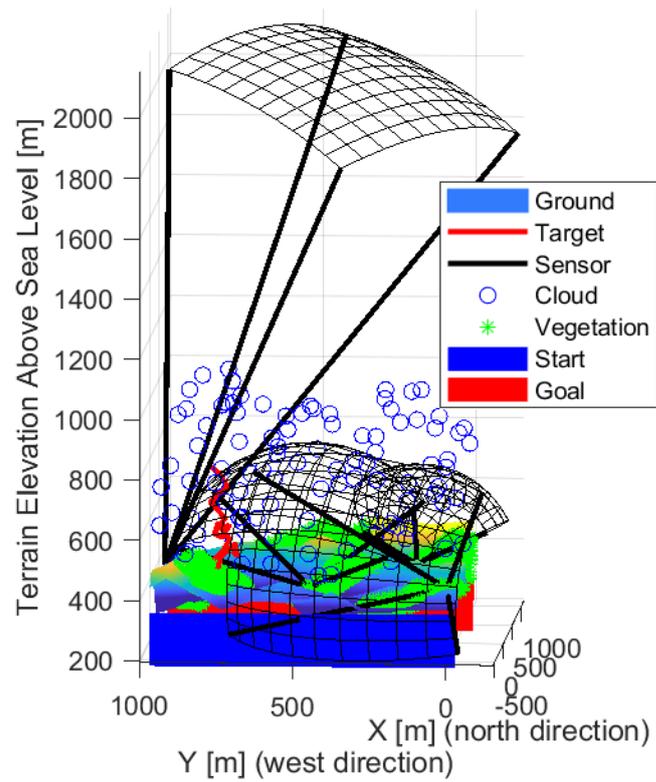


Figure 12. The solution in the case of the four sensors front-wise.

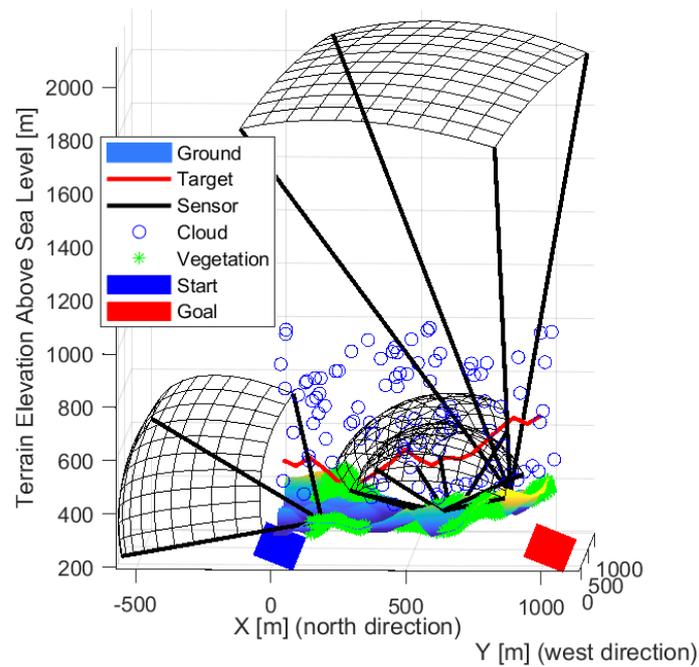


Figure 13. The solution in the case of the four sensors from the side view.

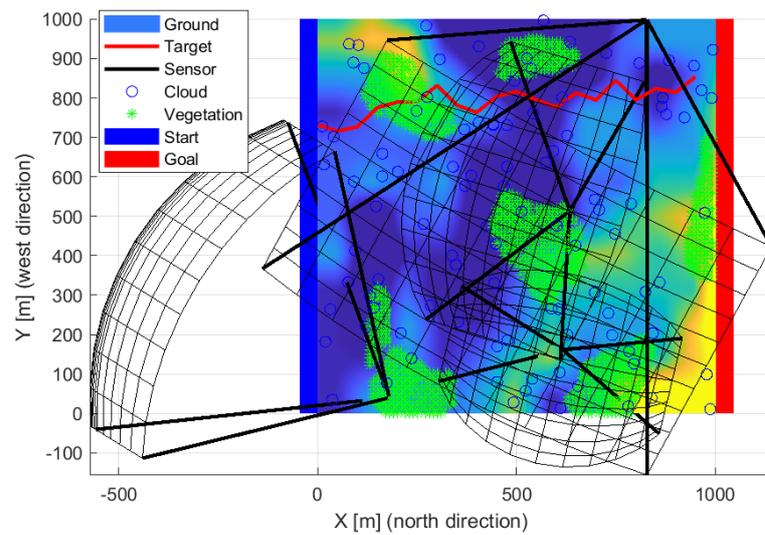


Figure 14. The solution in the case of the four sensors from above.

Figure 13 shows the sensors looking up and located in the valley are more optimal as the objects are more recognizable in the sky background. The shape of the detection space shows the task's difficulty, as, from a distance, a sensor can detect less. On the contrary, it sees a smaller area closer. Three sensors look forward and two up. A lower resolution sensor is also included in the middle. Overall, the sensors are better distributed. There are some gaps, but it covers well overall.

Figures 15–17 show solution for five sensors. They present that we can approach the optimal solution with forward and upward facing sensors.

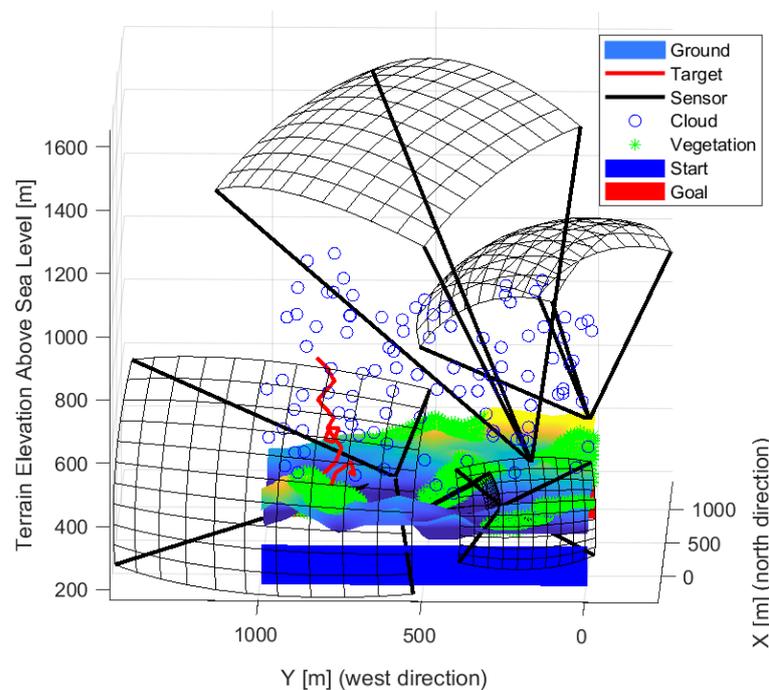


Figure 15. The solution in the case of the five sensors front-wise.

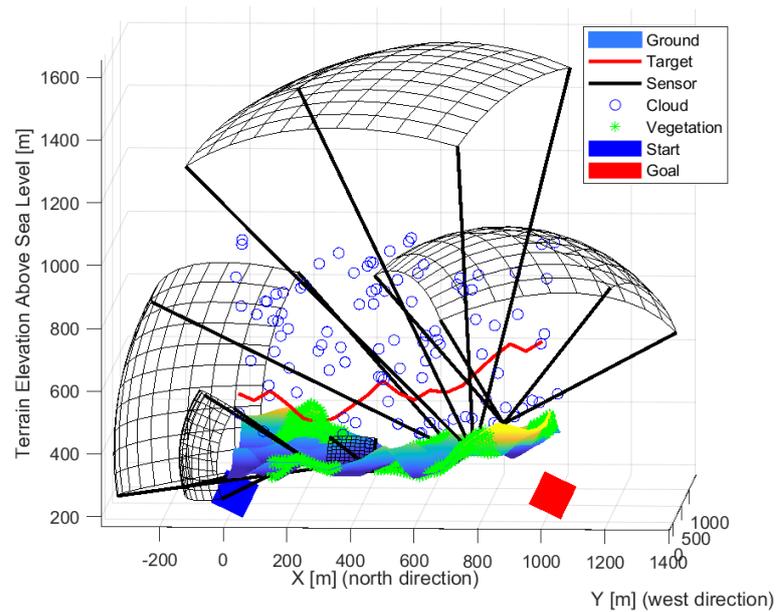


Figure 16. The solution in the case of the five sensors from the side view.

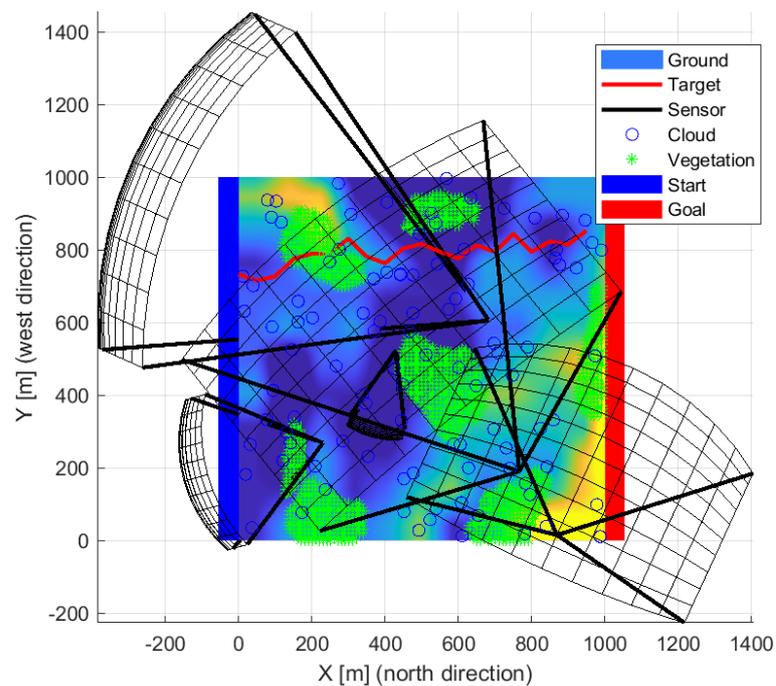


Figure 17. The solution in the case of the five sensors from above.

An arrangement using thirteen sensors is shown in the following Figures 18–20, which managed to prevent unnoticed passage. The sensors are located in almost one group, so there is no gap between them. Three upward-sensing sensors are more exposed to the effects of the weather. The visibility ranges of the upward-facing sensors touch each other and cover high-altitude routes well. Lower resolution sensors also appear mainly in valleys.

Figure 21 shows the change in cost during optimization for a different number of sensors. Each individual received an initial value of 0.5. The re-initialization of the intruder route has caused a constant fluctuation in costs. Figure 21 shows that the threshold cost is consistently reached in time with different numbers of sensors. Different threshold cost levels have been set for each case. Keeping the threshold cost in two iterations will result in a one-step lower cost per individual and an additional cost per number of sensors.

Some individuals in the population also inherit the cost of the original individual through bacterial mutation and gene transfer, and the optimization can run in parallel at the two cost levels. The inheritance of the cost gene is smaller due to the variable number of sensors, and the result of the current simulation is given more weight.

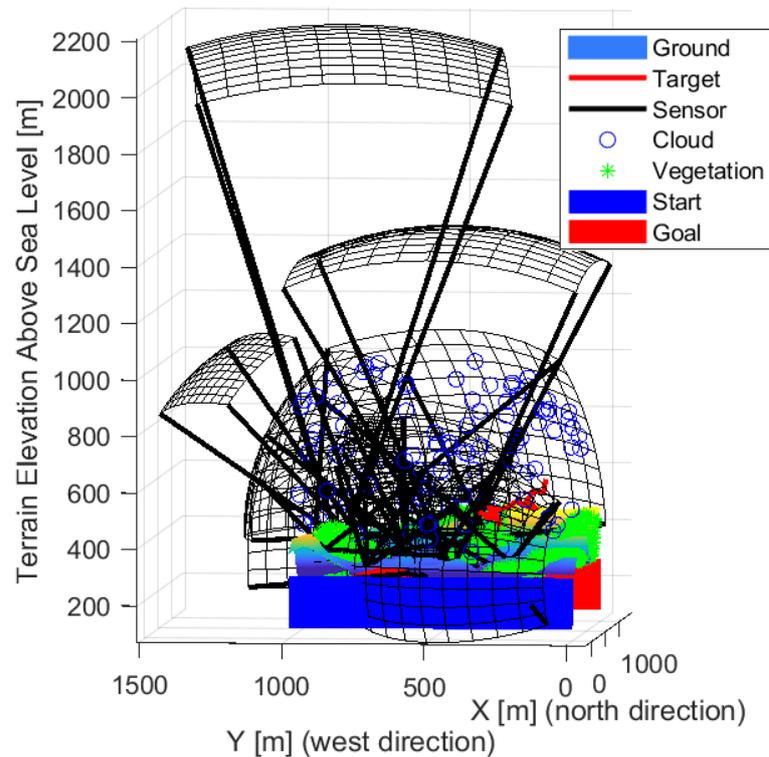


Figure 18. The solution in the case of the thirteen sensors front-wise.

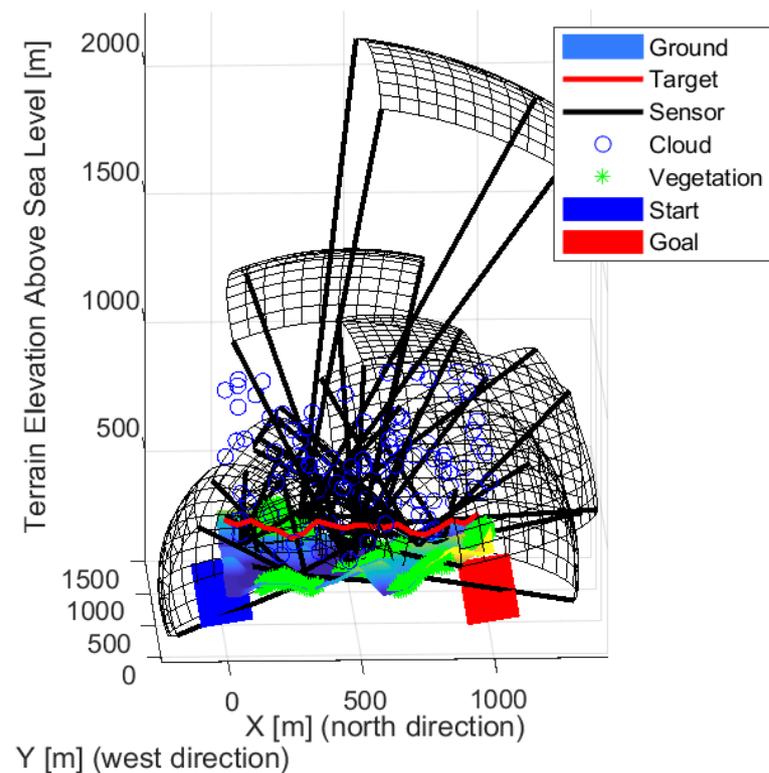


Figure 19. The solution in the case of the thirteen sensors from the side view.

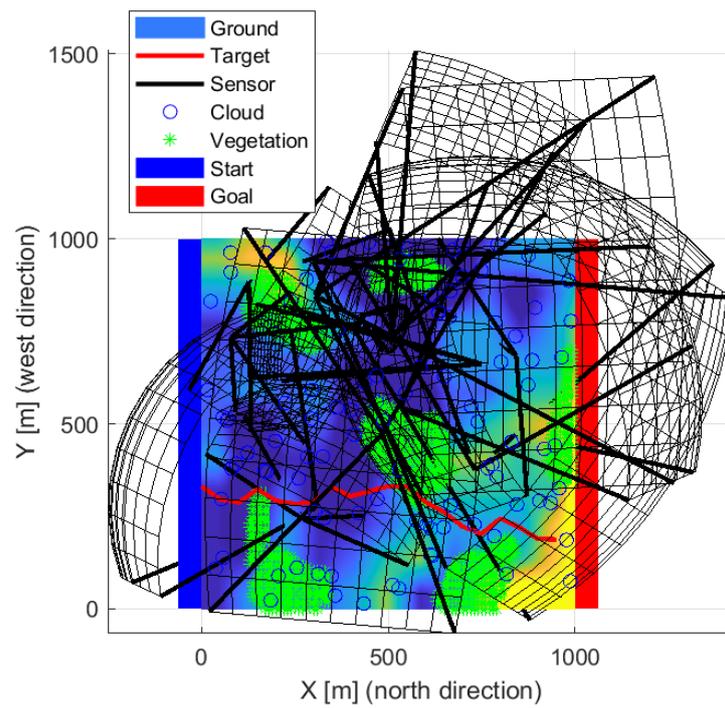


Figure 20. The solution in the case of the thirteen sensors from above.

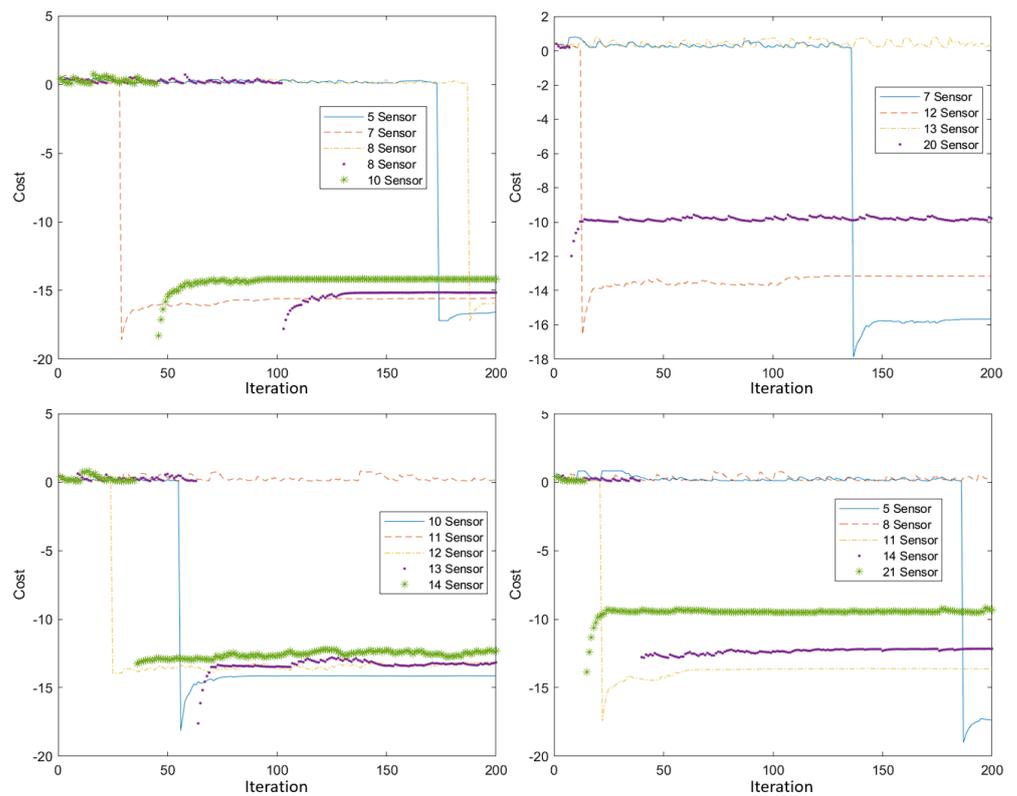


Figure 21. The two-level cost function for different numbers of sensors.

Sensor placements were tested with 10,000 random paths. Figure 22 shows an object detection probability distribution for different sensor numbers and layouts. Most detections fall into the high-reliability range of 90...100%. By using intelligent intruders [61], simulations can be made more efficient. There were both superior and inferior solutions for smaller and larger numbers of sensors. At the trend level, it can be seen that more sensors means higher reliability.

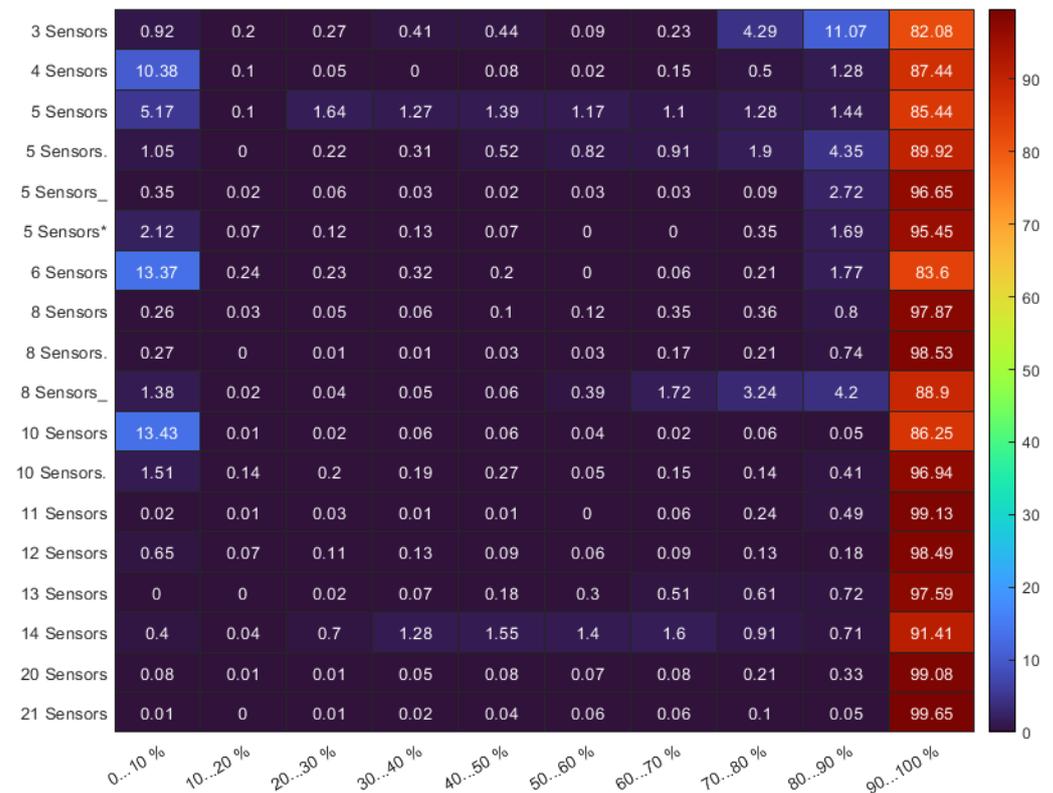


Figure 22. The detection probability distribution for different arrangements and sensor numbers.

6. Conclusions and Further Work

We have developed a method based on three-dimensional simulation compared to the currently available quasi-two-dimensional methods to prevent detection-free passage. Instead of the homogeneous or near-homogeneous detection models, we optimized a highly inhomogeneous detection model in a dynamic environment. We have developed a method based on 3D simulation compared to the currently available quasi-two-dimensional methods to prevent detection-free passage. Instead of the homogeneous/near-homogeneous detection models, we optimized a highly inhomogeneous detection model in a dynamic environment. We tested a stepped cost function for hierarchical multi-purpose optimization. The applied bacterial evolutionary algorithm was able to optimize the sensor placement. The sensors were positioned correctly in a complex environment even without local operators. For random routes, the majority of intruders (>80%) were detected with a high probability (>90% detection). Due to the simple path generation of intruders, it is more important to investigate 'quasi' intelligent intruders in the low detection categories. With the presented optimization, we succeeded in preventing undetected intrusions for 10,000 trials in the studied environment. Above the use of five sensors, undetected intrusions (<10% detection) and uncertain detections (<50% detection) can be reduced to less than 1% for intruders at low (>0.5 m) and high (<700 m) altitudes. In addition to flying intruders, the case of standing humans is also included in the parameter range. By using ten sensors, undetected intrusions and uncertain detections can be reduced to <0.1%. In addition to detection, classification is another crucial aspect. Classification is only possible with a

high detection probability (>70%). The results show that a high percentage (>80%) of the intruders has a high probability of detection (>90%) could be classified with reasonable confidence. A high percentage (99%) of the intruders are expected to classify using more than twenty sensors with an acceptable probability.

Further research aims to extract environmental data based on image segmentation automatically. Other goals are to improve the environment model with the effects of weather and sun, apply intelligent intruders, and implement appropriate local search. The implemented simulation is suitable for extracting gradient approximation. Due to discrete simulations, a local search procedure using momentum would be the most efficient [71].

Author Contributions: Conceptualization, S.K., B.B. and J.B.; Methodology, S.K. and J.B.; Software, S.K.; Supervision, J.B.; Validation, S.K.; Visualization, B.B. All authors have read and agreed to the published version of the manuscript.

Funding: Project no. 2020-00163 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2020-1.1.2-PIACI-KFI funding scheme.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Jackman, A. Consumer drone evolutions: Trends, spaces, temporalities, threats. *Def. Secur. Anal.* **2019**, *35*, 362–383. [CrossRef]
2. Yaacoub, J.P.; Noura, H.; Salman, O.; Chehab, A. Security analysis of drones systems: Attacks, limitations, and recommendations. *Internet Things* **2020**, *11*, 100218. [CrossRef]
3. Chamola, V.; Kotes, P.; Agarwal, A.; Gupta, N.; Guizani, M. A Comprehensive Review of Unmanned Aerial Vehicle Attacks and Neutralization Techniques. *Ad Hoc Netw.* **2021**, *111*, 102324. [CrossRef] [PubMed]
4. EU Research Horizon Projects. Available online: <https://frontex.europa.eu/future-of-border-control/eu-research/horizon-projects/> (accessed on 1 January 2022).
5. Fedele, R.; Merenda, M. An IoT System for Social Distancing and Emergency Management in Smart Cities Using Multi-Sensor Data. *Algorithms* **2020**, *13*, 254. [CrossRef]
6. Khan, W.; Crockett, K.; O’Shea, J.; Hussain, A.; Khan, B.M. Deception in the eyes of deceiver: A computer vision and machine learning based automated deception detection. *Expert Syst. Appl.* **2021**, *169*, 114341. [CrossRef]
7. Taha, B.; Shoufan, A. Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research. *IEEE Access* **2019**, *7*, 138669–138682. [CrossRef]
8. Nalamati, M.; Kapoor, A.; Saqib, M.; Sharma, N.; Blumenstein, M. Drone Detection in Long-Range Surveillance Videos. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–6. [CrossRef]
9. Lee, D.H. CNN-based single object detection and tracking in videos and its application to drone detection. *Multimed. Tools Appl.* **2021**, *80*, 34237–34248. [CrossRef]
10. Seidaliyeva, U.; Akhmetov, D.; Ilipbayeva, L.; Matson, E.T. Real-Time and Accurate Drone Detection in a Video with a Static Background. *Sensors* **2020**, *20*, 3856. [CrossRef]
11. Singha, S.; Aydin, B. Automated Drone Detection Using YOLOv4. *Drones* **2021**, *5*, 95. [CrossRef]
12. Jin, R.; Jiang, J.; Qi, Y.; Lin, D.; Song, T. Drone Detection and Pose Estimation Using Relational Graph Networks. *Sensors* **2019**, *19*, 1479. [CrossRef]
13. Behera, D.K.; Bazil Raj, A. Drone Detection and Classification using Deep Learning. In Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13–15 May 2020; pp. 1012–1016. [CrossRef]
14. Svanström, F.; Englund, C.; Alonso-Fernandez, F. Real-Time Drone Detection and Tracking With Visible, Thermal and Acoustic Sensors. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 7265–7272. [CrossRef]
15. Park, J.; Park, S.; Kim, D.H.; Park, S.O. Leakage Mitigation in Heterodyne FMCW Radar for Small Drone Detection With Stationary Point Concentration Technique. *IEEE Trans. Microw. Theory Tech.* **2019**, *67*, 1221–1232. [CrossRef]
16. Basak, S.; Rajendran, S.; Pollin, S.; Scheers, B. Combined RF-based drone detection and classification. *IEEE Trans. Cogn. Commun. Netw.* **2021**. [CrossRef]

17. Al-Sa'd, M.F.; Al-Ali, A.; Mohamed, A.; Khattab, T.; Erbad, A. RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database. *Future Gener. Comput. Syst.* **2019**, *100*, 86–97. [CrossRef]
18. Sciancalepore, S.; Ibrahim, O.A.; Oligeri, G.; Di Pietro, R. PiNcH: An effective, efficient, and robust solution to drone detection via network traffic analysis. *Comput. Netw.* **2020**, *168*, 107044. [CrossRef]
19. Anwar, M.Z.; Kaleem, Z.; Jamalipour, A. Machine Learning Inspired Sound-Based Amateur Drone Detection for Public Safety Applications. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2526–2534. [CrossRef]
20. Al-Emadi, S.; Al-Ali, A.; Al-Ali, A. Audio-Based Drone Detection and Identification Using Deep Learning Techniques with Dataset Enhancement through Generative Adversarial Networks. *Sensors* **2021**, *21*, 4953. [CrossRef]
21. Aledhari, M.; Razzak, R.; Parizi, R.M.; Srivastava, G. Sensor Fusion for Drone Detection. In Proceedings of the 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), Helsinki, Finland, 25–28 April 2021; pp. 1–7. [CrossRef]
22. Milani, I.; Bongioanni, C.; Colone, F.; Lombardo, P. Fusing active and passive measurements for drone localization. In Proceedings of the 2020 21st International Radar Symposium (IRS), Warsaw, Poland, 5–8 October 2020; pp. 245–249. [CrossRef]
23. Arjun, D.; Indukala, P.K.; Unnikrishna Menon, K.A. PANCHENDRIYA: A Multi-sensing framework through Wireless Sensor Networks for Advanced Border Surveillance and Human Intruder Detection. In Proceedings of the 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 17–19 July 2019; pp. 295–298. [CrossRef]
24. Arjun, D.; Indukala, P.; Menon, K.A.U. Integrated Multi-sensor framework for Intruder Detection in Flat Border Area. In Proceedings of the 2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC), Chennai, India, 21–23 August 2019; pp. 557–562. [CrossRef]
25. Qiao, Y.; Yang, J.; Zhang, Q.; Xi, J.; Kong, L. Multi-UAV Cooperative Patrol Task Planning Novel Method Based on Improved PFIH Algorithm. *IEEE Access* **2019**, *7*, 167621–167628. [CrossRef]
26. Surendonk, T.J.; Chircop, P.A. On the Computational Complexity of the Patrol Boat Scheduling Problem with Complete Coverage. *Naval Res. Logist.* **2020**, *67*, 289–299. [CrossRef]
27. Abushahma, R.I.H.; Ali, M.A.M.; Rahman, N.A.A.; Al-Sanjary, O.I. Comparative Features of Unmanned Aerial Vehicle (UAV) for Border Protection of Libya: A Review. In Proceedings of the 2019 IEEE 15th International Colloquium on Signal Processing Its Applications (CSPA), Penang, Malaysia, 8–9 March 2019; pp. 114–119. [CrossRef]
28. BorderUAS. Available online: <https://frontex.europa.eu/future-of-border-control/eu-research/horizon-projects/borderuas-xFanI/> (accessed on 1 January 2022).
29. Dong, Z.; Chang, C.Y.; Chen, G.; Chang, I.H.; Xu, P. Maximizing Surveillance Quality of Boundary Curve in Solar-Powered Wireless Sensor Networks. *IEEE Access* **2019**, *7*, 77771–77785. [CrossRef]
30. Xu, P.; Wu, J.; Shang, C.; Chang, C.Y. GSMS: A Barrier Coverage Algorithm for Joint Surveillance Quality and Network Lifetime in WSNs. *IEEE Access* **2019**, *7*, 1. [CrossRef]
31. Xu, X.; Zhao, C.; Ye, T.; Gu, T. Minimum Cost Deployment of Bistatic Radar Sensor for Perimeter Barrier Coverage. *Sensors* **2019**, *19*, 225. [CrossRef] [PubMed]
32. Liu, Y.; Chin, K.W.; Yang, C.; He, T. Nodes Deployment for Coverage in Rechargeable Wireless Sensor Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6064–6073. [CrossRef]
33. Wang, S.; Yang, X.; Wang, X.; Qian, Z. A Virtual Force Algorithm-Lévy-Embedded Grey Wolf Optimization Algorithm for Wireless Sensor Network Coverage Optimization. *Sensors* **2019**, *19*, 2735. [CrossRef] [PubMed]
34. Akbarzadeh, V.; Gagné, C.; Parizeau, M.; Argany, M.; Mostafavi, M.A. Probabilistic Sensing Model for Sensor Placement Optimization Based on Line-of-Sight Coverage. *IEEE Trans. Instrum. Meas.* **2013**, *62*, 293–303. [CrossRef]
35. Altahir, A.A.; Asirvadam, V.S.; Hamid, N.H.B.; Sebastian, P.; Hassan, M.A.; Saad, N.B.; Ibrahim, R.; Dass, S.C. Visual Sensor Placement Based on Risk Maps. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 3109–3117. [CrossRef]
36. Altahir, A.A.; Asirvadam, V.S.; Sebastian, P.; Hamid, N.H. Solving Surveillance Coverage Demand Based on Dynamic Programming. In Proceedings of the 2020 IEEE Sensors Applications Symposium (SAS), Kuala Lumpur, Malaysia, 9–11 March 2020; pp. 1–6. [CrossRef]
37. Lanza-Gutiérrez, J.M.; Caballé, N.; Gómez-Pulido, J.A.; Crawford, B.; Soto, R. Toward a Robust Multi-Objective Metaheuristic for Solving the Relay Node Placement Problem in Wireless Sensor Networks. *Sensors* **2019**, *19*, 677. [CrossRef]
38. Tahmasebi, S.; Safi, M.; Zolfi, S.; Maghsoudi, M.R.; Faragardi, H.R.; Fotouhi, H. Cuckoo-PC: An Evolutionary Synchronization-Aware Placement of SDN Controllers for Optimizing the Network Performance in WSNs. *Sensors* **2020**, *20*, 3231. [CrossRef]
39. Thomas, D.; Shankaran, R.; Sheng, Q.; Orgun, M.; Hitchens, M.; Masud, M.; Ni, W.; Mukhopadhyay, S.; Piran, M. QoS-Aware Energy Management and Node Scheduling Schemes for Sensor Network-Based Surveillance Applications. *IEEE Access* **2020**, *9*. [CrossRef]
40. Zhang, Y.; Liu, M. Regional Optimization Dynamic Algorithm for Node Placement in Wireless Sensor Networks. *Sensors* **2020**, *20*, 4216. [CrossRef]
41. Zaixiu, D.; Shang, C.; Chang, C.Y.; Sinha Roy, D. Barrier Coverage Mechanism Using Adaptive Sensing Range for Renewable WSNs. *IEEE Access* **2020**, *8*. [CrossRef]
42. Xu, S.; Ou, Y.; Wu, X. Optimal Sensor Placement for 3-D Time-of-Arrival Target Localization. *IEEE Trans. Signal Process.* **2019**, *67*, 5018–5031. [CrossRef]
43. Xu, S. Optimal Sensor Placement for Target Localization Using Hybrid RSS, AOA and TOA Measurements. *IEEE Commun. Lett.* **2020**, *24*, 1966–1970. [CrossRef]

44. Akbarzadeh, V.; Gagné, C.; Parizeau, M. Sensor control for temporal coverage optimization. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 4468–4475. [\[CrossRef\]](#)
45. Zhang, Y.; Liang, R.; Xu, S.; Zhang, L.; Zhang, Y.; Xiao, D. A One-step Pseudolinear Kalman Filter for Invasive Target Tracking in Three-dimensional Space. In Proceedings of the 2021 IEEE International Conference on Real-time Computing and Robotics (RCAR), Xining, China, 15–19 July 2021; pp. 353–358. [\[CrossRef\]](#)
46. Hu, J.; Zhang, C.; Xu, S.; Chen, C. An Invasive Target Detection and Localization Strategy Using Pan-Tilt-Zoom Cameras for Security Applications. In Proceedings of the 2021 IEEE International Conference on Real-time Computing and Robotics (RCAR), Xining, China, 15–19 July 2021; pp. 1236–1241. [\[CrossRef\]](#)
47. Wang, S.; Guo, Q.; Xu, S.; Su, D. A Moving Target Detection and Localization Strategy Based on Optical Flow and Pin-hole Imaging Methods Using Monocular Vision. In Proceedings of the 2021 IEEE International Conference on Real-time Computing and Robotics (RCAR), Xining, China, 15–19 July 2021; pp. 147–152. [\[CrossRef\]](#)
48. Pedrollo, G.; Konzen, A.A.; de Moraes, W.O.; Pignaton de Freitas, E. Using Smart Virtual-Sensor Nodes to Improve the Robustness of Indoor Localization Systems. *Sensors* **2021**, *21*, 3912. [\[CrossRef\]](#)
49. De Rainville, F.M.; Mercier, J.P.; Gagné, C.; Giguère, P.; Laurendeau, D. Multisensor placement in 3D environments via visibility estimation and derivative-free optimization. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 3327–3334. [\[CrossRef\]](#)
50. Herguedas, R.; López-Nicolás, G.; Sagüés, C. Multi-camera coverage of deformable contour shapes. In Proceedings of the 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE), Vancouver, BC, Canada, 22–26 August 2019; pp. 1597–1602. [\[CrossRef\]](#)
51. Cuiral-Zueco, I.; López-Nicolás, G. RGB-D Tracking and Optimal Perception of Deformable Objects. *IEEE Access* **2020**, *8*, 136884–136897. [\[CrossRef\]](#)
52. Lee, E.T.; Eun, H.C. Optimal Sensor Placement in Reduced-Order Models Using Modal Constraint Conditions. *Sensors* **2022**, *22*, 589. [\[CrossRef\]](#)
53. Zhang, X.; Zhang, B.; Chen, X.; Fang, Y. Coverage optimization of visual sensor networks for observing 3-D objects: Survey and comparison. *Int. J. Intell. Robot. Appl.* **2019**, *3*, 342–361. [\[CrossRef\]](#)
54. Spielberg, A.; Amini, A.; Chin, L.; Matusik, W.; Rus, D. Co-Learning of Task and Sensor Placement for Soft Robotics. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1208–1215. [\[CrossRef\]](#)
55. Zang, S.; Ding, M.; Smith, D.; Tyler, P.; Rakotoarivelo, T.; Kaafar, M.A. The Impact of Adverse Weather Conditions on Autonomous Vehicles: How Rain, Snow, Fog, and Hail Affect the Performance of a Self-Driving Car. *IEEE Veh. Technol. Mag.* **2019**, *14*, 103–111. [\[CrossRef\]](#)
56. Hasirlioglu, S.; Riener, A. Challenges in Object Detection Under Rainy Weather Conditions. *Intelligent Transport Systems, From Research and Development to the Market Uptake*; Ferreira, J.C., Martins, A.L., Monteiro, V., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 53–65.
57. Garg, K.; Nayar, S.K. Vision and Rain. *Int. J. Comput. Vis.* **2007**, *75*, 3–27. [\[CrossRef\]](#)
58. Hasirlioglu, S.; Riener, A. A General Approach for Simulating Rain Effects on Sensor Data in Real and Virtual Environments. *IEEE Trans. Intell. Veh.* **2020**, *5*, 426–438. [\[CrossRef\]](#)
59. Shapiro, F.R. The position of the sun based on a simplified model. *Renew. Energy* **2022**, *184*, 176–181. [\[CrossRef\]](#)
60. Stec, B.; Susek, W. Theory and Measurement of Signal-to-Noise Ratio in Continuous-Wave Noise Radar. *Sensors* **2018**, *18*, 1445. [\[CrossRef\]](#) [\[PubMed\]](#)
61. Wang, Y.; Chu, W.; Fields, S.; Heinemann, C.; Reiter, Z. Detection of Intelligent Intruders in Wireless Sensor Networks. *Future Internet* **2016**, *8*. [\[CrossRef\]](#)
62. Nawa, N.E.; Furuhashi, T. Fuzzy system parameters discovery by bacterial evolutionary algorithm. *IEEE Trans. Fuzzy Syst.* **1999**, *7*, 608–616. [\[CrossRef\]](#)
63. Botzheim, J.; Hámori, B.; Koczy, L.; Ruano, A. Bacterial algorithm applied for fuzzy rule extraction. In Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems, Annecy, France, 1–5 July 2002; pp. 1021–1026.
64. Botzheim, J.; Drobnics, M.; Koczy, L. Feature selection using bacterial optimization. In Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems, Perugia, Italy, 15–19 June 2004; pp. 797–804.
65. Das, S.; Chowdhury, A.; Abraham, A. A Bacterial Evolutionary Algorithm for automatic data clustering. In Proceedings of the 2009 IEEE Congress on Evolutionary Computation, Trondheim, Norway, 18–21 May 2009; pp. 2403–2410. [\[CrossRef\]](#)
66. Luh, G.C.; Lee, S.W. A Bacterial Evolutionary Algorithm for the Job Shop Scheduling Problem. *J. Chin. Inst. Ind. Eng.* **2006**, *23*, 185–191. [\[CrossRef\]](#)
67. Botzheim, J.; Cabrita, C.; Koczy, L.; Ruano, A. Fuzzy Rule Extraction by Bacterial Memetic Algorithms. *Int. J. Intell. Syst.* **2009**, *24*, 312–339. [\[CrossRef\]](#)
68. Bódis, T.; Botzheim, J. Bacterial Memetic Algorithms for Order Picking Routing Problem with Loading Constraints. *Expert Syst. Appl.* **2018**, *105*, 196–220. [\[CrossRef\]](#)
69. Balázs, K.; Botzheim, J.; Koczy, L. *Comparative Investigation of Various Evolutionary and Memetic Algorithms*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 313, pp. 129–140. [\[CrossRef\]](#)

-
70. Zhou, D.; Fang, Y.; Botzheim, J.; Kubota, N.; Liu, H. Bacterial memetic algorithm based feature selection for surface EMG based hand motion recognition in long-term use. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–7. [[CrossRef](#)]
 71. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. 2017. Available online: <https://arxiv.org/pdf/1412.6980.pdf> (accessed on 1 January 2022).