

Article

Energy-Aware Dynamic 3D Placement of Multi-Drone Sensing Fleet

Yawen Luo * and Yuhua Chen

Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204, USA; yuhua.chen@uh.edu

* Correspondence: yluo12@uh.edu

Abstract: Unmanned Aerial Vehicles (UAVs, also known as drones) have become increasingly appealing with various applications and services over the past years. Drone-based remote sensing has shown its unique advantages in collecting ground-truth and real-time data due to their affordable costs and relative ease of operability. This paper presents a 3D placement scheme for multi-drone sensing/monitoring platforms, where a fleet of drones are sent for conducting a mission in a given area. It can range from environmental monitoring of forestry, survivors searching in a disaster zone to exploring remote regions such as deserts and mountains. The proposed drone placing algorithm covers the entire region without dead zones while minimizing the number of cooperating drones deployed. Naturally, drones have limited battery supplies which need to cover mechanical motions, message transmissions and data calculation. Consequently, the drone energy model is explicitly investigated and dynamic adjustments are deployed on drone locations. The proposed drone placement algorithm is 3D landscaping-aware and it takes the line-of-sight into account. The energy model considers inter-communications within drones. The algorithm not only minimizes the overall energy consumption, but also maximizes the whole drone team's lifetime in situations where no power recharging facilities are available in remote/rural areas. Simulations show the proposed placement scheme has significantly prolonged the lifetime of the drone fleet with the least number of drones deployed under various complex terrains.

Keywords: 3D placement; coverage problem; terrains; drone; remote sensing; energy efficiency



Citation: Luo, Y.; Chen, Y. Energy-Aware Dynamic 3D Placement of Multi-Drone Sensing Fleet. *Sensors* **2021**, *21*, 2622. <https://doi.org/10.3390/s21082622>

Academic Editors: Anna Guerra, Francesco Guidi, Davide Dardari and Petar M. Djuric

Received: 5 March 2021
Accepted: 2 April 2021
Published: 8 April 2021

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



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As Unmanned Aerial Vehicles (UAVs) are getting more attractions due to their mobility, convenience and low costs, both industry and academia have investigated and deployed drone-related applications and services [1–4]. UAV-based remote sensing is one of those promising applications [5–7]. Drones as camera/sensor platforms offer key advantages over traditional spaceborne and aircraft methods. Compared to satellite observations, the drone-based systems are more capable of having high spatial resolution images. In addition, drone flights can be scheduled whenever the weather and hardware equipment are permitted, as opposed to spaceborne sensing which requires that the satellite orbits back to where the site of interest is covered [8]. As for systematic aerial photography through flights, it requires more resources including aircraft, pilots as well as photographers and has more constraints such as speeds and altitudes.

Due to the above-mentioned capabilities of drone-based sensing systems, drones find many promising applications in crop science, environmental studies, remote sensing and archaeological research [9–11].

1.1. Related Work

Most researchers focus on planning the flight trajectory of a single drone to serve the sensing/monitoring mission. In [12], Holness et al. let DJI Phantom 2 Vision+ fly linear transects, which were oriented over the area of interest and allowed for the collection

of overlapping imagery. They treated overlapping as a necessary requirement for post-processing image stitching to create the large high-resolution mosaic needed for analysis. Authors in [13] applied multispectral, thermal and visible light imaging modules simultaneously for acquiring diagnostic information to serve smart farming purposes. The drone ACSL-PF1 flew at the altitude of 100 m with the speed of 4 m s^{-1} over the target area. In [14], a drone scheduling problem in which a drone flew on a fixed path was presented. The authors proposed a dynamic algorithm to optimize the speed of the drone to complete tasks within a certain time and without depleting its battery.

The above works produce good results with applications that only require one-time noncontinuous data, but are not suitable with applications that require monitoring the whole area simultaneously for a certain period of time. Studies on coordinated multi-drone surveillance have gained increasing attention over the past years. The multi-drone surveillance mission requires two critical problems to be resolved: drone placement and battery endurance.

The drone placement is a coverage problem. The work in [15–17] studied the 3D placement of drone base station. Unlike remote sensing using visual cameras, it applies the electromagnetic (EM) wave for communication. The EM wave can be treated as a type of active sensing while the camera is passive sensing. Alzenad et al. [15] applied line-of-sight and non-line-of-sight communication models to find the best height for drones that maximizes the covered area with a fixed transmitting power. It then formulated the coverage problem to a 2D placement strategy. The authors in [16] applied the same model from [15] to do the placement with extra environmental parameters provided: urban, suburban and dense urban. Reference [17] proposed a simplified placement scheme that utilized equal circles without overlapping to improve coverage efficiency at the cost of some uncovered zones.

These studies, however, only considered the given landscape as a smooth horizontal plane. They typically obtained the optimal altitude first, then formulated the coverage problem into a 2D placement strategy. In real-world applications, especially for visual sensing, each drone's coverage varies tremendously as topography changes, which complicates the placement.

The battery endurance plays an important role in designing the drone sensing fleet. Naturally, flying drones have limited battery supplies which are supposed to cover physical motions, message transmissions and data calculation. Since the surveillance job demands a drone fleet to hold its position for certain periods, the drone energy model needs to be explicitly investigated and necessary adjustments should be deployed on drone locations. Some existing drone-related papers took energy efficiency into account and tried to minimize the power of each drone. In paper [15], authors tried to find the smallest coverage circle while supporting the same amount of clients, which saved the drone's transmitting power. Authors in [18,19] managed to find an optimal travel path to improve energy efficiency.

However, these only consider the total energy consumption without evaluating the impact of bottleneck drones on the lifetime of the drone fleet, which is an important factor in the operations of real-world applications. In addition, the fact that drones typically operate in geographically challenging areas is often ignored. Note that in our paper, the lifetime of the drone fleet is defined as the time of the first drone running out of power, which is widely adopted [20,21].

1.2. Our Contribution

To address these issues, in this paper, we propose a 3D placement algorithm for the multi-drone sensing fleet, which targets large areas with continuous omnidirectional surveillance requirements and various geographical conditions. The sensing technology ranges from visible light, multispectral, hyperspectral camera to EM transceivers. Potential applications include the assistance of rescuing missions in flooded regions, temporary

recovery of telecommunication systems in earthquake areas and environmental monitoring of areas with rough terrains.

In the 3D placement problem, we consider the visual blocking situation for each drone hovering over various terrains. To save the overall energy consumption, we analyze the drone's energy model and study the impact of mutual distances among the drones. Considering the finite amount of energy sources as well as the fact that each drone has to work in coordination with remaining drones to share information as required by the application, the lifetime of the entire drone fleet is strongly affected by the lifetime of individual drones, especially those in critical locations. In order to balance the entire network's energy, we design the routing algorithm that chooses the relaying neighbors and selects the data path to reach the destination. In addition, dynamic adjustments are applied to drones' locations during the mission to prolong the drone fleet's lifetime.

In our work, we have the following assumptions: The geographical conditions of the investigated area are known with its 3D dimensional contour map given. There are no charging facilities or battery replacements available. Compared to other research, our work contributes in the following ways: (1) We study the placement of a multi-drone sensing problem that requires continuous omnidirectional monitoring. (2) Our model considers topographical factors, including rough terrain situations where mountains, valleys or other obstacles exist. (3) Our 3D placement, routing scheme and real-time location adjustments aim at improving individual energy efficiency while prolonging the lifetime of the whole drone fleet at the same time.

The rest is organized as follows: First, the system architecture is described in Section 2. More specifically, the system model and energy model are elaborated. Next, research methods are illustrated in Section 3, which shows the details of the mesh simplification, sub-section division, drone placement, routing scheme and dynamic adjustments. Then, Section 4 presents simulation results and evaluates the performance. Finally, Section 5 draws the conclusion.

2. System Architecture

2.1. System Model

For simplicity, the target area T is defined as a square with a side length of a . The terrains are generated by translating and scaling Gaussian distributions. A set of drones $D = \{D_1, D_2, D_3, \dots, D_n\}$ are responsible for the area of interest. The number n is kept as small as possible to serve the monitoring job. The whole surveillance area is supposed to be covered by the drone fleet's cameras. Based on the requirements of different applications, the camera can be visible light, multispectral, hyperspectral and even be replaced by EM transceivers. Since our focus is on the placing algorithm and the mission duration, we simplify this model with the most common visible light camera HD 1080p. θ_h and θ_v are the camera's horizontal and vertical field of view (FOV), respectively. Figure 1 illustrates the drone camera's FOV and shows its coverage during the mission execution. χ (pixels m^{-1}) is the image resolution parameter required by the surveillance task, e.g., the requirements of χ vary greatly from object identification, motion detection to object tracking. The objectives of the placement algorithm include:

- building a regional coverage model;
- determining the least number of drones needed for coverage in a given area;
- finding optimal drones' 3D-placement locations to save communication energy;
- proposing a strategy to prolong the entire drone fleet's lifetime.

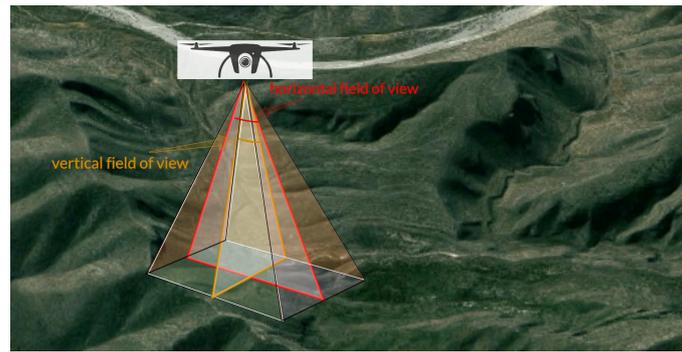


Figure 1. The drone camera's field of view.

2.2. Energy Model

The energy consumption of the drone D_i during the mission execution process is given by:

$$E_{D_i} = E_{flight} + E_{cal} + E_{com}, \quad (1)$$

where E_{flight} , E_{cal} and E_{com} are the energy consumed during the flight, calculation and communication modes, respectively.

2.2.1. Flight Mode

Even without any communication cost, drones need to consume power to maintain flying status, which consumes the majority of the battery energy. Drone's mechanical energy involves two processes, namely, hover and motion. It can be written as:

$$E_{flight} = E_{hov} + E_{mov}. \quad (2)$$

The hover energy E_{hov} can be calculated through:

$$E_{hov} = P_{hov} \times t_{hov}, \quad (3)$$

where t_{hov} is the time duration of the hover status, and P_{hov} is the hover power. According to the model in [18,22], P_{hov} is a function of the drone's mass m , the number of propellers n as well as the propeller's radius r :

$$P_{hov} = \sqrt{\frac{(mg^3)}{2\pi r^2 n \rho}}, \quad (4)$$

where g and ρ are the Earth's gravitational acceleration and the air density, respectively. As stated in Equation (4), we want to have a lighter drone with more and larger propellers to achieve low P_{hov} . It is a paradox since larger numbers and sizes of propellers will incur larger mass, which goes against the "lightweight" drones. In addition, there is another antinomy: in the real world, manufacturers intend to equip drones with high-capacity batteries for longer flight time; however, the mass of the battery itself has a linear relationship with its charging capacity.

The above constraints limit the drone's flight duration based on today's technology. New low-density materials and high-volume batteries are the technical bottlenecks for energy efficiency research in the flight mode, which is in the domain of mechanical and chemical engineering. We apply the latest commercial drone's parameters as the basis of our following analysis.

As for the motion energy, it can be obtained by:

$$E_{mov} = \int P_{mov}(t) dt, \quad (5)$$

where P_{mov} is the motion power that measures the “moving” process from one location to another and can be modeled as a linear function of the drone speed v [18]:

$$P_{mov}(t) = \frac{P_{max} - P_{idle}}{v_{max}}v(t) + P_{idle}, \quad (6)$$

where P_{max} and P_{idle} represent the power of drone when moving at the maximum speed v_{max} and holding the hover status, respectively. To some extent, P_{idle} is equal to $P_{hov} \cdot P_{max}$ and v_{max} are usually known drone parameters. The speed v is time-variant and is subject to:

$$d = \int v(t)dt, \quad (7)$$

where d is the distance between the departure and destination locations.

2.2.2. Calculation Mode

The calculation energy E_{cal} can be written as:

$$E_{cal} = \int P_{cal}(t)dt, \quad (8)$$

where P_{cal} is the power consumption of computation. These calculations range from image processing, pattern identification, object tracking to decision making and event reporting. The critical technologies that impact energy efficiency are the calculation algorithms and data structures that are related to specific tasks and applications. In this paper, we do not restrict the type of applications of the drone fleet. However, for simulation purposes, P_{cal} is simplified as a constant and the detail is shown in the Section 4.

2.2.3. Communication Mode

The energy spent in message sharing among drones made up another significant portion of battery consumption. For example, in some applications, surveillance area videos should be transmitted to the server where groups of experts and officials can make real-time decisions. It incurs a large amount of data to be sent from drones' antennas, which makes the batteries drain faster and it might even exceed the energy consumption of the flight mode. Even for cases where there is no need for real-time video transmissions, the critical event reporting messages and nearby drone information sharing packets can still involve a large energy consumption when the communication distance is far. As we do not restrict the type of applications of the drone fleet, we focus on 3D placement schemes that optimize energy spent in the communication mode.

The communication energy is:

$$E_{com} = E_{tx} + E_{rx}, \quad (9)$$

where E_{tx} and E_{rx} are the energy required for the transmitter and the receiver, respectively. To simplify the energy consumption during the radio communication process, we adopt the model as depicted in [23]. For transmitter or receiver circuitry, the radio transmission dissipates E_{elec} (nJ bit⁻¹). The E_{amp} (nJ bit⁻¹ m⁻²) is defined for the transmitter amplifier's consumption. The energy to transfer a message of l_{packet} bits over distance d is:

$$E_{tx}(l_{packet}, d) = l_{packet} \times E_{elec} + l_{packet} \times E_{amp} \times d^2. \quad (10)$$

The energy consumption to receive the packet is:

$$E_{rx}(l_{packet}, d) = l_{packet} \times E_{elec}. \quad (11)$$

3. Research Methods

We define $f(x, y, z) = 0$ as the terrain function. The location of the drone D_i is denoted as $L_{D_i}(x_{D_i}, y_{D_i}, z_{D_i})$. $A_{D_i} = \{P_i | (x_{P_i}, y_{P_i}, z_{P_i}) \in R^3\}$ labeled as the covering region of drone D_i , is the collection of all points where each $P_i(x_{P_i}, y_{P_i}, z_{P_i})$ can be detected by drone D_i . As a result that the sensing camera is HD 1080p (1920p \times 1080p), with the task resolution requirement χ (pixels m^{-1}) known, we can obtain the maximum region a drone can supervise by:

$$l = \frac{1920}{x}, \quad (12)$$

and

$$w = \frac{1080}{y}, \quad (13)$$

where l and w represent the length in the X axis (east–west) and the width in the Y axis (north–south), respectively. To meet the resolution requirement, A_{D_i} is constrained by:

$$A_{D_i} \subseteq R_{l \times w}, \quad (14)$$

where $R_{l \times w}$ represents an $l \times w$ rectangular area. The horizontal and vertical FOV θ_h, θ_v are constrained by:

$$\frac{\tan \frac{\theta_h}{2}}{\tan \frac{\theta_v}{2}} = \frac{1920}{1080} \approx 1.78. \quad (15)$$

For convenience, we assume that the drone's horizontal FOV is always along the X axis and the vertical FOV is corresponds to the Y axis. Considering the light blockage, the coverage problem can be formulated as follows: For any $P_i(x_{P_i}, y_{P_i}, z_{P_i}) \in f(x, y, z) = 0$, we say P_i is under the surveillance of drone D_i , if

$$|z_{P_i} - z_{D_i}| \times \tan \frac{\theta_h}{2} \geq |x_{P_i} - x_{D_i}|, \quad (16)$$

and

$$|z_{P_i} - z_{D_i}| \times \tan \frac{\theta_v}{2} \geq |y_{P_i} - y_{D_i}|, \quad (17)$$

and no other $P_j(x_{P_j}, y_{P_j}, z_{P_j}) \in f(x, y, z) = 0$ that satisfies Equations (16) and (17) as well as:

$$0 < \frac{x_{P_j} - x_{D_i}}{x_{P_i} - x_{D_i}} = \frac{y_{P_j} - y_{D_i}}{y_{P_i} - y_{D_i}} = \frac{z_{P_j} - z_{D_i}}{z_{P_i} - z_{D_i}} < 1. \quad (18)$$

3.1. Terrain Simplification

In real-world surveillance tasks, the target region can stretch for several miles. Time and space complexity are critical bottlenecks for the coverage problem and the drone placement scheme. Considering the target area's size, it is not possible or even necessary to test feasible drone locations point by point. We only need to try the particular coordinates which are iconic "landmarks". We simplify the terrains as illustrated in Figure 2. Only the space above the simplified vertices is explored. The given topography of the target area T is a height map $H = \{h_{i,j}(x_i, y_j) | (x_i, y_j) \in R^2\}$.

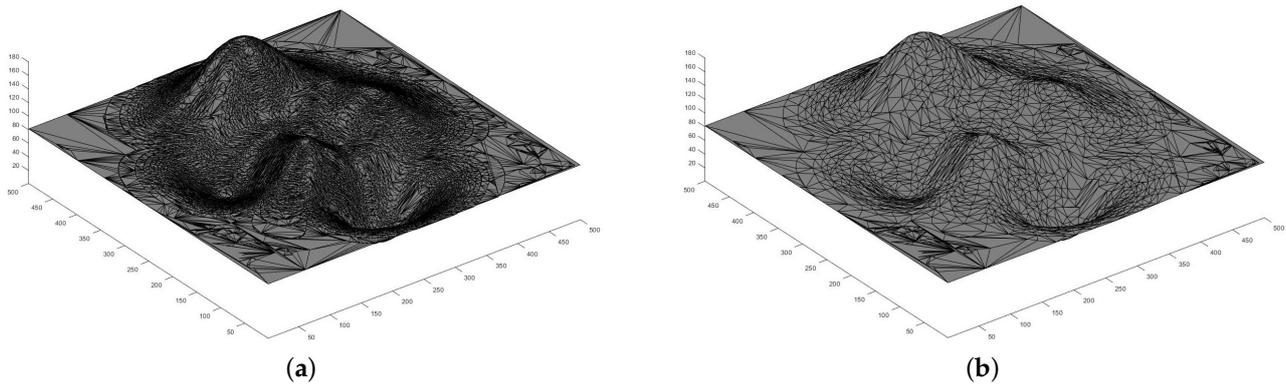


Figure 2. The results of the terrains' simplification: (a) the terrain before simplification, and (b) the terrain after simplification.

Simplification is performed as follows:

- transfer height map H into a 3D binary volume array $V = \{v_{i,j,k}(x_i, y_j, z_k) | (x_i, y_j, z_k) \in R^3\}$;
- extract the isosurface of the volumetric array V and store the faces and vertices information as FV ;
- apply a mesh decimation algorithm to simplify the surface and obtain faces and vertices as SFV .

The isosurface extraction and patch reduction algorithm are mature [24–26] and can be found in many packages or libraries. One example using MATLAB library is shown in Algorithm 1 for other researchers to replicate our works. In addition, some of the notations in the algorithm are used in later discussion.

Algorithm 1: Terrain Simplification: Calculate SFV

Require: $h_{ij}(x_i, y_j) > 0, \forall h_{ij} \in H$

Ensure: $SFV \neq \emptyset$

```

1:  $V \leftarrow 0$ 
2: for  $i = 1$  to  $a$  do
3:   for  $j = 1$  to  $a$  do
4:     for  $k = 1$  to  $h_{ij}$  do
5:        $V(i, j, k) \leftarrow 1$ 
6:     end for
7:   end for
8: end for
9:  $FV \leftarrow \text{isosurface}(V)$ 
10:  $SFV \leftarrow \text{reducepatch}(FV, \text{ratio})$ 

```

3.2. Area Division

The given area T is supposed to be covered by the drone fleet, where each drone will be assigned to a small section S_{D_i} . We denote the number of sections as n . In papers where a flat plane is assumed for the whole area T , the division problem is usually simplified. More specifically, every drone will be distributed with an equal rectangular shape. However, due to the sophisticated terrains in our study, the above algorithm does not apply to our system model. As shown in Figure 3, the coverage of a drone can be stretched when placed over the valley (green line) and shrunk when near the mountain (red line). Hence, we propose our division algorithm, which detects the terrain variances and automatically adjusts each subarea's range so that the drone can cover as large a region as possible, with the least number of drones needed for the mission. The objective of sub-section division is formulated as:

$$\min(n), \quad (19)$$

subject to:

$$\bigcup_{i=1}^n S_{D_i} = T, \quad (20)$$

and

$$S_{D_i} \subseteq A_{D_i}, \forall i \in (1, 2, \dots, n). \quad (21)$$

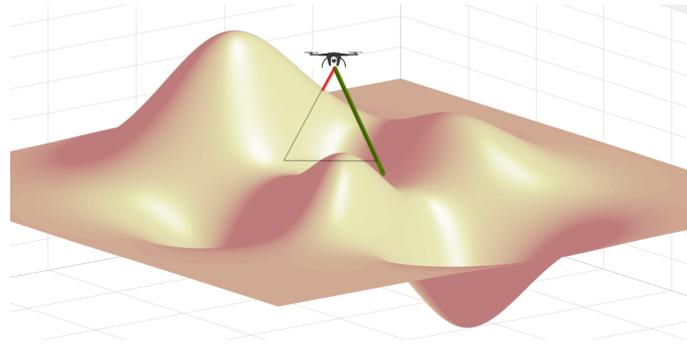


Figure 3. The coverage of the drone is impacted by topography's variations.

Since the coverage area S_{D_i} is a variable and there is no explicit math model to follow, we adopt the greedy algorithm. We perform a search on the remaining uncovered area to find the largest region for each S_{D_i} , and a location $L_{D_i}(x_{D_i}, y_{D_i}, z_{D_i})$ that allows A_{D_i} to cover S_{D_i} . As we analyzed earlier, A_{D_i} can be calculated by Equations (12)–(18). However, the terrain function is not available and needs to be inferred first. In Algorithm 1, we have simplified faces and vertices stored as *SFV*. We can fit the vertices $v_{i,j,k}(x_i, y_j, z_k)$ to a function, which approximates the terrain function $f(x, y, z) = 0$ [27,28]. More details are shown in Algorithm 2.

Algorithm 2: Area Division: Calculate S_{D_i}

Require: $T \neq \emptyset$
Ensure: $\min(n)$

- 1: $f(x, y, z) = 0 \Leftarrow \text{fit}(SFV)$
- 2: $n \Leftarrow 0$
- 3: **while** $T \neq \emptyset$ **do**
- 4: $n \Leftarrow n + 1$
- 5: *find the bottom left corner* (x_0, y_0) of T
- 6: $len \Leftarrow l; wd \Leftarrow w$
- 7: **repeat**
- 8: $flag \Leftarrow false$
- 9: $S_{D_n} \Leftarrow \emptyset$
- 10: **for** $\forall (x_i, y_j) \in T$ **do**
- 11: **if** $(x_0 \leq x_i \leq x_0 + len)$ **and** $(y_0 \leq y_j \leq y_0 + wd)$ **then**
- 12: $S_{D_n} \Leftarrow S_{D_n} \cup (x_i, y_j)$
- 13: **end if**
- 14: **end for**
- 15: **for** $\forall (x_i, y_j) \in S_{D_n}$ **do**
- 16: **for** $z_k = h_{i,j}(x_i, y_j)$ **to** z_{max} **do**
- 17: $L_{D_n} \Leftarrow (x_i, y_j, z_k)$
- 18: $A_{D_n} \Leftarrow \emptyset$
- 19: **for** $\forall (x, y, z) \in SFV$ **do**
- 20: **if** (x, y, z) *covered by* D_n **then**
- 21: $A_{D_n} \Leftarrow A_{D_n} \cup (x, y, z)$
- 22: **end if**
- 23: **end for**
- 24: *find* $\max(x), \min(x), \max(y), \min(y)$ of A_{D_n}
- 25: **if** $S_{D_n} \subseteq A_{D_n}$ **and** $(\max(x) - \min(x)) \leq l$ **and** $(\max(y) - \min(y)) \leq w$ **then**
- 26: $flag \Leftarrow true; break$
- 27: **end if**
- 28: **end for**
- 29: **if** $flag == true$ **then**
- 30: $break$
- 31: **end if**
- 32: **end for**
- 33: $len \Leftarrow len - \delta_l; wd \Leftarrow wd - \delta_w$
- 34: **until** $flag == true$
- 35: $T \Leftarrow T \setminus S_{D_n}$
- 36: **end while**

3.3. Placement

In the placement scheme, we aim to save overall energy consumption, which can be written as:

$$\min\left(\sum_{i=1}^n E_{D_i}\right). \quad (22)$$

According to Equations (1)–(11), we know that E_{flight} depends on drone's hardware parameters, and E_{cal} is related to the specific application. We focus on 3D placement schemes that optimize the energy spent in the communication mode E_{com} . Equation (10) shows that distance $d_{i,j}$, between a pair of drones D_i and D_j , has large impacts on the energy consumed at the transmitter E_{tx} . We further rewrite the target Equation (22) as:

$$\min\left(\sum_{i,j=1}^n d_{i,j}^2\right), \quad (23)$$

subject to:

$$d_{ij}^2 = \text{shortest_path}(L_{D_i}, L_{D_j}), \quad (24)$$

where the shortest path algorithm is applied to calculate the distance by considering the fact that to save the transmitting power during communication, drone D_k might be assigned to relay the packets sending from D_i to D_j . More specifically, we adopted Dijkstra's algorithm [29].

In order to decide the drone location $L_{D_i}(x_{D_i}, y_{D_i}, z_{D_i})$, we need to find the collection of all feasible drone locations first. We label the desired collection within subarea S_{D_i} as R_{D_i} . The calculation of L_{D_i} is implemented in Algorithm 3. The main steps are as follows:

1. search R_{D_i} for every sub-area S_{D_i} according to Equations (12)–(18);
2. initialize each sub-area's drone location L_{D_i} with a random element of R_{D_i} ;
3. for each sub-area S_{D_i} , find the best location within R_{D_i} that minimizes the total distance from drone D_i to D_j where $j \in (1, 2, \dots, n), j \neq i$;
4. compare the new drone coordinate set with previous ones;
5. if the differences are greater than a predefined threshold, repeat steps 3 and 4. Otherwise, return current drone fleet coordinates.

Algorithm 3: Placement: Calculate L_{D_i}

```

Require:  $S_{D_i} \neq \emptyset, \forall i \in (1, 2, \dots, n)$ 
Ensure:  $\min \sum_{D_i, D_j \in D} \|L_{D_i} - L_{D_j}\|^2$ 
1: for  $id = 1$  to  $n$  do
2:   for  $\forall (x_i, y_j) \in S_{D_{id}}$  do
3:     for  $z_k = h_{ij}(x_i, y_j)$  to  $z_{max}$  do
4:        $L_{D_{id}} \leftarrow (x_i, y_j, z_k)$ 
5:        $A_{D_{id}} \leftarrow \emptyset$ 
6:       for  $\forall (x, y, z) \in SFV$  do
7:         if  $(x, y, z)$  covered by  $D_{id}$  then
8:            $A_{D_{id}} \leftarrow A_{D_{id}} \cup (x, y, z)$ 
9:         end if
10:      end for
11:      find  $\max(x), \min(x), \max(y), \min(y)$  of  $A_{D_{id}}$ 
12:      if  $S_{D_{id}} \subseteq A_{D_n}$  and  $(\max(x) - \min(x)) \leq l$  and  $(\max(y) - \min(y)) \leq w$  then
13:         $R_{D_{id}} \leftarrow R_{D_{id}} \cup (x_i, y_j, z_k)$ 
14:      end if
15:    end for
16:  end for
17: end for
18: for  $id = 1$  to  $n$  do
19:    $L_{D_{id}} \leftarrow R_{D_{id}}(1)$ 
20: end for
21:  $OldL \leftarrow L$ 
22:  $Dif \leftarrow Threshold + 1$ 
23: while  $Dif \geq Threshold$  do
24:   for  $i = 1$  to  $n$  do
25:      $sum \leftarrow MAX$ 
26:     for  $\forall (x, y, z) \in R_{D_i}$  do
27:        $temp \leftarrow \sum_{j \in (1, 2, \dots, n) \text{ and } j \neq i} \text{Dijkstra}(L_{D_j}, (x, y, z))$ 
28:       if  $temp < sum$  then
29:          $sum \leftarrow temp$ 
30:          $L_{D_i} \leftarrow (x, y, z)$ 
31:       end if
32:     end for
33:   end for
34:    $Dif \leftarrow \|OldL - L\|$ 
35:    $OldL \leftarrow L$ 
36: end while

```

3.4. Dynamic Adjustment

According to Equation (10), the radio transmission power significantly increases with distances. When communication packets are delivered between two remote drones, we prefer to use one or more intermediary drones for relaying. As all the drone's locations are

fixed and known, we use Dijkstra's algorithm to find the shortest path from each drone D_i to D_j . We denote Q as the routing table. $Q_{i,j}$ is the next-hop drone when the packet source is D_i and the destination is D_j . Detailed routing table calculation is illustrated in Algorithm 4.

Algorithm 4: Routing Table: Calculate $Q_{i,j}$

Require: $L_{D_i} \neq \emptyset, \forall i \in N = (1, 2, \dots, n)$
Ensure: $Q_{i,j} \neq \emptyset, \forall i, j \in N$

- 1: **for** $i = 1$ **to** n **do**
- 2: **for** $j = 1$ **to** n **do**
- 3: $Q_{i,j} = j$
- 4: $dis_{i,j} = ||L_{D_i}, L_{D_j}||^2$
- 5: **end for**
- 6: **end for**
- 7: **for** $i = 1$ **to** n **do**
- 8: $d = dis$
- 9: $M \leftarrow N \setminus (i)$
- 10: $temp \leftarrow i$
- 11: **while** $M \neq \emptyset$ **do**
- 12: **for** $\forall j \in M$ **do**
- 13: **if** $d_{i,temp} + d_{temp,j} < d_{i,j}$ **then**
- 14: $d_{i,j} = d_{i,temp} + d_{temp,j}$
- 15: $Q_{i,j} = temp$
- 16: **end if**
- 17: **end for**
- 18: $find\ j\ that\ d_{i,j} = \min_{k \in M} d_{i,k}$
- 19: $M \leftarrow M \setminus j$
- 20: $temp \leftarrow j$
- 21: **end while**
- 22: **end for**

The initial 3D placement ensures optimal overall energy saving. During the mission execution, the drone at the team center is more likely to relay the most packets and have the least energy left. We further propose a dynamic location switching scheme to prolong the drone fleet's lifetime. The main idea is to switch the location of the lowest remaining battery drone with the highest remaining battery drone when a threshold condition is met.

Compared to the initial 3D placement, which is performed before the mission with no constraints in time, memory and energy, the dynamic adjustments are calculated by the drones in real time. Each drone maintains a battery volume table, which stores every drone's remaining energy and will be updated periodically by receiving other drone's broadcast packets. This somehow implies extra energy consumption. However, comparing to other messages and information exchanged among the drones, the remaining battery information can be sent with a single short packet and it has negligible overheads comparing with the potential gains in extending the lifetime of the fleet. In addition, the maintenance costs of a one-dimensional array are negligible compared to the drone's other processing tasks. Considering energy efficiency, only the drone with the highest remaining battery will run the dynamic switching algorithm. Denote the highest energy drone as D_i . By checking its battery volume table, D_i will start to execute the switching strategy as follows. First, D_i calculates the approximate power of each drone through:

$$POWER = \frac{ENERGY_{pre} - ENERGY}{INTERVAL}, \quad (25)$$

where $ENERGY_{pre}$ is the old battery volume table and $ENERGY$ is the newly updated table. $INTERVAL$ is the table update period. Next, D_i computes the ideal maximum lifetime left T_{left} for the drone fleet via:

$$T_{left} = \frac{\sum_i ENERGY_i}{\sum_i POWER_i}. \quad (26)$$

T_{left} is the upper limit, which can only be achieved when the entire drone fleet is balanced and every drone has the same power. Then, D_i checks whether the lowest battery drone (denoted as D_j) needs switching according to:

$$\frac{ENERGY_j}{POWER_i} \leq T_{left}, \quad (27)$$

where the inequality means the battery capacity of D_j is extremely low and a switching process is urgently needed. If the condition in Equation (27) is met, D_i will further compare the movement energy consumption E_{mov} from L_{D_i} to L_{D_j} and E_{sw_ovhd} with the switching benefits. E_{sw_ovhd} is the energy consumption caused by the switching overheads and is calculated by:

$$E_{sw_ovhd} = P_{sw_ovhd} \times t_{sw}, \quad (28)$$

where P_{sw_ovhd} accounts for the extra power needed for switching and t_{sw} is the time needed for the switching process between L_{D_i} and L_{D_j} . As a result that our works do not restrict the type of applications running on the drone fleet or the specific hardware configurations, and the switching overheads are application/machine-dependent, we simplify P_{sw_ovhd} as a constant, which is given in Section 4 and identical to P_{cal} . Finally, as long as the E_{mov} and E_{sw_ovhd} are affordable compared to the savings, drone D_i will initiate a switching request to D_j . Once drone D_j received the request, it will respond according to its application process. If drone D_j is executing a critical program, it will decline the request with a busy response. Drone D_i can wait for a certain period before initiating the next request.

To avoid the cases where the high energy drone D_i keeps switching with other drones, we design in such a way that once it makes a switch, it will initiate a protocol to inform the next highest energy drone to take over the processing role. D_i also broadcasts a message to mark D_j as switched, and D_j will not be taken into account in future switching processes. Details are shown in Algorithm 5.

Algorithm 5: Positions Dynamic Switch: Calculate T_{life}

Require: $ENERGY > 0$
Ensure: T_{life}

- 1: **while** $\min(ENERGY) > 0$ **do**
- 2: update($ENERGY$)
- 3: **if** $ID == \max(ENERGY)$ **then**
- 4: $POWER \leftarrow \frac{ENERGY_{pre} - ENERGY}{INTERVAL}$
- 5: $T_{left} \leftarrow \frac{\sum(ENERGY)}{\sum(POWER)}$
- 6: $j \leftarrow$ lowest battery drone
- 7: **if** $(\frac{ENERGY_j}{POWER_{ID}} \leq T_{left})$ **then**
- 8: Calculate SwitchMoveCost(ID, j)
- 9: Initiate switch protocol
- 10: **if** request accepted **then**
- 11: Inform next highest battery drone
- 12: Broadcast D_j has been switched
- 13: **end if**
- 14: **end if**
- 15: **end if**
- 16: **end while**
- 17: $T_{life} \leftarrow$ Current time

4. Results Evaluation

We conducted simulations with rough terrain profiles using MATLAB and the results are analyzed. The proposed scheme is evaluated by comparing the drone team lifetime with/without the 3D-placement algorithm and dynamic location switching strategy. Table 1 defines the parameters used in the simulation.

Table 1. The simulation parameters.

Parameters	Values	Parameters	Values
Field side length a	3000 m	Drone maximum moving power P_{max}	4 W
Horizontal FOV θ_h	120°	Drone maximum speed v_{max}	25 m s ⁻¹
Resolution requirements χ	1 pixel m ⁻¹	Drone processing power P_{cal}	1 W
Drone's mass m	800 g	Transceiver circuitry energy consumption E_{elec}	50 nJ bit ⁻¹
Number of propellers n	4	Transmitter amplifier energy consumption E_{amp}	100 pJ bit ⁻¹ m ⁻²
Propeller's radius r	12 cm	Drone initial energy storage $ENERGY_{ini}$	34,632 J
Gravitational acceleration g	9.8 m s ⁻²	Communication rate within drones	49.8 Mbit s ⁻¹
Air density ρ	1.225 kg m ⁻³	Drone switching overheads power P_{sw_ovhd}	1 W

To ensure that our model applies to various landscapes, we ran three sets of simulations with different terrains as shown in Figure 4: mountain, valley and complex topography.

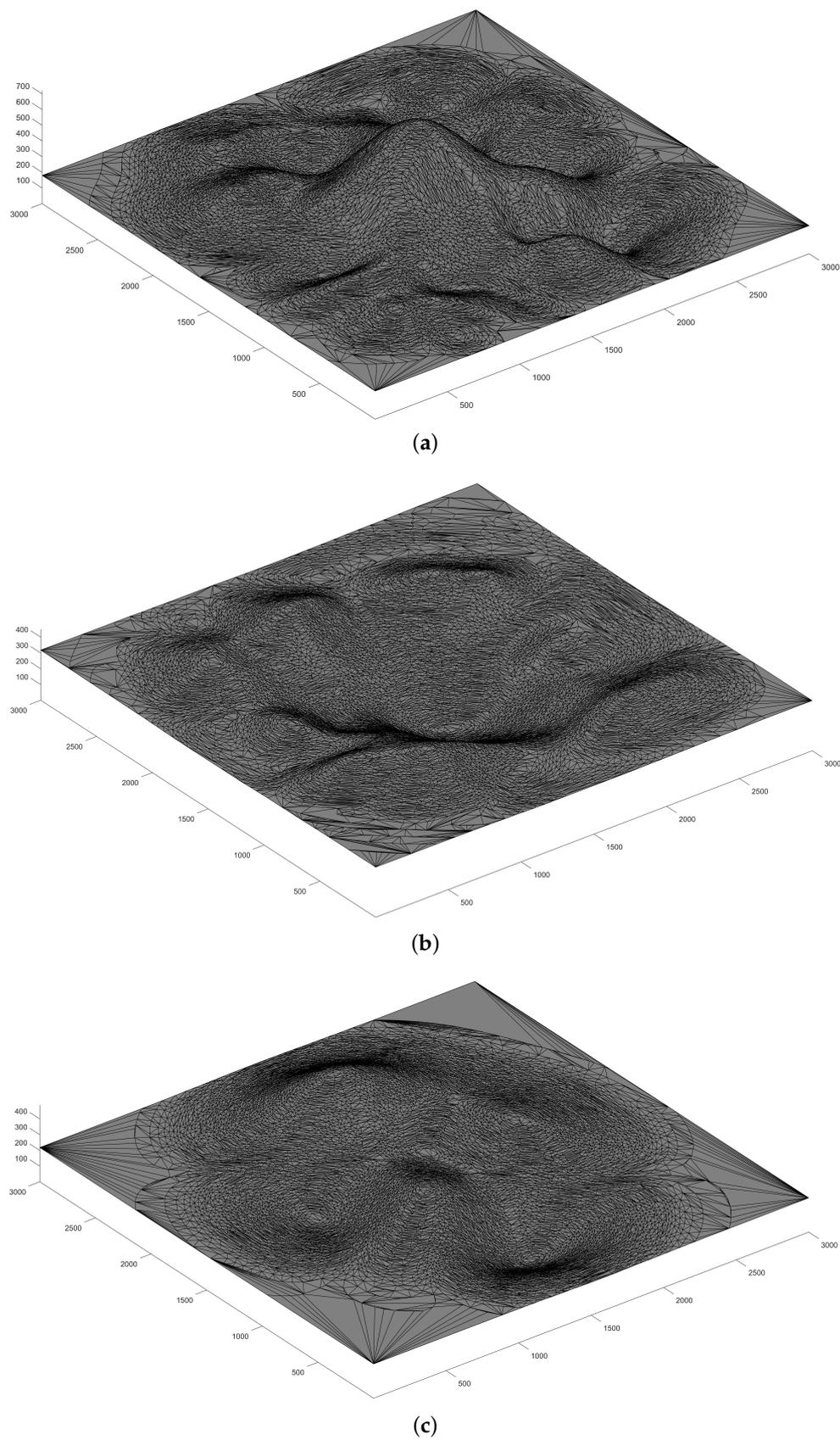


Figure 4. The three different terrains used in our simulations: (a) the mountain topography; (b) the valley topography and (c) the complex topography.

The first part of the simulations deals with the terrains' profile simplification. As described in Algorithm 1, the height map of the target area needs to be converted to a volumetric array and be reduced further in scale. Figure 5 shows the terrains' extraction results. The simplification ratio we applied is 0.0001, which means the number of vertices is approximately reduced by a factor of 10,000. Originally, given the target square area side length as 3000 m, even if we only consider the integer coordinates, there are 9,000,000 possible locations for drone placement in the horizontal plane. After applying the terrain simplification algorithm, there are only 1295, 1196 and 1127 possible locations for mountain, valley and complex topography, respectively. It is clear that the simplified terrain tremendously reduces the number of vertices and edges while preserving critical landmarks and features.

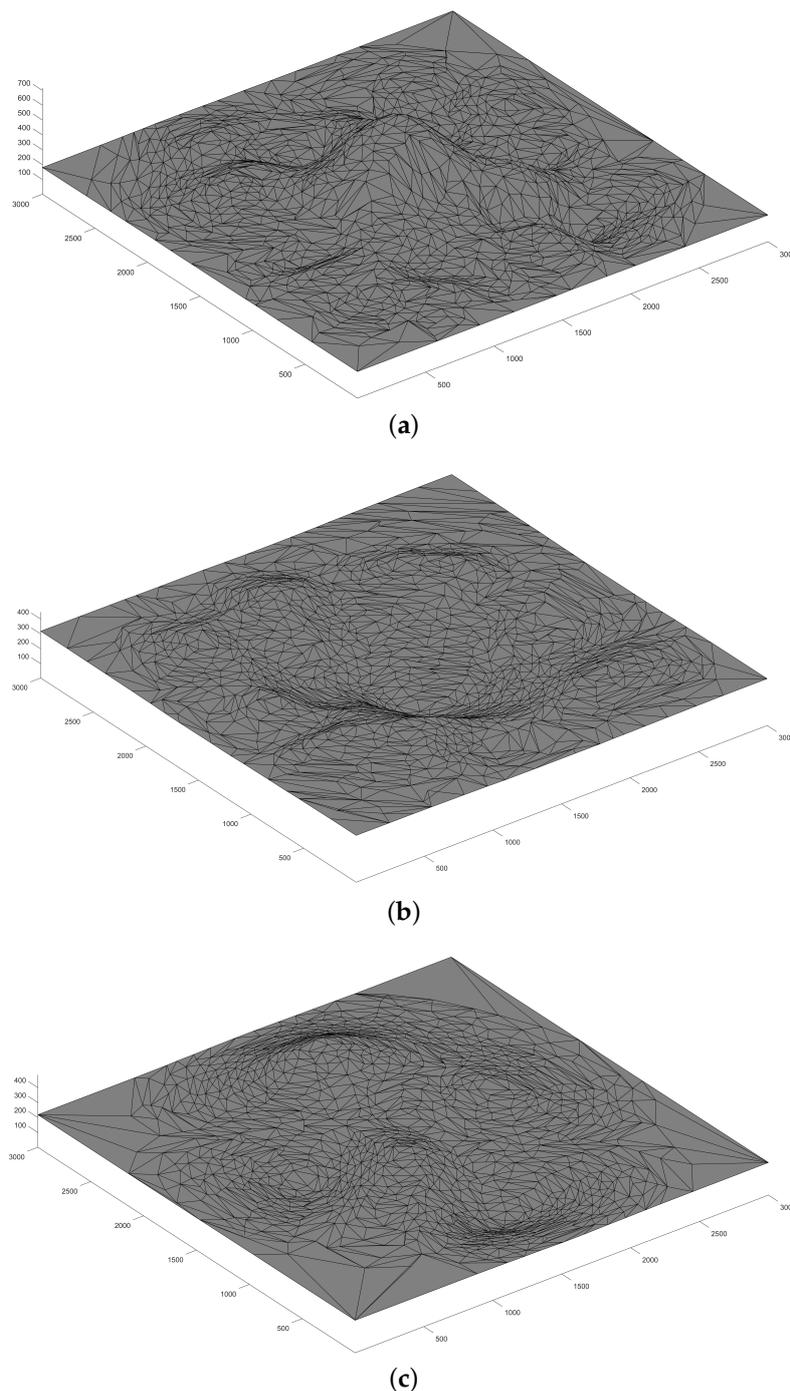
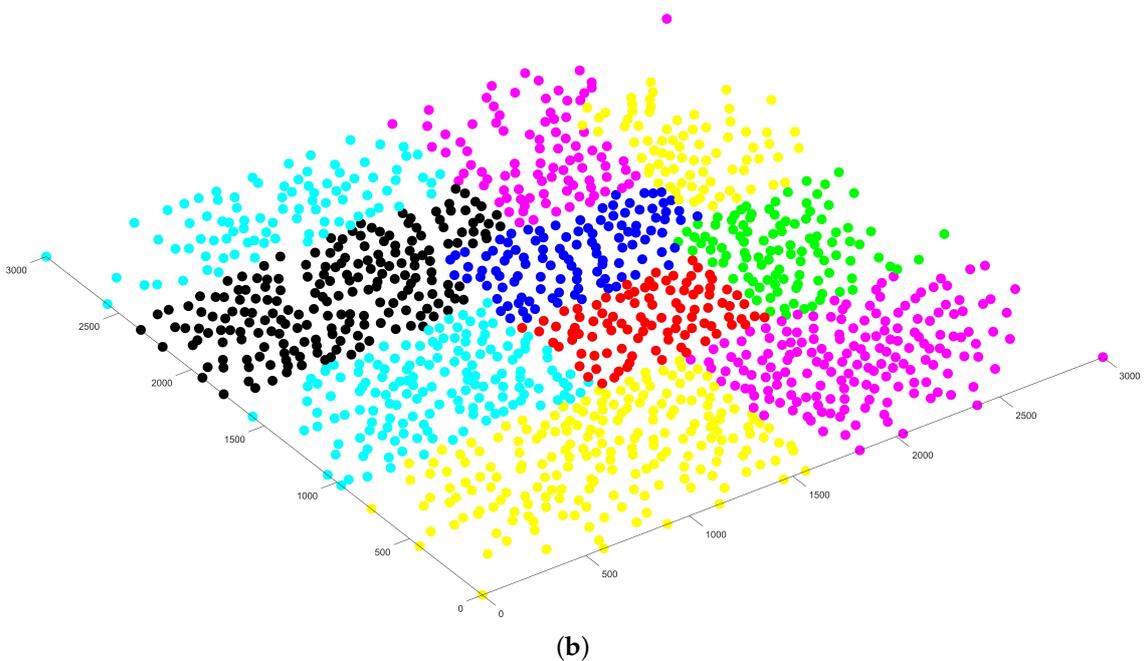
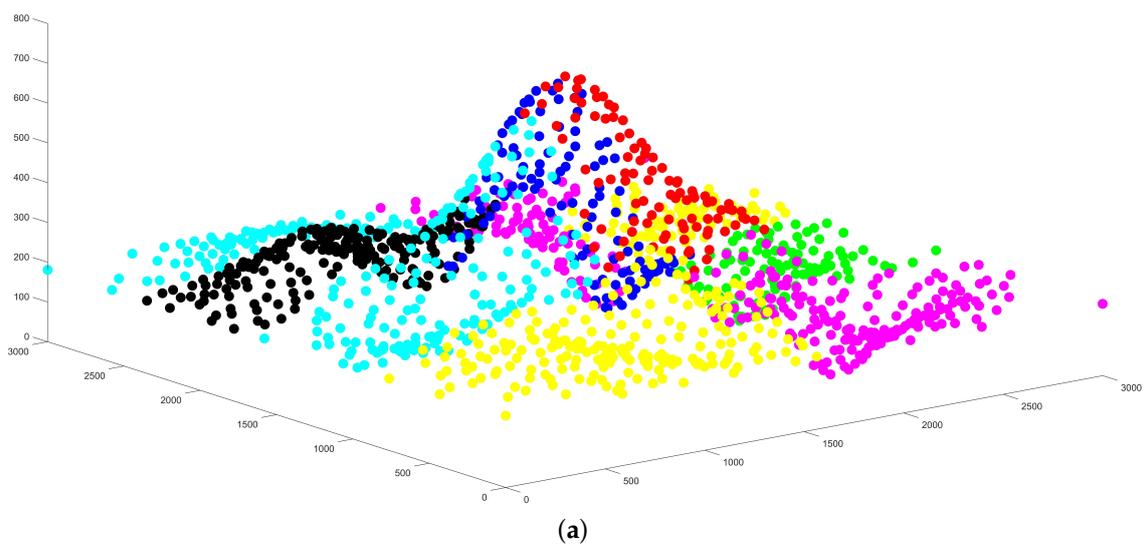
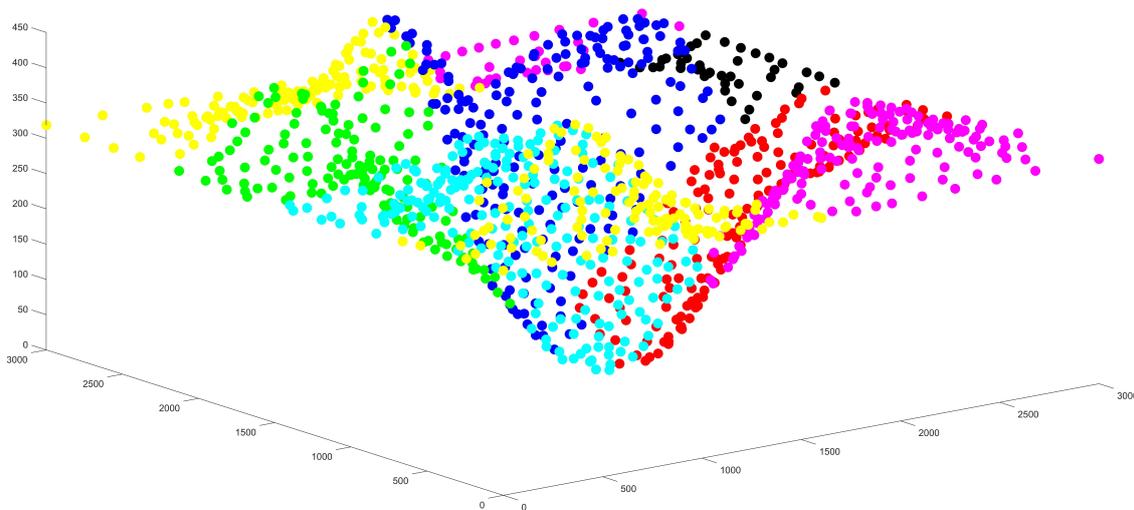


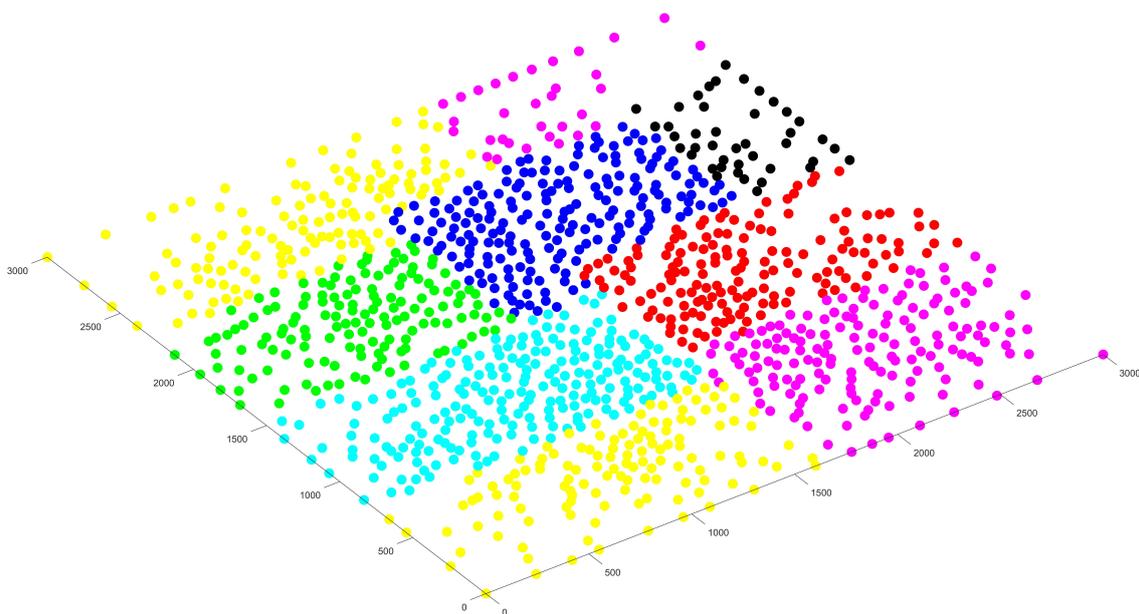
Figure 5. The terrain simplification results of: (a) mountain topography; (b) valley topography and (c) complex topography.

The simulations of the sub-area divisions are immediately followed. The segmentation of various terrains is illustrated in Figure 6. Each sub-region is identified with dots of different colors. For better understanding, each terrain's division results are demonstrated with both aerial and top-down views. By applying Algorithm 2, we segment the target mountain, valley and complex topography area into 10, 9 and 10 sub-areas, respectively. We can discover that the rectangular coverage over the area with steep slopes is relatively small, which is conformed to the theoretical analysis that the highlands are vulnerable to line-of-sight blockage, resulting in a small surveillance region for the drone. Our area division algorithm can adapt to terrain variances and use the least number of drones for the monitoring mission.

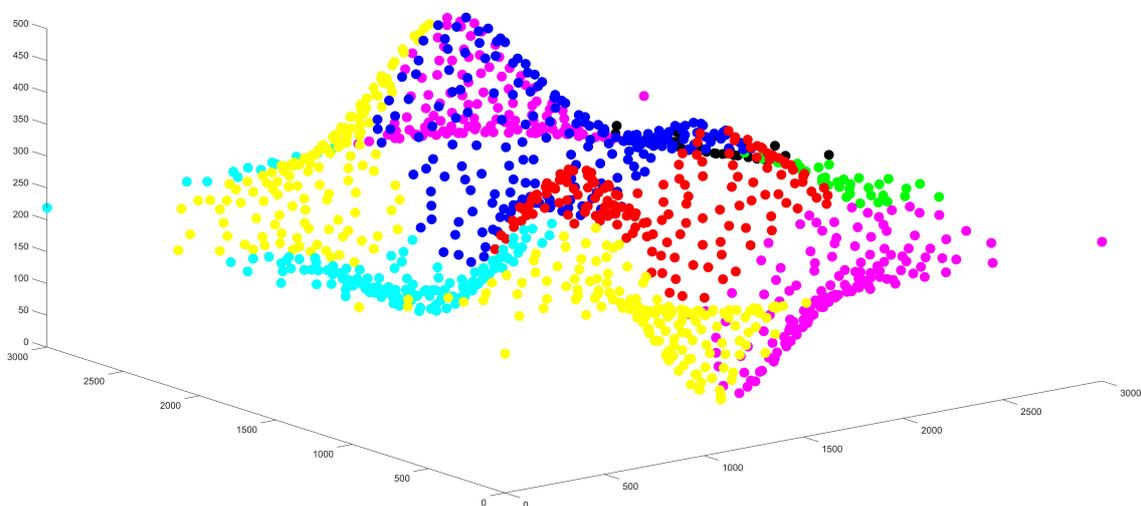




(c)



(d)



(e)

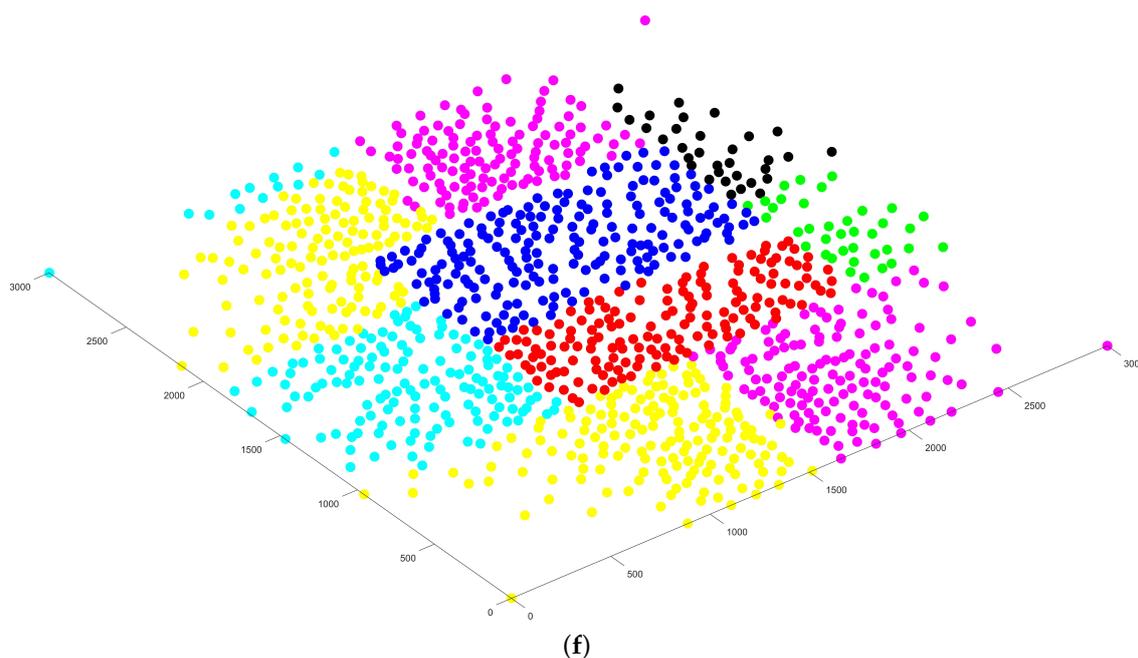
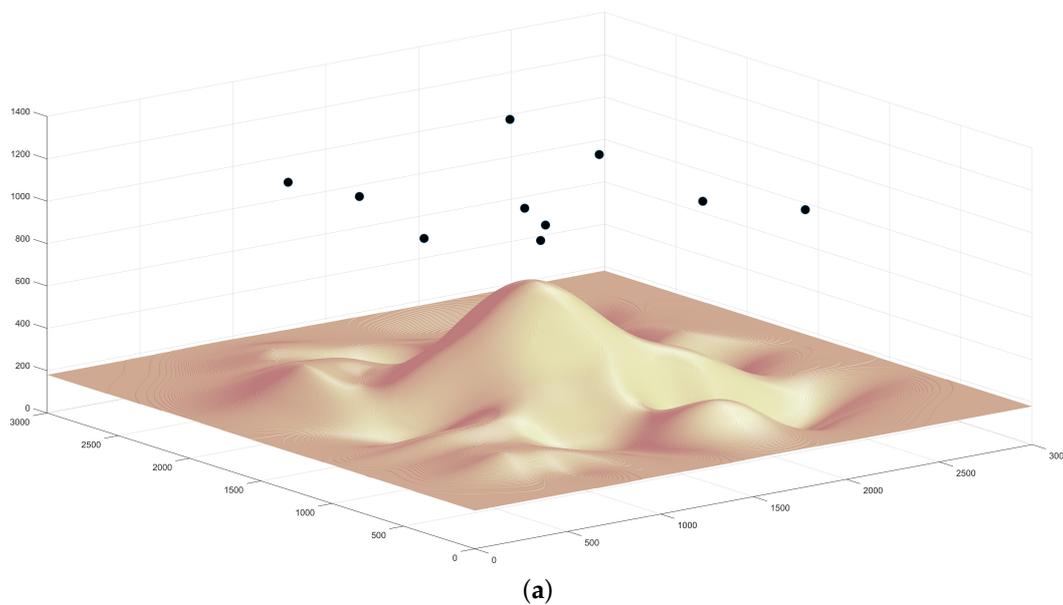
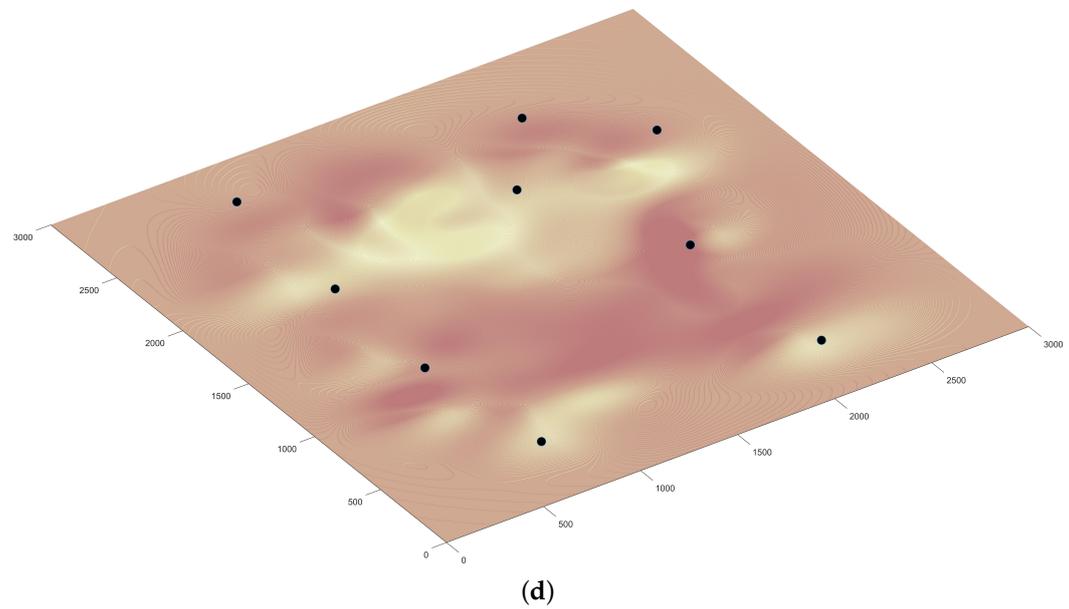
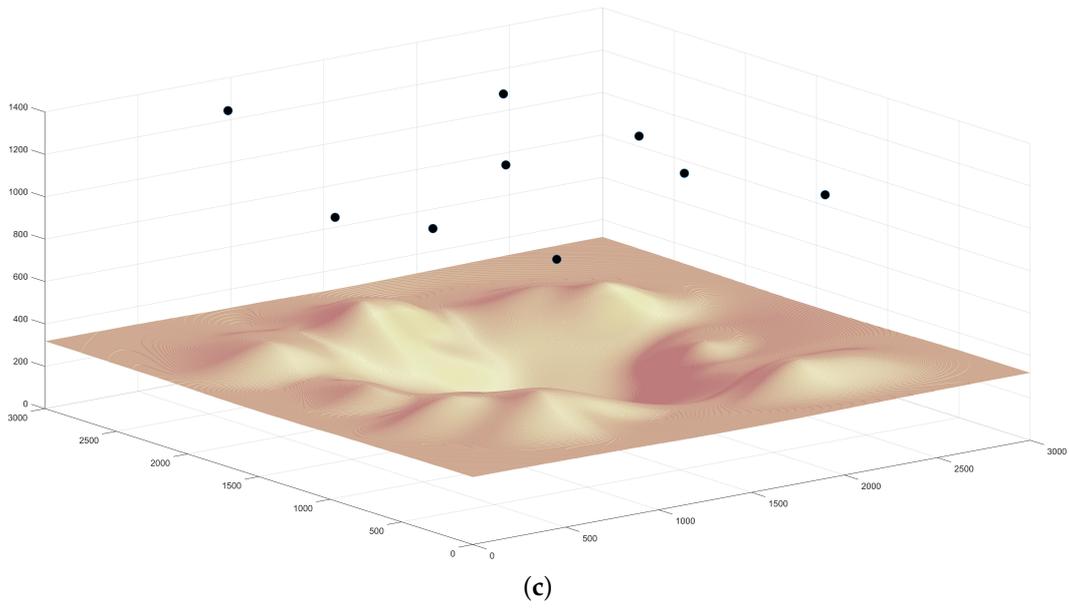
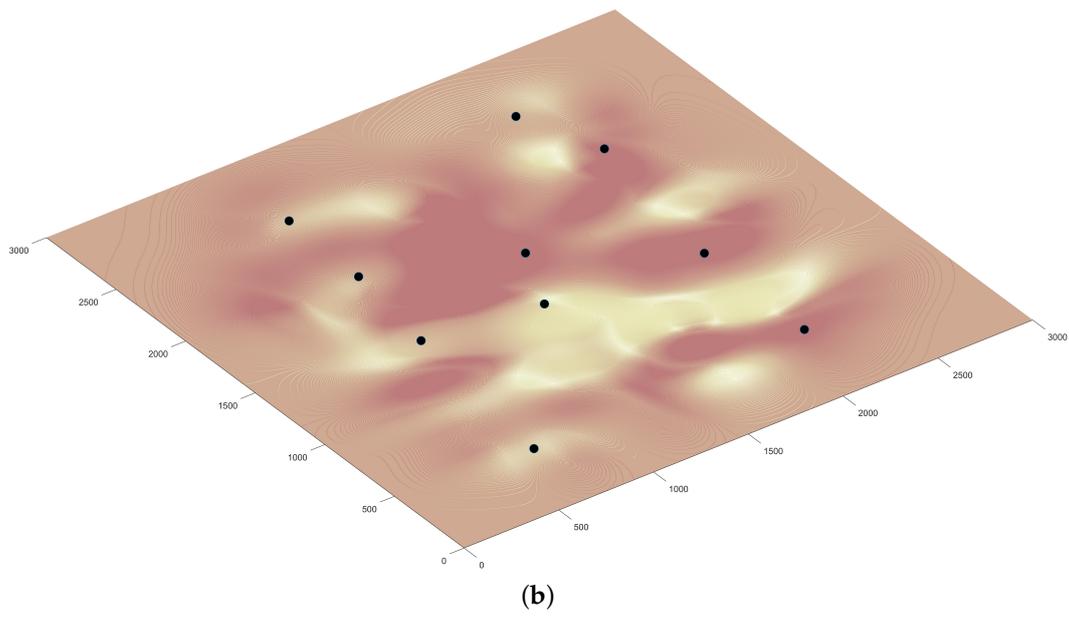


Figure 6. The area division results of: (a) mountain topography in aerial view; (b) mountain topography in top-down view; (c) valley topography in aerial view; (d) valley topography in top-down view; (e) complex topography in aerial view; and (f) complex topography in top-down view.

The following simulations deal with the optimal positioning of the drones in order to cover the target area with energy constraints applied. The final 3D-placement locations of the three different terrains are depicted in Figure 7. The black dots represent the drones' positions over the target zone. A more detailed drone placement coordinates for the three terrains are illustrated in Table 2.





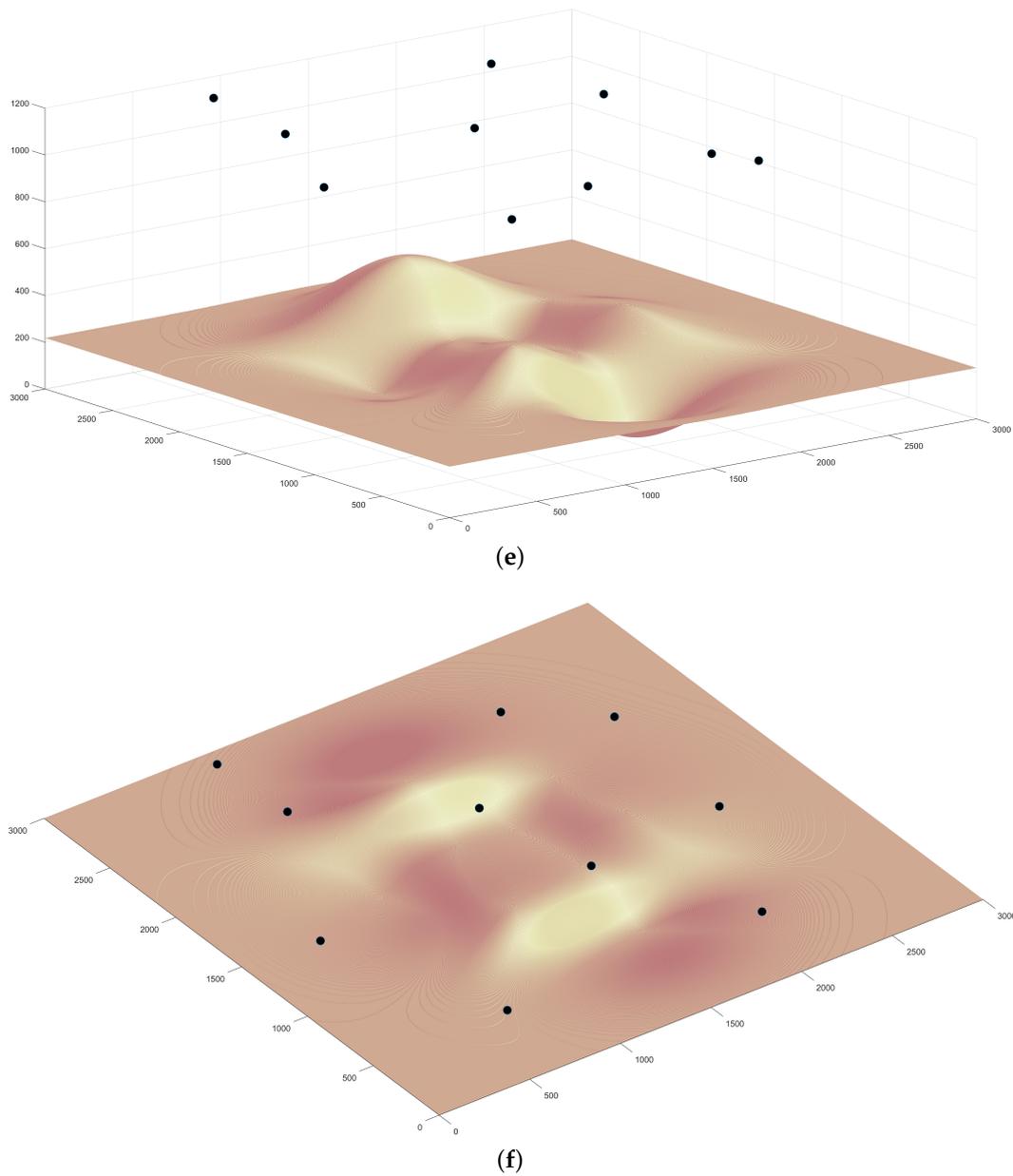
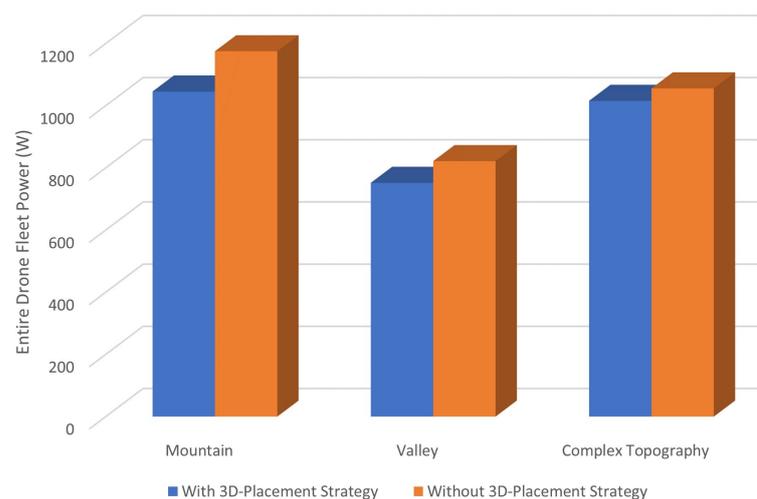


Figure 7. The 3D placement results of: (a) mountain topography in aerial view; (b) mountain topography in top-down view; (c) valley topography in aerial view; (d) valley topography in top-down view; (e) complex topography in aerial view; and (f) complex topography in top-down view.

Table 2. The drone fleet's locations of three different topographies.

Drone Location	Mountain	Valley	Complex
L_{D_1}	(698, 448, 1245)	(778, 425, 1127)	(751, 516, 1074)
L_{D_2}	(2174, 515, 1134)	(2210, 409, 1200)	(2144, 500, 1132)
L_{D_3}	(809, 1410, 1031)	(691, 1180, 1125)	(410, 1465, 1085)
L_{D_4}	(1401, 1330, 1014)	(2166, 1339, 1110)	(1748, 1251, 941)
L_{D_5}	(2183, 1246, 1016)	(721, 1905, 1018)	(2496, 1308, 964)
L_{D_6}	(1571, 1697.5, 987)	(1797, 2110, 1045)	(1621, 1926, 1083)
L_{D_7}	(890, 1969, 1096)	(2553, 2161, 1046)	(2548, 2176, 1052)
L_{D_8}	(2322, 2153, 1020)	(757, 2703, 1345)	(911, 2405, 1070)
L_{D_9}	(2168, 2579, 1120)	(2130, 2561, 1229)	(2160, 2505, 1176)
$L_{D_{10}}$	(909, 2494, 1048)	N/A	(885, 2903, 1136)

To evaluate our placement algorithm, we further simulated drone task processing and packet transmissions with the energy model applied. Equation (22) indicates that our placement algorithm aims to save the overall drone fleet energy consumption. In order to show the performance of our 3D positioning algorithm, we made a comparison of drone fleet power between our placement and other random placements, as shown in Figure 8. The reason why random placements are applied is because there is no similar placement strategy available due to the terrain complexity, multi-unit coordination and supervision continuity. To the best of the authors' knowledge, our work is the pioneer in the topography-aware 3D placement of a multi-drone fleet. The random placements are performed 10 times and only the best performance (minimum power) is used for comparison with our 3D placement algorithm. Clearly, with our 3D-placement scheme applied, the drone fleet consumes less power under all three terrains. This is because our strategy takes drones' mutual distances into account to minimize the overall communication cost.

**Figure 8.** The drone positioning algorithm performance.

Since our proposed 3D placement algorithm is efficient in saving overall drone fleet power, the following simulation results are based on the 3D placement algorithm. To further extend the duration of drone-based tasks, we applied the dynamic position switching algorithm. Figure 9 shows the detailed performance improvements after the switching algorithm is deployed. Since the lifetime parameter in our simulation results can be greatly

impacted by the drone's initial energy $ENERGY_{ini}$, we use the battery utilization U instead for evaluation purposes. U is obtained by:

$$U = \frac{\sum ENERGY_{ini} - \sum ENERGY_{left}}{\sum ENERGY_{ini}}, \quad (29)$$

where $ENERGY_{left}$ is the drone's remaining energy after the sensing mission is terminated. Of course, large battery utilization is preferred, which indicates a longer lifetime. Obviously, our strategy significantly improved battery utilization. Since the battery utilization is proportional to the drone fleet's lifetime, it is clear that with our dynamic switching scheme applied, the lifetime of mountain topography is increased by 43% (from 55.6 to 79.5). There are 71.3% (from 47.2 to 80.9) improvements for valley topography and complex topography is increased by 59.7% (from 53.1 to 84.8). Due to the unbalanced drone networks, some drones need to relay more data, and some may supervise larger regions and spend more power on data processing. As a result, there are cases when a drone drains all its battery while most of the others have a lot of power left, which causes the entire surveillance mission to be terminated prematurely. Our scheme addresses this problem by dynamically switching low-energy-capacity drones with high-energy-capacity drones.

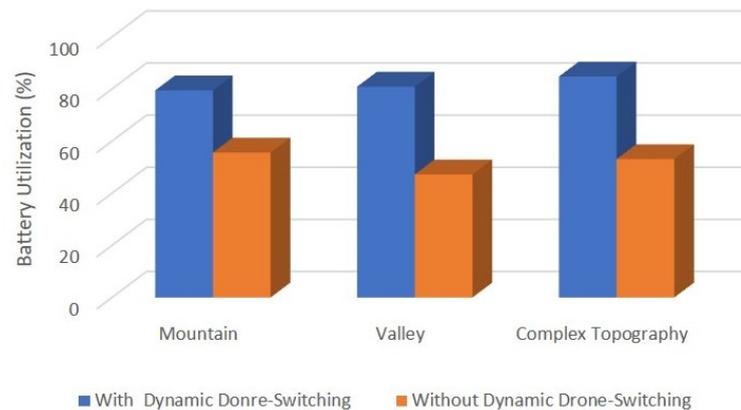


Figure 9. The location dynamic switching algorithm performance.

We also analyzed the communication rate parameter. The data rate of drone's transmission is proportional to the energy consumption in communication. The terrain applied in this set of simulations is the mountain. Figure 10 illustrates the impact of the transmission rate on the placement strategy. The lifetime prolonging ratio (LPR) is defined as:

$$LPR = \frac{lifetime_{3D_DA} - lifetime_{ref}}{lifetime_{ref}}, \quad (30)$$

where $lifetime_{3D_DA}$ is the lifetime with both our 3D placement algorithm and dynamic adjustments algorithm applied. $lifetime_{ref}$ is the lifetime without applying the 3D placement algorithm and dynamic adjustments. It is obvious that our placement scheme is a clear winner, especially as the transmission rate increases.

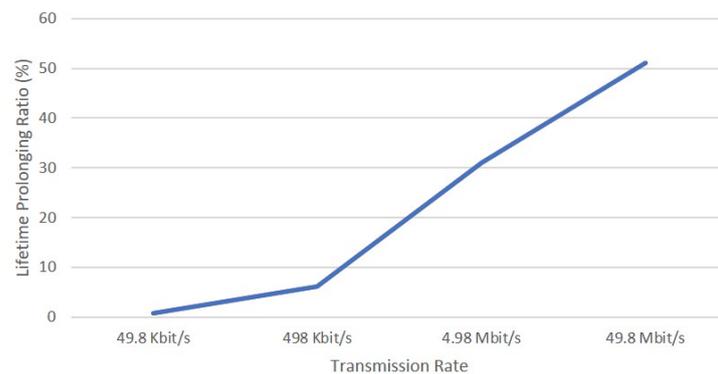


Figure 10. Shows the transmission rate's impacts on placement strategy.

5. Conclusions

In this paper, we studied the dynamic 3D placement of a multi-drone-based sensing system that maximizes the sensing mission time with a minimized number of drones. The target sensing area is of various rough terrains. We analyzed the 3D coverage problem by extracting the terrain features and dividing sub-areas. The drone fleet placement was deployed with energy efficiency taken into account. Moreover, a dynamic position switching algorithm was proposed to prolong the entire drone fleet's lifetime. Simulations have shown that our placement and routing schemes, as well as the dynamic switching algorithms, are effective in improving the lifetime of the fleet. In the future, a more dynamic positioning algorithm will be studied, where the drone can tour within a specific range, which increases the surveillance area and avoids any potential blind spot.

Author Contributions: Conceptualization, Y.L. and Y.C.; methodology, Y.L.; software, Y.L.; validation, Y.L.; formal analysis, Y.L. and Y.C.; writing—original draft preparation, Y.L.; writing—review and editing, Y.C.; visualization, Y.L.; supervision, Y.C.; project administration, Y.C.; funding acquisition, Y.C. 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.

Acknowledgments: The authors would like to thank the computational resources support of Hewlett Packard Enterprise Data Science Institute (HPE DSI), University of Houston.

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

References

1. Buffi, A.; Nepa, P.; Cioni, R. SARFID on drone: Drone-based UHF-RFID tag localization. In Proceedings of the IEEE International Conference on RFID Technology & Application (RFID-TA 2017), Warsaw, Poland, 20–22 September 2017; pp. 40–44.
2. Alwateer, M.; Loke, S.W.; Rahayu, W. Drone services: An investigation via prototyping and simulation. In Proceedings of the IEEE 4th World Forum on Internet of Things (WF-IoT 2018), Singapore, 5–8 February 2018; pp. 367–370.
3. Nouacer, R.; Ortiz, H.E.; Ouhammou, Y.; González, R.C. Framework of Key Enabling Technologies for Safe and Autonomous Drones' Applications. In Proceedings of the 22nd Euromicro Conference on Digital System Design (DSD 2019), Kallithea, Greece, 28–30 August 2019; pp. 420–427.
4. He, S.; Chen, X.; Hung, M.H.; Chen, X.; Ji, Y. Steering Angle Measurement of UAV Navigation Based on Improved Image Processing. *J. Inf. Hiding Multimed. Signal Process.* **2019**, *10*, 384–391.
5. Wallerman, J.; Bohlin, J.; Nilsson, M.B.; Franssen, J.E. Drone-Based Forest Variables Mapping of ICOS Tower Surroundings. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2018), Valencia, Spain, 22–27 July 2018; pp. 9003–9006.
6. Mitcheson, P.D.; Boyle, D.; Kkelis, G.; Yates, D.; Saenz, J.A.; Aldhaher, S.; Yeatman, E. Energy-autonomous sensing systems using drones. In Proceedings of the 2017 IEEE SENSORS, Glasgow, UK, 29 October–1 November 2017; pp. 1–3.

7. D’Odorico, P.; Besik, A.; Wong, C.Y.; Isabel, N.; Ensminger, I. High-throughput drone-based remote sensing reliably tracks phenology in thousands of conifer seedlings. *New Phytol.* **2020**, *226*, 1667–1681. [[CrossRef](#)] [[PubMed](#)]
8. Guo, Y.; Jia, X.; Paull, D.; Zhang, J.; Farooq, A.; Chen, X.; Islam, M.N. A drone-based sensing system to support satellite image analysis for rice farm mapping. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2019), Yokohama, Japan, 28 July–2 August 2019; pp. 9376–9379.
9. Reinecke, M.; Prinsloo, T. The influence of drone monitoring on crop health and harvest size. In Proceedings of the 1st International Conference on Next Generation Computing Applications (NextComp 2017), Moka, Mauritius, 19–21 July 2017; pp. 5–10.
10. Nithyavathy, N.; Kumar, S.A.; Rahul, D.; Kumar, B.S.; Shanthini, E.; Naveen, C. *Detection of Fire Prone Environment Using Thermal Sensing Drone*; IOP Conference Series: Materials Science and Engineering; IOP Publishing: Bristol, UK, 2021; Volume 1055, p. 012006.
11. Hill, A.C.; Laugier, E.J.; Casana, J. Archaeological remote sensing using multi-temporal, drone-acquired thermal and Near Infrared (NIR) Imagery: A case study at the Enfield Shaker Village, New Hampshire. *Remote Sens.* **2020**, *12*, 690. [[CrossRef](#)]
12. Holness, C.; Matthews, T.; Satchell, K.; Swindell, E.C. Remote sensing archeological sites through unmanned aerial vehicle (UAV) imaging. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2016), Beijing, China, 11–15 July 2016; pp. 6695–6698.
13. Inoue, Y.; Yokoyama, M. Drone-Based Optical, Thermal, and 3D Sensing for Diagnostic Information in Smart Farming—Systems and Algorithms. In Proceedings of the 2019 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2019), Yokohama, Japan, 28 July–2 August 2019; pp. 7266–7269.
14. Yi, W.; Sutrisna, M. Drone scheduling for construction site surveillance. *Comput. Aided Civ. Infrastruct. Eng.* **2021**, *36*, 3–13. [[CrossRef](#)]
15. Alzenad, M.; El-Keyi, A.; Lagum, F.; Yanikomeroğlu, H. 3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage. *IEEE Wirel. Commun. Lett.* **2017**, *6*, 434–437. [[CrossRef](#)]
16. Wang, L.; Hu, B.; Chen, S. Energy efficient placement of a drone base station for minimum required transmit power. *IEEE Wirel. Commun. Lett.* **2018**, *9*, 2010–2014. [[CrossRef](#)]
17. Gao, F.; Zhou, Y.; Ma, X.; Yang, T.; Cheng, N.; Lu, N. Coverage-maximization and Energy-efficient Drone Small Cell Deployment in Aerial-Ground Collaborative Vehicular Networks. In Proceedings of the IEEE 4th International Conference on Computer and Communication Systems (ICCCS 2019), Singapore, 23–25 February 2019; pp. 559–564.
18. Ghorbel, M.B.; Rodríguez-Duarte, D.; Ghazzai, H.; Hossain, M.J.; Menouar, H. Joint position and travel path optimization for energy efficient wireless data gathering using unmanned aerial vehicles. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2165–2175. [[CrossRef](#)]
19. Kouroshezhad, S.; Peiravi, A.; Haghghi, M.S. Energy-Efficient Drone Trajectory Planning for the Localization of 6G-enabled IoT Devices. *IEEE Internet Things J.* **2020**, *8*, 5202–5210. [[CrossRef](#)]
20. Huynh, T.; Hwang, W.J. Network lifetime maximization in wireless sensor networks with a path-constrained mobile sink. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 679093. [[CrossRef](#)]
21. Engmann, F.; Katsriku, F.A.; Abdulai, J.D.; Adu-Manu, K.S.; Banaseka, F.K. Prolonging the lifetime of wireless sensor networks: A review of current techniques. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 8035065. [[CrossRef](#)]
22. Verbeke, J.; Hulens, D.; Ramon, H.; Goedeme, T.; De Schutter, J. The design and construction of a high endurance hexacopter suited for narrow corridors. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS 2014), Orlando, FL, USA, 27–30 May 2014; pp. 543–551.
23. Yuan, H.; Xie, J.; Hu, N. Lifetime-optimal sensor deployment for wireless sensor networks with mobile sink. In Proceedings of the 2011 6th IEEE Joint International Information Technology and Artificial Intelligence Conference, Chongqing, China, 20–22 August 2011; Volume 1, pp. 451–454.
24. Sun, Y.; Liu, Y.; Zhou, N.; Lueth, T.C. A MATLAB-Based Framework for Designing 3D Topology Optimized Soft Robotic Grippers. *TechRxiv* **2020**. [[CrossRef](#)]
25. Remelli, E.; Lukoianov, A.; Richter, S.R.; Guillard, B.; Bagautdinov, T.; Bague, P.; Fua, P. MeshSDF: Differentiable Iso-Surface Extraction. *arXiv* **2020**, arXiv:2006.03997.
26. Echegaray, S.; Bakr, S.; Rubin, D.L.; Napel, S. Quantitative Image Feature Engine (QIFE): An open-source, modular engine for 3D quantitative feature extraction from volumetric medical images. *J. Digit. Imaging* **2018**, *31*, 403–414. [[CrossRef](#)] [[PubMed](#)]
27. Maddams, W. The scope and limitations of curve fitting. *Appl. Spectrosc.* **1980**, *34*, 245–267. [[CrossRef](#)]
28. Qing-Wan, H. Curve fitting by curve fitting toolbox of Matlab. *Comput. Knowl. Technol.* **2010**, *21*, 68.
29. Fan, D.; Shi, P. Improvement of Dijkstra’s algorithm and its application in route planning. In Proceedings of the 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery, Yantai, China, 10–12 August 2010; Volume 4, pp. 1901–1904.