

Article

Collision Detection for UAVs Based on GeoSOT-3D Grids

Weixin Zhai ¹, Xiaochong Tong ² , Shuangxi Miao ^{3,*} , Chengqi Cheng ³ and Fuhu Ren ⁴

¹ Academy for Advanced Interdisciplinary Studies, Peking University, Beijing 100871, China

² Zhengzhou Institute of Surveying and Mapping, Zhengzhou 450001, China

³ College of Engineering, Peking University, Beijing 100871, China

⁴ Beijing Institute of Big Data Research, Beijing 100871, China

* Correspondence: miaosx@pku.edu.cn; Tel.: +86-184-8212-7707

Received: 7 May 2019; Accepted: 13 July 2019; Published: 15 July 2019



Abstract: The increasing number of unmanned aerial vehicles (UAVs) has led to challenges related to solving the collision problem to ensure air traffic safety. The traditional approaches employed for collision detection suffer from two main drawbacks: first, the computational burden of a pairwise calculation increases exponentially with an increasing number of spatial entities; second, existing grid-based approaches are unsuitable for complicated scenarios with a large number of objects moving at high speeds. In the proposed model, we first identified UAVs and other spatial objects with GeoSOT-3D grids. Second, the nonrelational spatial database was initialized with a multitable strategy, and spatiotemporal data were inserted with the GeoSOT-3D grid codes as the primary key. Third, the collision detection procedure was transformed from a pairwise calculation to a multilevel query. Four simulation experiments were conducted to verify the feasibility and efficiency of the proposed collision detection model for UAVs in different environments. The results also indicated that 64 m GeoSOT-3D grids are the most suitable basic grid size, and the reduction in the time consumption compared with traditional methods reached approximately 50–80% in different scenarios.

Keywords: collision detection; unmanned aircraft vehicles; GeoSOT-3D

1. Introduction

Over the last decade, unmanned aerial vehicles (UAVs) have been utilized in a wide range of military and civilian applications, including surveillance [1,2], monitoring [3,4], imaging [5,6], and reconnaissance [7]. UAVs boast many advantages, including remarkable flexibility, low energy consumption, high efficiency and the capacity for real-time monitoring. The increasing number of UAVs, however, has led to challenges related to solving the collision problem to ensure air traffic safety.

Traditionally, two categories of approaches have been employed for collision detection. The first category includes simple collision detection methods based mainly on sensing technologies, such as automatic dependent surveillance-broadcast (ADS-B) technology, visual sensors, optical flow, radar, and light detection and ranging (LiDAR), to alter the course of a flying vehicle toward another direction in the short term to avoid collisions. For example, Scherer et al. and Achtelik et al. employed laser scanners on UAVs to detect obstacles [8,9]. Bryson et al. presented the ongoing design and implementation of a robust inertial sensor-based simultaneous localization and mapping (SLAM) algorithm with bearing-only observations [10]. Gronzka used a multilevel SLAM algorithm with estimates of aircraft motion and altitude for three-dimensional (3D) mapping, positioning, and navigation [11]. Sabatini et al.'s research provided an obstacle detection solution especially suitable for low-altitude UAVs with the remarkable angular resolution and accuracy of LiDAR [12]. Becker and Bouabdallah [13] employed four ultrasound sensors for collision detection and a camera system based

on above-ground optical flow computations for positioning. However, the abovementioned methods can be ineffective under poor lighting conditions, such as smoke and fog, and they are unable to detect certain objects composed of transparent or signal-absorbing materials. Moreover, it is challenging for these methods to develop a flight plan ahead of time.

The second category of collision detection approaches is based on global or relative positioning, i.e., the coordinates of UAVs and obstacles within the airspace. These positions are acquired either by global continuous positioning systems such as the Global Positioning System (GPS) and Beidou or by obtaining exact indoor location information with Wi-Fi [14] or beacons (Bluetooth Low Energy, BLE) [15]. Fiorini and Shiller presented a computationally efficient heuristic approach applicable to the online planning of UAVs [16]. Nordlund et al. proposed a probabilistic method designed for noncooperative collision avoidance [17]. Several optimization-based schemes focus on solving optimization problems either deterministically or in a probabilistic setting, often in a centralized way, as exemplified in [18,19]. Martin and Alan integrated a triaxial magnetometer aiding sensor into an extended Kalman filter (EKF)-based navigation and collision avoidance algorithm [20]. However, coordinate-based conflict detection suffers from high computational complexity, especially in the context of a highly dynamic environment with a large data volume. In addition, resolving one conflict may introduce new conflicts into the trajectory, further increasing the computational requirements of conflict detection algorithms [21].

Multidimensional grids provide another viewpoint for UAV conflict detection. Grid-based conflict detection algorithms have been proven to be more computationally efficient than existing algorithms because they eliminate the need for pairwise computations of interaircraft distances. In 2003, Jardin applied a three-dimensional grid (two horizontal spatial dimensions and one time dimension) to real-time conflict-free trajectory optimization [22], and in 2005, Jardin extended the theory to four dimensions [23,24]. The grid concept was subsequently implemented as a data structure to enable conflict data integration and solution space visualization [25]. Kushleyev et al. conducted an experiment with 16 microquadrotors operating autonomously with an external localization system possessing two-dimensional (2D) grids [26]. Spatial grids were further employed for indoor UAV collective motion control experiments [27–29].

Previous studies on UAV collision detection suffered from two main drawbacks. First, in most traditional studies, pairwise calculations must be performed on the spatial coordinates acquired from either airborne or satellite sensors. However, directly calculating the values of the equations of space curves using analytical expressions will constitute an enormous computational burden. Potential conflicts with air traffic need to be considered, and it is also important to consider other factors, such as terrain and weather, when performing collision detection. Second, the existing grid-based approaches were mostly restricted within a confined airspace, and the scenarios of their applications were relatively simple. When dealing with collision detection issues under complex and high-speed conditions with multiple spatial entities, the aforementioned grid-based approaches lacked a comprehensive and unified computation system. Moreover, the improved computational efficiency of grid-based methods is accompanied by a certain increase in the computer memory requirement. Therefore, with the sharp increase in the number of aircrafts and the expansion of the scope of their applications, it is highly important to develop an efficient UAV collision detection approach capable of adapting to complex scenes.

A number of recent studies on global grid systems have provided opportunities for geospatial data integration, retrieval, and reference [30,31]. Discrete global grid systems (DGGs) are being increasingly utilized for the creation of such grid systems. DGGs are a discretization of the Earth into hierarchical sets of highly regular grids, each of which represents a distinct region of the Earth to which data may be assigned [30,32]. In this research, we referred to geographical coordinate subdividing grid with one-dimension integer coding on a 2^n tree in three dimensions (GeoSOT-3D) theory to construct the basic spatial reference framework. GeoSOT-3D has been widely studied in various fields, including spatial data reintegration [32,33], database indexing [34,35], risk analysis for natural

gas pipelines [36], building information modeling (BIM) data visualization [37], remote sensing data management [38], and city component identification [39]. The applications of GeoSOT-3D theory have demonstrated its importance and suitability for highly efficient and accurate spatial data processing and management endeavors.

Referring to the idea of DGGs in this paper, we adopted the detection of UAV collisions as the optimal problem and adapted a GeoSOT-3D grid-based system to coordinate multi-UAV flights. In our approach, we first established a unified spatial grid subdivision system for the studied airspace and mapped the spatial and temporal information of the UAVs and other spatial entities onto different grids. Second, we applied GeoSOT-3D codes to multilevel spatiotemporal indexing to store spatial information in a spatial database. Finally, the multilevel massive data query engine was shown to support the implementation of multiple processes with high concurrency and massive amounts of data. The proposed method not only provides a powerful means for real-time UAV collision detection but also maximizes airspace utilization through effective management and control while ensuring security.

2. Materials and Methods

2.1. GeoSOT-3D Grid System

The GeoSOT-3D grid system proposed for this study is a universal and hierarchical three-dimensional geospatial reference system [40]. By subdividing the Earth through three iterations (initially expanding the Earth ($180^\circ \times 360^\circ$) into $512^\circ \times 512^\circ$ grids, then expanding each 1° into $64'$, and finally expanding each $1'$ into $64''$), two-dimensional quadtree subdivisions at the degree, minute, and second levels are obtained. GeoSOT-3D is congruent and aligned: the largest subdivision grid in the highest level (Level 0) can represent the entire Earth surface, while the smallest subdivision grid in the lowest level (Level 32) can represent the centimeter scale.

Based on our previous work, we extended the two-dimensional global subdivision framework into a third spatial dimension: the elevation dimension. The division of the elevation dimension is directly adopted through a dichotomy. The entire range over which the elevation dimension is divided spans from the geocentric position to 50,000 km above sea level. In this way, this GeoSOT-3D grid system not only possesses global multidimensional octree hierarchical characteristics but also maintains consistency with different types of applications. With an octree-like structure, the GeoSOT-3D coding exhibits three-dimensional components. The node index that branches out from the top level is determined by a formula that uses bits to characterize the longitude, latitude, and altitude within the former node's space. The digit is either 0 or 1 at each level in one dimension; therefore, a digit ranging in value from 0 to 7 can be applied to represent one code at each level by occupying 3 bits. The next (lower) level inherits the digit from the former digit, and an identical formulation is repeated. In this research, if grid_A is spatially contained by grid_B , we claimed grid_A as the parent grid and grid_B as the child grid. The Level-2 GeoSOT-3D grid system is illustrated in Figure 1. To better explain the grid code structure, we did not present the actual three-dimensional forms. There are 64 three-dimensional grids ($2^3 \times 2^3 = 64$), and the corresponding codes are marked on the regular cubic faces, which are also shown on the four cross-section planes. The corresponding binary codes were listed for the front side, the right side, and the four layers for reference. The Z-order coding scheme maps multidimensional data into one dimension while preserving the locality of the original data. In each code, every bit belongs to a certain dimension and has dichotomous implications. To elaborate, we highlighted the binary bit with the same values in one group to demonstrate that the data of a certain side or layer are determined only by specific binary bits.

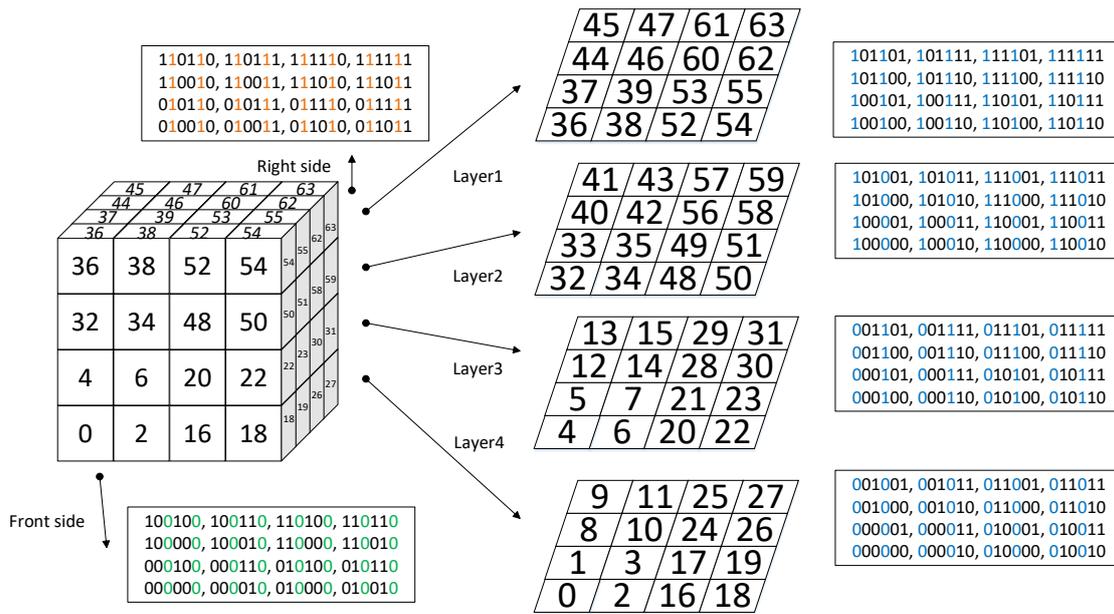


Figure 1. Detailed illustration of GeoSOT-3D grids on Level 2. The codes for the three-dimensional grids are marked on the surface of the cube and on four separate layers. The binary codes are also listed for the front side, right side, and each layer.

2.2. Multilevel Spatiotemporal Indexing

The studied airspace comprises different categories of information (e.g., other aircraft and weather conditions). Each spatial entity is represented by one or more GeoSOT-3D grids according to the occupied grids. The measurement errors of the positions of moving targets such as UAVs exist due to a large number of spatial perturbations (including the atmosphere, illumination, and gravity) and the control ability. When there is only absolute position information at the sampling time without considering measurement error, missing or mistaken selections may arise in the process of mapping the moving objects and computing their mutual relations. Therefore, the modeling of spatial objects should consider not only the spatial location of the object but also the small-scale space surrounding the object. Therefore, the basic partitioning unit under the three-dimensional framework is the subdivision grid; that is, all three-dimensional spatial objects are composed of one or more subdivision grids, as indicated in Figure 2.

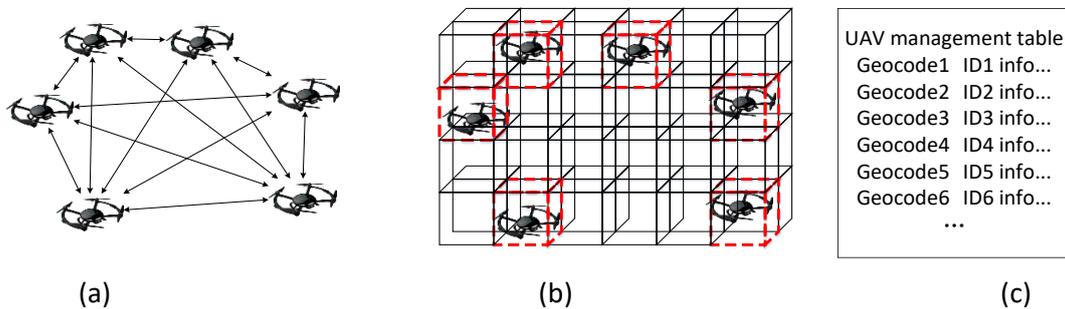


Figure 2. Traditional calculation method and spatial database-based query method for unmanned aerial vehicle (UAV) collision detection. Traditional collision detection methods depend on pairwise calculations based on three-dimensional coordinates (a). The calculation complexity increases rapidly as the number of UAVs increases. The grid-based method simplifies the collision detection procedure as a comparison process for regular-sized spatial grids (b). The grid-based method can be further advanced with a UAV management table (c). In the table, the GeoSOT-3D code is set as the primary key, and the collision can be detected by a straightforward query.

The spatiotemporal information of the object is recorded by spatiotemporal coding corresponding to a certain spatial region and time range. The primary keys of data tables are initialized according to the experimental airspace. Sorted spatial grid codes are associated with spatial grids with unique characteristics; therefore, they are adopted as the primary key of the spatial database. We reduce the sorting time by using the prebuilt fixed table structure. In the simplest case, when there is an object in one spatial grid, the record of the corresponding code is 1; otherwise, it is 0. Temporal information is also recorded in each row of the database with a start time stamp and an end time stamp, indicating which objects have been in the space within that time. Thus, we simplify complex pairwise calculation problems into a spatial database query problem. In our experiment, we chose nonrelational databases to store spatial data with multiple tables. Other than the temporal and spatial information, each row of data records various types of information, such as categories, data sources, and physical attributes of the original data. The overall table structure is stable, and the row size of the table does not change, which is suitable for the complex scenes of dynamic objects in space.

One of the great advantages of the GeoSOT-3D grid-based collision detection technique is the ability to filter out redundant calculations with multilevel queries. To meet the needs of a multilevel query, we designed the multitable strategy shown in Figure 3. More than one data table is available to manage and record the UAVs' information in a specific airspace. The primary key of the data table is a spatial code at a certain level that uniquely corresponds to a certain space in the actual three-dimensional grid system. Each table stores only the data corresponding to the encoding of a certain level. Each parent code retains a pointer to the data table in which the next level of the child code is located. In this way, after the parent code is queried, if a further detailed query is needed, it can directly link to the data table where the child code is located. At the same time, the sizes of the tables can be kept constant, which is suitable for distributed management; this approach maintains load balancing and efficiency when dealing with the requirements of immense data volumes and high concurrency.

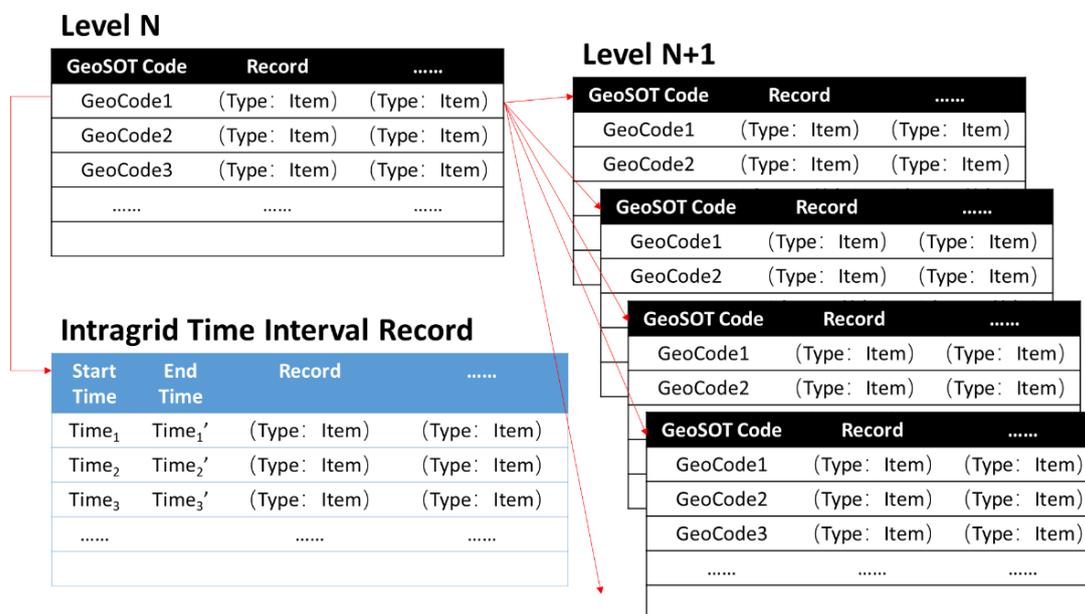


Figure 3. Multitable strategy. There are a number of data tables in the nonrelational database. Each data table records spatiotemporal data at a specific subdivision level. Records at higher levels preserve pointers to the data table at lower levels. The GeoSOT-3D grid code is set as the primary key for each table. Temporal information is also recorded corresponding to each GeoSOT-3D code.

Each spatial object, such as a UAV, corresponds to different spatial grid codes on different spatial scales. Generally, one data table stores only information on a certain scale. Tables with different levels may record one entity's spatiotemporal information. This strategy reduces the burden on each data table because the spatiotemporal information is assigned to multiple tables; in addition, this strategy is suitable for high-concurrency multiprocess queries at the backend.

In this framework, a conflict may occur when the planned route of a UAV passes through a spatial grid that is occupied by another object (either a UAV or another obstacle). The GeoSOT-3D grids were coded in an octree structure, preserving the locality of the original spatial data. Thus, two spatially close grids have partially matching codes from the beginning. In contrast, two distant grids would have codes with different digits from the beginning. Objects that are unlikely to collide with each other are directly screened out because their corresponding spatial codes are not close, and thus, no further operations are required.

2.3. Multilevel Query Methods

In this framework, we conducted collision detection by using a multilevel query method. In previous studies, a conflict was detected when two entities (e.g., aircrafts) were in the same cell. We improved the previous coding systems into a hierarchical system, in which the query operation is not limited to one level. In the collision detection process, each code has the child code or father code, and codes at different levels are calculated. The new data, such as UAV trajectory data, and obstacle data should also be inserted after being transformed into GeoSOT-3D codes. In this way, the computational redundancy is reduced significantly. After a collision is detected, the new codes representing UAV trajectories or obstacles are inserted as new columns instead of new rows.

Collision detection for a single grid is shown in Figure 4. We first assumed a grid is present with a code of $Code_A$ at level N over a time interval of $[T_A, T_A']$. The codes are stored in data tables relating to higher or equal levels from 0 to N . Then, the data level variable i starts to loop from 0 to $N-1$. At each level i , we need to obtain the parent code $Fa(Code_A)_i$ of $Code_A$ and check the database to determine whether the corresponding row is null. If there are no data and no neighboring data, we can break the loop for the multilevel insert. If a data record already exists but is not included or adjacent to the time interval of $[T_A, T_A']$, the loop is also skipped, and the next step is composed of the multilevel insert. If the loop reaches level N , the method described above can be applied to determine the grid's relationship: intersecting, neighboring, or disjoint.

If the grid contains an intersecting relationship, two objects will share the same space at the same time, which would probably lead to a collision. If the grid contains an adjacent relationship, there is a risk of collision to a certain degree. Finally, if the grid contains a disjoint relation, there is no risk of collision. The parallel insert operation consists of inserting the parent code of $Code_A$ at levels from 0 to $N-1$ within the time interval $[T_A, T_A']$.

The spatiotemporal collision detection process for multiple grids is shown in Figure 5. We assumed that several spatiotemporal grids are present; each grid corresponds to a spatial grid code $Code_A$ and time interval $[T_A, T_A']$, and the lowest-level space code is set as N . The collection of the spatiotemporal grid is recorded as $\{Code_A, [T_A, T_A']\}$. For each individual grid, we obtain the parent grid codes on levels from 0 to $N-1$, and the time records are also retained. Thus, several spatial grid codes can be stored at one level, and one spatial grid code may have several corresponding time interval records. The level variable i loops from level 0 to level $N-1$. For all spatial grid codes at level i with the corresponding time interval $[T_A, T_A']$, if it is null, the condition for the data insert operation is satisfied; otherwise, the collision conditions should be examined individually according to the previous formula, and the collision-free grids with their child grid codes should be removed. When the variable i reaches N , we should make one-to-one judgments with each existing grid code. If any of them intersect with $Code_A$, the result is a collision. If it is an adjacent relationship, there is a risk of collision. If it is a disjoint relationship, there is no risk. When the last subprocesses all satisfy the insert conditions, the multilevel data can be inserted into the multilevel data tables in parallel.

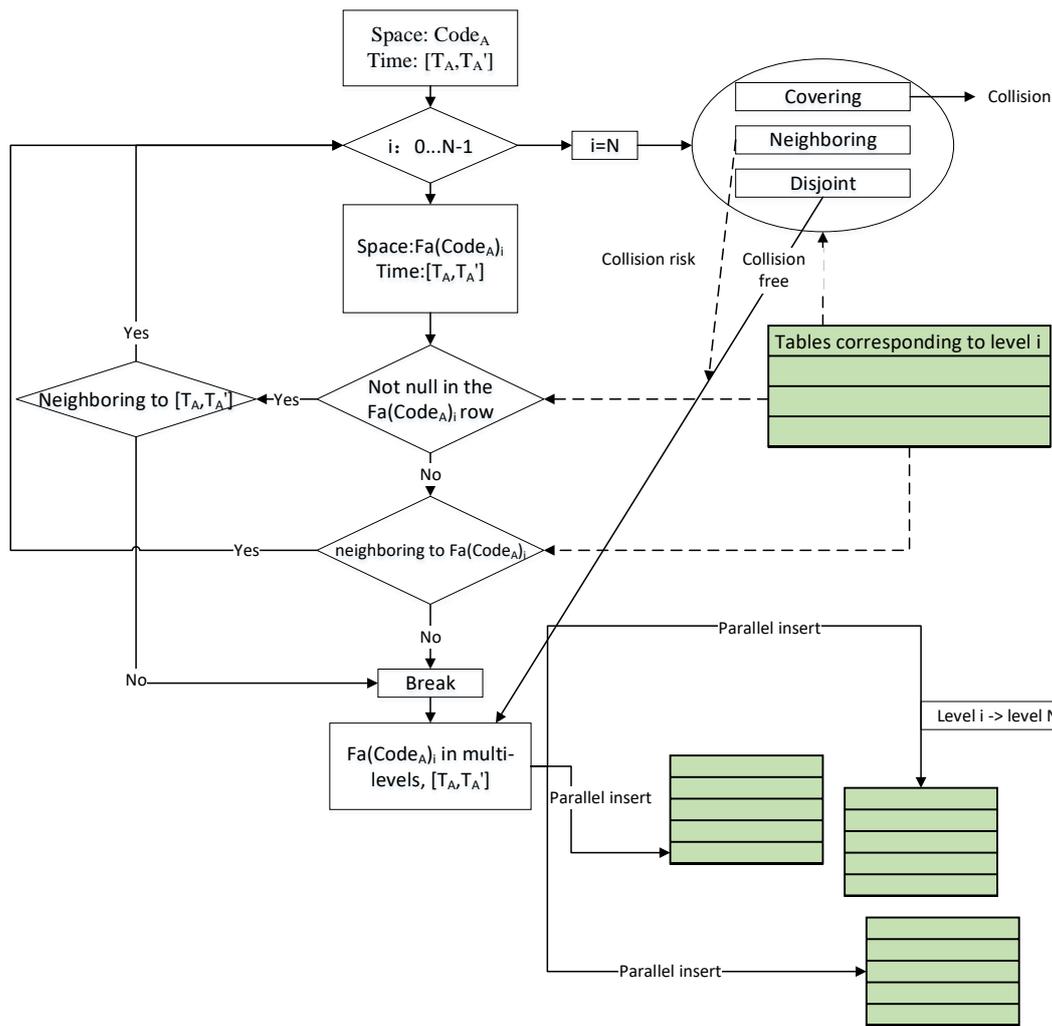


Figure 4. Collision detection for a single grid. A single spatiotemporal grid with Code_A and a time interval from T_A to T_A' is compared with the existing data at multiple levels. After determining the spatial relationship, the spatiotemporal grid information can be inserted into multiple data tables in parallel.

The spatiotemporal collision avoidance technique utilizing spatial subdivision grids has the following advantages:

- 1) The octree structure of the space reduces the number of comparisons, thereby filtering out many codes that need no further consideration and greatly improving the comparison efficiency;
- 2) Multiple operations are required to obtain the parent code at all levels in the operation flow, but the operations to obtain the parent code are simple and need only to be shifted;
- 3) Multilevel storage enables the parallel insertion of different large tables without the insertion of once table affecting that of another, which improves the insertion efficiency;
- 4) Temporal and spatial data are stored in a table with spatial codes, avoiding the requirement for multiple indexes and simplifying the detection procedure.

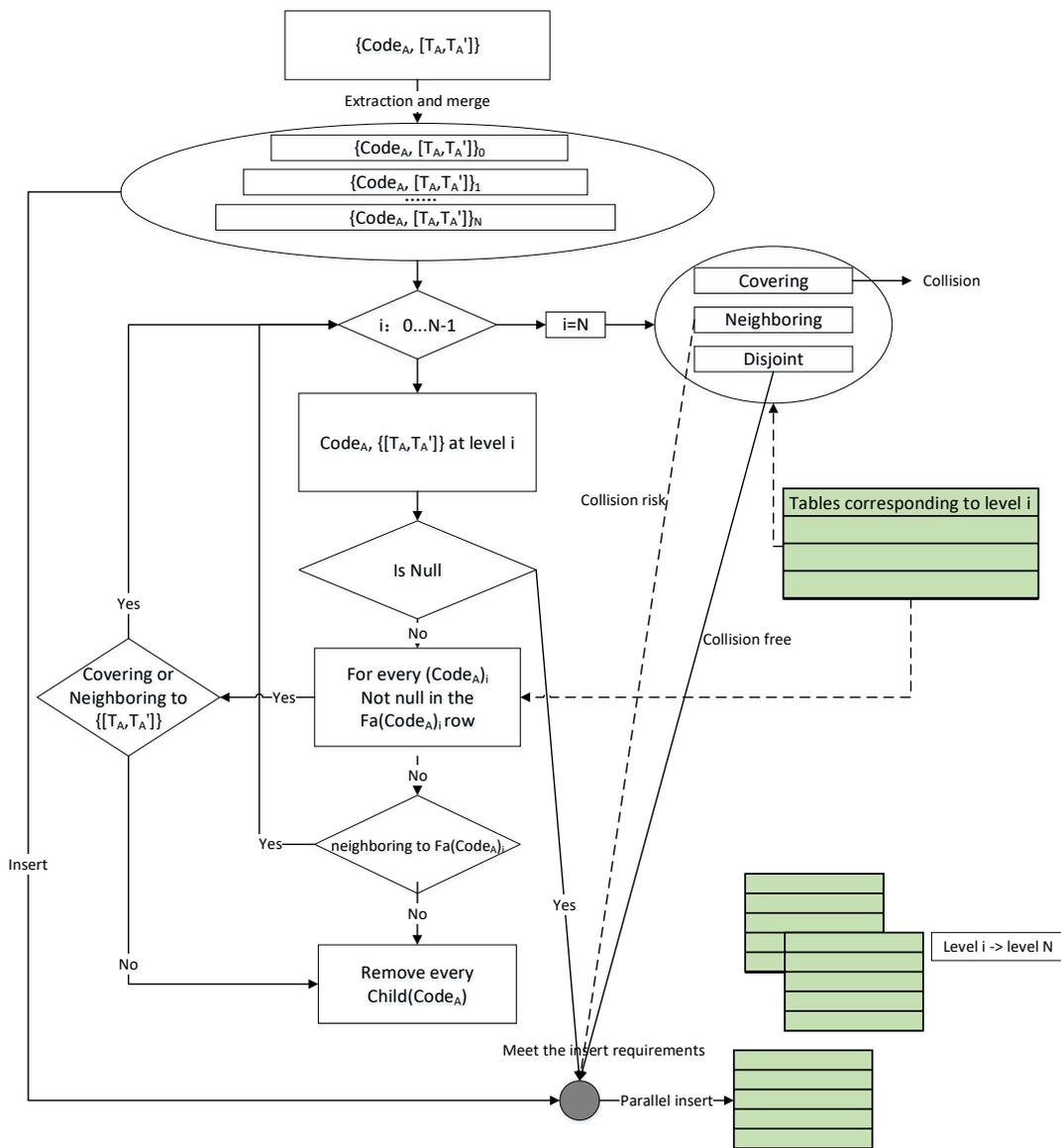


Figure 5. Collision detection for multiple grids. Multiple spatiotemporal grids are compared individually with the existing data at multiple levels. After determining the spatial relationship, the spatiotemporal grid information can be inserted into multiple data tables in parallel.

3. Experiments

Based on the proposed model, four flight simulation experiments were carried out using the simulation data to prove the feasibility and efficiency of the proposed collision detection model for UAVs in different environments. The first experiment aimed to identify the most appropriate level for UAV collision detection. The second experiment was conducted to evaluate the computational efficiency of the collision detection procedure among three different methods in an airspace without any other obstacles. The third experiment was the same as the second experiment, but other obstacles were also present within the airspace. The last experiment was a UAV formation flight simulation experiment in a dynamic scenario with formation optimization. The experiments were conducted on a personal computer with an Intel(R) Core(TM) i5-4200M CPU @ 2.50 GHz 2.49 GHz and a 16.0 GB memory space.

3.1. The Appropriate Level of Identification for UAV Collision Detection

We aimed to find a suitable grid size for managing the formation. If the selected grid is too small, each UAV may require more than one grid for representation; if the grid is too large, false alarms will be induced. The collision detection calculation between two UAVs involves multiple grid comparison algorithms, which will be time consuming. If the selected grid is too large, the two UAVs may still be considered to collide; otherwise, the neighbor may need to be judged to be further away when the distance is great. This kind of grid might also lead to a low efficiency. An optimal grid size can balance these two costs. In the simulation experiment, the basic temporal resolution was 1 s, and the minimum bounding box size of the UAV was 20 m × 20 m × 20 m. The average flight speed of the UAV was 300 km/h (83 m/s) with a variance of 30 km/s; thus, the spatial displacement was approximately 80 m/s. For classification according to size, we can refer to the U.S. Army Roadmap for UAS 2010–2035. The UAV in our simulation experiment can be categorized as a “large” UAV [41]. The positions, flying directions, and speeds of the UAVs can be acquired at each time, and the flying directions and speeds vary randomly within a small range. Once a collision was detected, the aircraft turned to another direction and continued to fly. The airspace was a cube that is 5 km × 5 km in area and 15–20 km in height. The UAVs fly in an unconstrained way throughout the airspace and are detected for collisions in real time. This section compares the simulation results under different conditions to determine the size of the GeoSOT-3D grid code that is the most suitable for collision detection. The experimental results are shown in Figure 6. According to the size of the UAVs, the suitable grid sizes were restricted to 128 m, 64 m, and 32 m, which correspond to levels 20, 21, and 22, respectively. In Figure 6, the horizontal axis represents the total number of UAVs, and the vertical axis represents the total time taken to detect collisions for all drones. We tested the time required for detection for numbers of UAVs ranging from 100 to 1000. The time consumed for the 32 m GeoSOT-3D grids and 128 m GeoSOT-3D grids were similar, and both were greater than that for the 64 m GeoSOT-3D grids, indicating that the 64 m grid code results in the best performance.

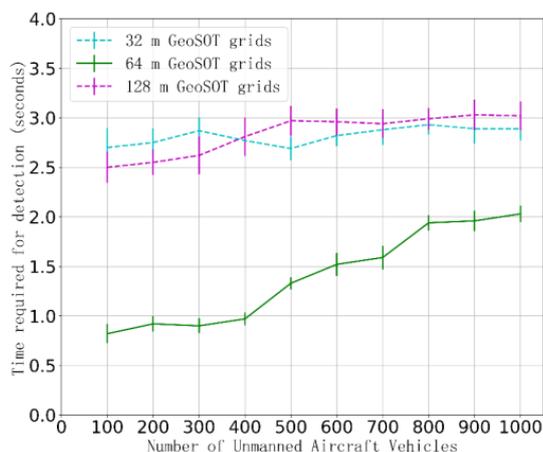


Figure 6. Time required for detection with different GeoSOT-3D grid sizes. The three selected GeoSOT-3D grid sizes are 128 m, 64 m, and 32 m, which correspond to levels 19, 20, and 21, respectively.

3.2. Evaluation of the Computational Efficiencies of Three Methods in an Obstacle-Free Airspace

After we selected the 64 m GeoSOT-3D grid as the basic grid size, we designed further experiments to evaluate the time consumed by the collision detection with three-dimensional coordinates, 64 m simple grids and 64 m GeoSOT-3D grids. Collision detection with three-dimensional coordinates is based on the pairwise calculation approach, in which the collision is detected by computing the Euclidean distance between every two UAVs and comparing the calculated distance with the threshold (UAV size). Collision detection with simple 64 m grids is based on traversing the grids and identifying whether there is more than one UAV in a GeoSOT-3D grid. Collision detection with 64 m GeoSOT-3D

grids, namely, our proposed method, employs 64 m grids as the basic grids. Figure 7 displays the results of our simulated UAV formation flight. The experimental conditions were the same as those in the previous experiment, and the number of UAVs ranged from 100 to 1000. We find that, when the quantity of UAVs was less than 300, the three methods displayed similar performance; however, as the quantity of UAVs increased, the calculation times of the two grid methods were better than that of the method based on three-dimensional coordinates. Moreover, the method based on 64 m GeoSOT-3D grids was superior to that based on simple 64 m grids. For large numbers of UAVs, the proposed method can reduce the time consumption by up to approximately 80%.

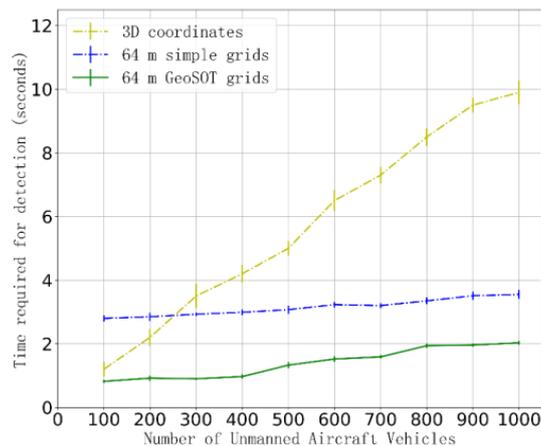


Figure 7. Time required for detecting collisions with three methods. The three methods consist of calculations with three-dimensional coordinates, calculations with simply 64 m grids, and spatial database queries with 64 m GeoSOT-3D grids.

3.3. Evaluation of the Computational Efficiencies of Three Methods in an Airspace with Obstacles

Conflicts may occur, not only between UAVs but also between a UAV and another aircraft or adverse weather conditions. Hence, it is fairly straightforward to extend the proposed system to include available information on other types of conflicts. When a UAV passes through a GeoSOT-3D grid that is occupied by another spatial object, the situation is processed in the same way as a conflict between UAVs.

Figure 8 displays the results of another simulated UAV formation flight. Different from the previous experiments, a $1 \text{ km} \times 1 \text{ km} \times 1 \text{ km}$ obstacle was placed at the center of the airspace, and the experimental results are shown in subfigure (a). In addition, 100 obstacles of the same size as the UAVs were placed within the airspace, and the experimental results are shown in subfigure (b). We find that, when the quantity of UAVs was less than 300, the three methods exhibited little difference; however, as the quantity of UAVs increased, the computation times of the two grid methods were significantly better than that of the method based on three-dimensional coordinates. Moreover, the method based on 64 m grids was superior to that based on basic 64 m grids. The general trends were similar to those observed in the obstacle-free experiment, and the time consumed for collision detection was reduced by approximately 70–80% with the proposed method.

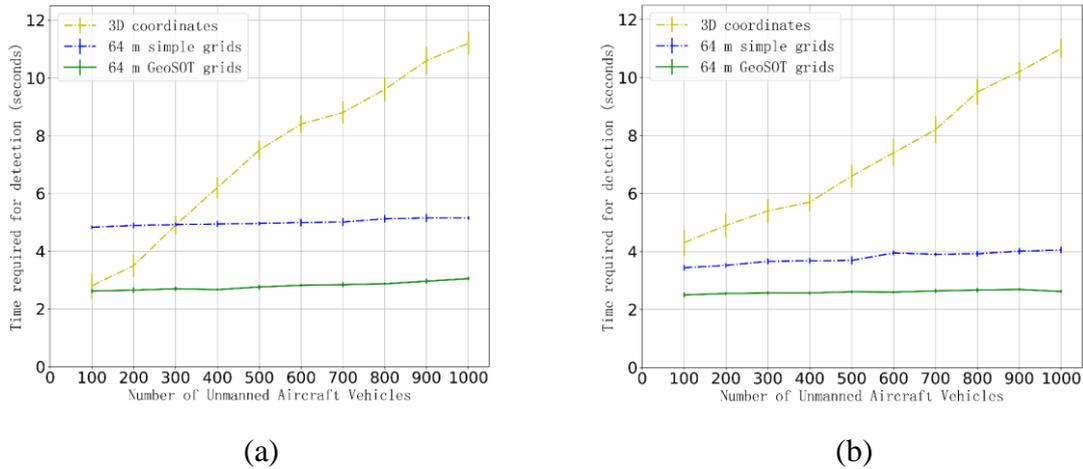


Figure 8. Time required to detect collisions with three methods in an airspace containing obstacles. In simulation experiment (a), a 1 km × 1 km × 1 km obstacle was placed at the center of the airspace. In simulation experiment (b), 100 obstacles of the same size as the UAVs were placed within the airspace.

3.4. Detection with Formation Optimization

Next, we optimized the calculations for the UAV formation with fixed relative positions. When the UAVs perform cooperative tasks by flying as a group, they can be regarded as flying in a formation. A UAV formation can be rigidly defined by relative position vectors. Because the speed of each aircraft is limited, it is impossible to achieve a large displacement in a short time. If the satellite positioning refresh frequency can be maintained, we need only to detect the collision for each previously occupied grid in a restricted space, thereby reducing the overall computation time. In this experiment, the search was limited to 2 levels. Different from the previous simulation experiments, the UAVs in this simulation experiment were constrained by a formation. Figure 9 indicates that formation optimization can reduce the time consumption by up to 25%.

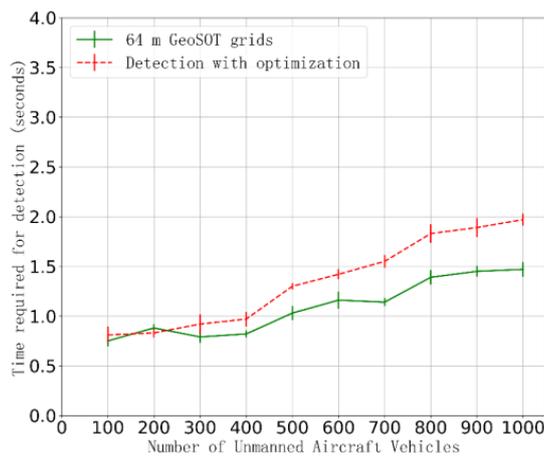


Figure 9. Time required to detect a collision after formation optimization. Collision detection with optimization is shown with a green line. The optimization is based on the premise that the UAVs are established in a fixed formation.

4. Discussion

It is feasible to employ the collision detection method based on GeoSOT-3D grids proposed in this study for better UAV management and monitoring, as the novel technique also benefits flight path planning, airspace control, and formation flight. The reduction in the time consumption compared with the traditional methods is approximately 50–80%. Furthermore, as the amount of data in the airspace

increases, the approach proposed in this study becomes more advantageous. These improvements may arise for the following reasons:

1) Traditional methods require one-to-one calculations between different entities to determine their spatial relationship, but the proposed method with grid coding needs only to query within the database, and thus, the computation times are reduced.

2) Hierarchical coding schemes enable efficient filtering to remove a large number of redundant or unnecessary calculations.

3) Using nonrelational database storage along with a multitable strategy improves the multilevel operation efficiency and allows parallel computing.

As the size of the grid increases, the computations required for detection within each grid also increase. The position, flying direction, and speed of UAVs at each time can be acquired, and the flying direction and speed vary randomly within a small range. In addition, the UAVs can be categorized according to size and maximum speed; therefore, the optimal grid sizes are different. In general, to ensure a safe distance during flight and high computational efficiency, if the flying speed is low, a smaller grid size would be preferred; if the UAV is small, a smaller grid size would also be preferred. The simulations in this study are provided to demonstrate the grid size selection for one type of UAV. The calculation and comparison process can be applied to UAVs of other sizes and flying speeds. In this research, we did not carry out real tests, only simulation experiments. In future research, we intend to perform experiments on real UAVs to further verify the utility of the GeoSOT-3D grid-based model.

5. Conclusions

Collision detection in a highly dynamic and complex environment is key for a variety of UAV applications. Two traditional categories of approaches for collision detection were criticized for their computational complexity. The first category of methods utilizes different sensing technologies as the basis of collision detection, whereas the second category uses global or relative positioning systems to acquire the coordinates of UAVs and obstacles within an airspace. Two disadvantages of previous studies on UAV collision detection were discussed. First, pairwise calculations must be performed based on the spatial coordinates acquired by either airborne or satellite sensors; consequently, the computational burden increases exponentially with an increase in the number of spatial entities. Second, although multidimensional grids provide an alternative for UAV collision detection, the existing grid-based approaches are not suitable for complicated scenarios with large numbers of high-speed moving objects.

This study demonstrated an approach for detecting collisions among UAVs based on GeoSOT-3D grids. The proposed method is based on the GeoSOT-3D spatial subdivision theory, which has proven its applicability in various applications. All UAVs and other entities in a studied airspace should first be transformed into one or more GeoSOT-3D grids. The code for each grid should then be set as the primary key for the nonrelational database. The multitable strategy was adopted in the proposed method by considering the hierarchical advantages of GeoSOT-3D coding, and the collision detection calculation was transformed into a multilevel query. We described two algorithms for a single grid and multiple grids in detail. In the detection process, new data, such as UAV trajectory data and obstacle data, should be inserted after being transformed into GeoSOT-3D codes. In this way, the computational efficiency is significantly improved. After detecting collisions, the new codes are inserted as new columns instead of new rows. The proposed method involves not only potential collisions with traffic but also other categories of entities in the studied airspace.

To verify the suitability and efficiency of the proposed method, we designed and conducted four UAV formation flight simulation experiments. Accordingly, the feasibility of detecting the collisions of UAVs based on GeoSOT-3D grids in three-dimensional space was proven. We compared different grid sizes for collision detection, with the number of UAVs ranging from 100 to 1000, and the computation time of the 64 m GeoSOT-3D grids was the lowest. For different numbers of UAVs, the time required to achieve collision detection with different calculation methods was acquired. The efficiency of the grid

method was much higher than that of the ordinary three-dimensional coordinate calculation method, and the computational efficiency was further improved with the proposed method. Calculations in different simulation scenarios all proved the advancement of the GeoSOT-3D grid-based method. For a UAV formation with fixed relative positions, the query process can be optimized; in our experiment, formation optimization reduced the time consumption by up to 25%.

Author Contributions: Weixin Zhai conceived, designed, and performed the experiments and wrote the manuscript; Xiaochong Tong and Shuangxi Miao supervised the study; Chengqi Cheng and Fuhu Ren offered helpful suggestions and reviewed the manuscript. All authors have read and approved the submitted manuscript, have agreed to be listed, and have accepted this version for publication.

Funding: This research was funded by the National Key Research and Development Projects (2018YFB0505300, 2017YFB0503703), the Guangxi Science and Technology Major Project (AA18118025), and the National Defense Science and Technology Innovation Special Zone Project.

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

References

1. Nigam, N. The multiple unmanned air vehicle persistent surveillance problem: A review. *Machines* **2014**, *2*, 13–72. [CrossRef]
2. Yang, L.; Qi, J.; Song, D.; Xiao, J.; Han, J.; Xia, Y. Survey of robot 3D path planning algorithms. *J. Control Sci. Eng.* **2016**, *2016*, 5. [CrossRef]
3. Wallace, L.; Lucieer, A.; Watson, C.; Turner, D. Development of a UAV-LiDAR system with application to forest inventory. *Remote Sens.* **2012**, *4*, 1519–1543. [CrossRef]
4. Paranjape, A.A.; Meier, K.C.; Shi, X.; Chung, S.-J.; Hutchinson, S. Motion primitives and 3D path planning for fast flight through a forest. *Int. J. Robot. Res.* **2015**, *34*, 357–377. [CrossRef]
5. Zang, Y.; Yang, B.; Li, J.; Guan, H. An Accurate TLS and UAV Image Point Clouds Registration Method for Deformation Detection of Chaotic Hillside Areas. *Remote Sens.* **2019**, *11*, 647. [CrossRef]
6. Torres, M.; Pelta, D.A.; Verdegay, J.L.; Torres, J.C. Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. *Expert Syst. Appl.* **2016**, *55*, 441–451. [CrossRef]
7. Ryan, A.; Zennaro, M.; Howell, A.; Sengupta, R.; Hedrick, J.K. An overview of emerging results in cooperative UAV control. In Proceedings of the 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No. 04CH37601), Nassau, Bahamas, 14–17 December 2004; pp. 602–607.
8. Achtelik, M.; Bachrach, A.; He, R.; Prentice, S.; Roy, N. Autonomous Navigation and Exploration of a Quadrotor Helicopter in GPS-Denied Indoor Environments. Available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.330.2679&rep=rep1&type=pdf> (accessed on 1 May 2019).
9. Scherer, S.; Singh, S.; Chamberlain, L.; Elgersma, M. Flying fast and low among obstacles: Methodology and experiments. *Int. J. Robot. Res.* **2008**, *27*, 549–574. [CrossRef]
10. Bryson, M.; Sukkarieh, S. Building a Robust Implementation of Bearing-only Inertial SLAM for a UAV. *J. Field Robot.* **2007**, *24*, 113–143. [CrossRef]
11. Grzonka, S.; Grisetti, G.; Burgard, W. A fully autonomous indoor quadrotor. *IEEE Trans. Robot.* **2012**, *28*, 90–100. [CrossRef]
12. Sabatini, R.; Gardi, A.; Richardson, M. LIDAR obstacle warning and avoidance system for unmanned aircraft. *Int. J. Mech. Aerosp. Ind. Mechatron. Eng.* **2014**, *8*, 718–729.
13. Becker, M.; Sampaio, R.C.B.; Bouabdallah, S.; Perrot, V.d.; Siegwart, R. In-flight collision avoidance controller based only on OS4 embedded sensors. *J. Braz. Soc. Mech. Sci. Eng.* **2012**, *34*, 294–307. [CrossRef]
14. Yang, C.; Shao, H.-R. WiFi-based indoor positioning. *IEEE Commun. Mag.* **2015**, *53*, 150–157. [CrossRef]
15. Bekkelien, A.; Deriaz, M.; Marchand-Maillet, S. Bluetooth Indoor Positioning. Master's Thesis, University of Geneva, Geneva, Switzerland, 2012.
16. Fiorini, P.; Shiller, Z. Motion planning in dynamic environments using velocity obstacles. *Int. J. Robot. Res.* **1998**, *17*, 760–772. [CrossRef]
17. Nordlund, P.-J.; Gustafsson, F. Probabilistic noncooperative near mid-air collision avoidance. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 1265–1276. [CrossRef]
18. Malaek, S.; Abbasi, A. Near-optimal terrain collision avoidance trajectories using elevation maps. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 2490–2501. [CrossRef]

19. Frazzoli, E.; Mao, Z.-H.; Oh, J.-H.; Feron, E. Resolution of conflicts involving many aircraft via semidefinite programming. *J. Guid. Control. Dyn.* **2001**, *24*, 79–86. [[CrossRef](#)]
20. Barczyk, M.; Lynch, A.F. Integration of a triaxial magnetometer into a helicopter UAV GPS-aided INS. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 2947–2960. [[CrossRef](#)]
21. Krozel, J.; Peters, M.; Bilimoria, K.D.; Lee, C.; Mitchell, J.S. System performance characteristics of centralized and decentralized air traffic separation strategies. *Air Traffic Control Q.* **2001**, *9*, 311–332. [[CrossRef](#)]
22. Jardin, M.R. Real-time conflict-free trajectory optimization. In Proceedings of the fifth USA/Europe Air Traffic Management R&D Seminar, Budapest, Hungary, 23–27 June 2003; p. 85.
23. Jardin, M.R. Analytical relationships between conflict counts and air-traffic density. *J. Guid. Control. Dyn.* **2005**, *28*, 1150–1156. [[CrossRef](#)]
24. Jardin, M.R. Grid-based strategic air traffic conflict detection. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, CA, USA, 15–18 August 2005; p. 5826.
25. Koeners, J.; de Vries, M. Conflict resolution support for air traffic control based on solution spaces: Design and implementation. In Proceedings of the 2008 IEEE/AIAA 27th Digital Avionics Systems Conference, St. Paul, MN, USA, 26–30 October 2008; pp. 3. C. 5-1–3. C. 5-9.
26. Kushleyev, A.; Mellinger, D.; Powers, C.; Kumar, V. Towards a swarm of agile micro quadrotors. *Auton. Robot.* **2013**, *35*, 287–300. [[CrossRef](#)]
27. Virágh, C.; Vásárhelyi, G.; Tarcai, N.; Szörényi, T.; Somorjai, G.; Nepusz, T.; Vicsek, T. Flocking algorithm for autonomous flying robots. *Bioinspir. Biomim.* **2014**, *9*, 025012. [[CrossRef](#)] [[PubMed](#)]
28. Vásárhelyi, G.; Virágh, C.; Somorjai, G.; Tarcai, N.; Szörényi, T.; Nepusz, T.; Vicsek, T. Outdoor flocking and formation flight with autonomous aerial robots. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 3866–3873.
29. Choi, H.H.; Nam, S.H.; Shon, T. Two tier search scheme using micro UAV swarm. *Wirel. Pers. Commun.* **2017**, *93*, 349–363. [[CrossRef](#)]
30. Amiri, A.; Samavati, F.; Peterson, P. Categorization and conversions for indexing methods of discrete global grid systems. *ISPRS Int. J. Geo-Inf.* **2015**, *4*, 320–336. [[CrossRef](#)]
31. Mahdavi-Amiri, A.; Alderson, T.; Samavati, F. A survey of digital earth. *Comput. Graph.* **2015**, *53*, 95–117. [[CrossRef](#)]
32. Tong, X.; Ben, J.; Wang, Y.; Zhang, Y.; Pei, T. Efficient encoding and spatial operation scheme for aperture 4 hexagonal discrete global grid system. *Int. J. Geogr. Inf. Sci.* **2013**, *27*, 898–921. [[CrossRef](#)]
33. Li, S.; Pu, G.; Cheng, C.; Chen, B. Method for managing and querying geo-spatial data using a grid-code-array spatial index. *Earth Sci. Inform.* **2019**, *12*, 173–181. [[CrossRef](#)]
34. Zhai, W.; Cheng, C.; Tong, X.; Chen, B. Subdivision R-Tree Index Model of the Earth-based Three-dimensional Subdivision Grids. *Geomat. Inf. Sci. Wuhan Univ.* **2016**, *41*, 443–449.
35. Hu, X.; Cheng, C. The three-dimensional data organization method based on GeoSOT-3D. In Proceedings of the 2014 22nd International Conference on Geoinformatics, Kaohsiung, Taiwan, 25–27 June 2014; pp. 1–4.
36. Li, S.; Cheng, C.; Pu, G. QRA-Grid: Quantitative Risk Analysis and Grid-based Pre-warning Model for Urban Natural Gas Pipeline. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 122. [[CrossRef](#)]
37. Zhang, H.; Cheng, C.; Miao, S. A Precise Urban Component Management Method Based on the GeoSOT Grid Code and BIM. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 159. [[CrossRef](#)]
38. Li, S.; Cheng, C.; Chen, B.; Meng, L. Integration and management of massive remote-sensing data based on GeoSOT subdivision model. *J. Appl. Remote Sens.* **2016**, *10*, 034003. [[CrossRef](#)]
39. Qi, K.; Cheng, C.; Hu, Y.n.; Fang, H.; Ji, Y.; Chen, B. An improved identification code for city components based on discrete global grid system. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 381. [[CrossRef](#)]
40. Cheng, C.; Tong, X.; Chen, B.; Zhai, W. A subdivision method to unify the existing latitude and longitude grids. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 161. [[CrossRef](#)]
41. Dempsey, M.E.; Rasmussen, S. *Eyes of the Army—US Army Roadmap for Unmanned Aircraft Systems 2010–2035*; US Army UAS Center of Excellence: Ft. Rucker, AL, USA, 2010; Volume 9.

