

Article

Three-Dimensional Flight Corridor: An Occupancy Checking Process for Unmanned Aerial Vehicle Motion Planning Inside Confined Spaces

Sherif Mostafa ^{1,*}  and Alejandro Ramirez-Serrano ²¹ Department of Electrical and Software Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada² Department of Mechanical and Manufacturing Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada; aramirez@ucalgary.ca

* Correspondence: sherif.mostafa@ucalgary.ca

Abstract: To deploy Unmanned Aerial Vehicles (UAVs) inside heterogeneous GPS-denied confined (potentially unknown) spaces, such as those encountered in mining and Urban Search and Rescue (USAR), requires the enhancement of numerous technologies. Of special interest is for UAVs to identify *collision-free Safe Flight Corridors* (SFC^+) within highly cluttered convex- and non-convex-shaped environments, which requires UAVs to perform advanced flight maneuvers while exploiting their flying capabilities. Within this paper, a novel auxiliary *occupancy checking* process that augments traditional 3D flight corridor generation is proposed. The 3D flight corridor is established as a topological structure based on a hand-crafted path either derived from a computer-generated environment or provided by the human operator, which captures humans' preferences and desired flight intentions for the given space. This corridor is formulated as a series of interconnected overlapping convex polyhedra bounded by the perceived environmental geometries, which facilitates the generation of suitable 3D flight paths/trajectories that avoid local minima within the corridor boundaries. An occupancy check algorithm is employed to reduce the search space needed to identify 3D obstacle-free spaces in which their constructed polyhedron geometries are replaced with alternate convex polyhedra. To assess the feasibility and efficiency of the proposed SFC^+ methodology, a comparative study is conducted against the *Star-Convex Method* (SCM), a prominent algorithm in the field. The results reveal the superiority of the proposed SFC^+ methodology in terms of its computational efficiency and reduced search space for UAV maneuvering solutions. Various challenging confined-environment scenarios, each with different obstacle densities (confined scenarios), are utilized to verify the obtained outcomes.



Citation: Mostafa, S.; Ramirez-Serrano, A. Three-Dimensional Flight Corridor: An Occupancy Checking Process for Unmanned Aerial Vehicle Motion Planning Inside Confined Spaces. *Robotics* **2023**, *12*, 134. <https://doi.org/10.3390/robotics12050134>

Academic Editors: Gerardo Flores, Hector M. Becerra, Juan-Pablo Ramirez-Paredes and Alexandre Brandão

Received: 8 September 2023

Revised: 23 September 2023

Accepted: 25 September 2023

Published: 29 September 2023



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

Keywords: flight corridor; confined spaces; indoor environments; hyperplanes; convex polyhedron; UAV flight; motion planning

1. Introduction

Unmanned Aerial Vehicles (UAVs) have emerged as a versatile technology with significant potential for various applications. Though effective, with further development, UAVs are envisioned to increase their usage particularly in environments that are hazardous or challenging to penetrate for humans, military personnel, and ground robotic systems. The operation of UAVs within confined environments, such as underground installations, mines, and geometrically complex urban spaces (e.g., collapsed buildings) is of specific interest. Such spaces present unique challenges in terms of path planning, exploration/reconnaissance, navigation, and mobility. The ability to navigate within these and other confined spaces efficiently is crucial for applications such as mining [1,2], Urban Search and Rescue (USAR) [3], and military applications [4–6]. Figure 1 exemplifies applications where such UAV abilities are required in underground mines and sewers, where

data collection, repair, and maintenance, for example, are challenging, hazardous, and time-consuming, putting human workers at risk. The complex nature of these environments, characterized by obstacles and openings with non-convex geometrical shapes, and limited space necessitate the development of specialized technologies to ensure safe, collision-free flight.

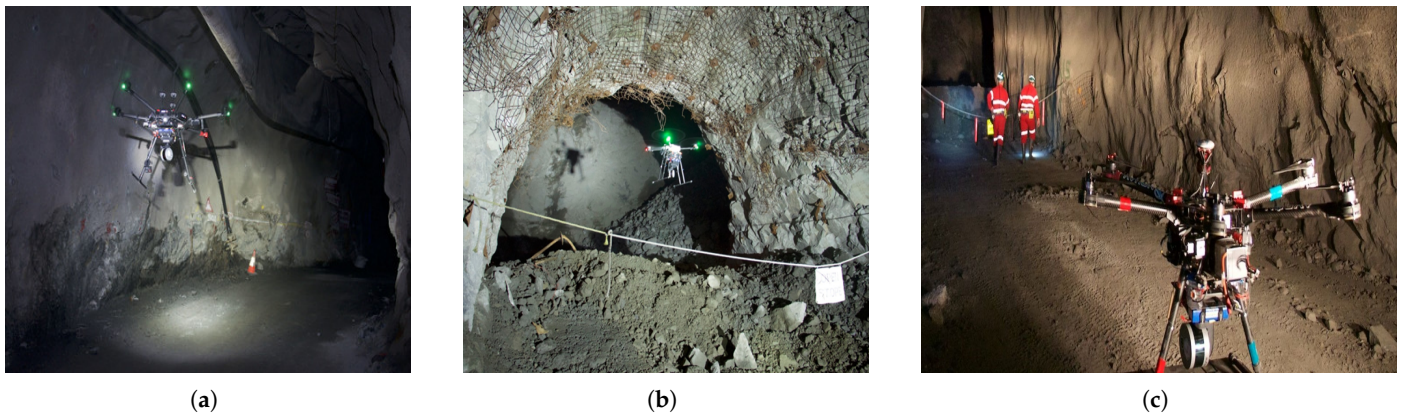


Figure 1. Surveying in geometrically complex unstructured spaces [7–9]: (a) underground tunnels; (b) collapsed underground installations; (c) mines.

In recent years, notable advancements in UAV data processing and flight abilities control, as well as future research directions [10], have been achieved in addressing the multifaceted challenges associated with UAV operations within confined environments. Of particular importance is the generation of 3D flight corridors, which establish secure pathways for UAVs to navigate within constrained spaces [11–15]. However, current methodologies cannot be easily applied in confined chaotic spaces where the flight corridor would require the embodiment of a somewhat complex topological framework that incorporates various environmental constraints. This process needs to adequately account for the potentially intricate geometry of the environment and potentially moving obstacles present in it. Consequently, within the context of this paper, a 3D flight corridor is regarded as a 3D tunnel characterized by varying cross-section profiles. In confined spaces, flight corridors tend to be relatively compact in volume, underscoring the significance of generating a sufficiently voluminous flight corridor while considering the available flight maneuvers that the UAV of interest possesses. In this paper, an advanced flight corridor generation mechanism is proposed that is capable of generating effective geometrically complex tunnels based on the perceived environment, a target goal, and the geometrical physical characteristics of the UAV. The proposed methodology draws upon the existing work of Liu [11], which models the 3D environment using an occupancy grid map. Leveraging this foundation, the newly developed algorithm, termed the Collision-Free Safe Flight Corridor or *SFC⁺* for short, demonstrates improved computational efficiency while yielding superior performance in terms of facilitating flight paths capable of being executed within the confined environment, as substantiated through a comparative evaluation against the known *SCM* algorithm across diverse environmental scenarios.

2. Prior Work (Literature Review)

Several methods have been proposed to generate 3D flight corridors for UAVs operating in diverse environments [16–21]. However, the existing methodologies have limitations in terms of their ability to handle complex non-convex geometries (and thus be applied to complex confined spaces). They cannot handle dynamic obstacles/entities present in the environment (and thus cannot change/morph in real-time, allowing UAVs to replan their flight trajectories in real-time) or search for optimal paths within confined spaces. Furthermore, the existing methods (e.g., [16–21]) do not consider the preferences and intentions

of the UAV operator, which, in some cases, might be considered crucial for effective UAV operations in confined hazardous environments (e.g., buildings on fire [22]).

There are numerous algorithms for obtaining nearly minimal convex decompositions of both 2D and 3D environments focused on creating convex or nearly convex covers. Lien [16] proposed an algorithm for segmenting non-convex polygons with polygonal holes with the goal of minimizing the number of polygons representing the environment. Similarly, Mamou [23] converted a triangulated 3D mesh into a collection of approximately convex pieces by iteratively clustering mesh faces based on the heuristics related to convexity and the aspect ratio. Such an approach, however, can only be applicable to spaces/regions with certain characteristics such as convex geometrical volumes. In contrast, ref. [17] described an approach that is applicable to spaces of arbitrary dimensions that can compute a set of environmental cuts that divide obstacles into approximately convex pieces. These methods are not well-suited for convex optimization within obstacle-free spaces as they require convex regions, which, when applied to any arbitrary environment, might result in regions thought to be empty (obstacle-free) but that do intersect with the obstacle set.

Although effective in geometrically complex spaces meeting certain geometrical criteria, such algorithms are computationally expansive. However, for the past 20 years, new methodologies have been developed with polynomial-time approximations. Eidenbenz et al. [18] described an algorithm that computes a nearly minimal set of overlapping convex pieces for a polygon with holes. Although their approach provides results within a logarithmic error bound relative to the number of vertices in the environment, it requires a high running time of $O(n^{29} \log n)$, where n represents the number of vertices in the polygon. Feng [19] presented an approach that divides an input polygon with holes into pieces and generates a tree structure of adjacent pieces. While this is a promising approach, their algorithm is limited to 2D cases. There have been convex decompositions that do not aim to find the minimum number of segments. For instance, Sarmiento [24] generates convex polytopic regions in N dimensions by sampling points in free space and checking their visibility from a set of “guard” positions. Unfortunately, this requires a set of samples covering the workspace, which prevents subsequent optimizations from being performed without any consideration for obstacle positions, resulting in a high computational time.

In the context of finding polyhedra, Deits [20,21] introduced the Iterative Regional Inflation by Semi-Definite Programming algorithm (IRIS), which allows users to choose a starting point on a terrain map to identify a maximum-volume ellipsoid contained within a polyhedron defined by hyperplanes. The main drawbacks of IRIS are its high computational effort to find the largest possible ellipsoid in the environment and a deficiency in finding a proper representation of the obstacles present within the environment. Another method employed for generating obstacle-free convex regions in motion planning is Stereographic Projection [25]. This approach relies on spherical projections utilizing convex hull generation and inverse vertex enumeration as subprocedures. One notable limitation of Stereographic Projection is that the selection of the initial point significantly impacts the volume of the resulting polyhedron. In contrast to Stereographic Projection, Zhong et al. [26] employed a technique known as sphere flipping to map original points into a nonlinear space. The algorithm calculates the convex hull of the mapped points and inversely maps the vertices of the hull into the Cartesian space. As with other approaches, this approach is of high computational complexity.

Based on the limitations of the existing research, there is a need to develop novel formulations to generate effective flight corridors in geometrically complex spaces.

Proposed Methodology

In this paper, an improved methodology for generating Collision-Free Safe Flight Corridors is introduced, which is referred to as the SFC^+ . The proposed approach ensures the safety of the UAV's flying operations, particularly in confined regions. The SFC^+ formulation enhances 3D flight corridor generation by incorporating a novel auxiliary

occupancy checking property (to be described in Section 3.1.4). Initially, the 3D flight corridor is established as a topological structure based on a hand-crafted path derived from the perceived environment as sensed by the robot and the map of the environment as either artificially generated as a computer-generated environment or sensed by the UAV's on-board sensors. The hand-crafted path is used to capture any human preferences (if available) and intentions for how it suggests the robot should move through the given space. The corridor consists of interconnected overlapping convex polyhedra bounded by the perceived environmental geometries, enabling UAV maneuvering solutions to be found that avoid local minima within the corridor boundaries. Additionally, an occupancy check algorithm is employed to identify obstacle-free spaces, allowing for construction of polyhedron geometries with convex polyhedra while maintaining a reasonable safety margin to accommodate for any potential UAV's changing geometry (e.g., reconfiguration) during flight. Thus, the methodology is envisioned to enable UAVs to fly through tight spaces in a similar fashion to how birds of prey (e.g., falcons) reconfigure their wings as they fly through tight spaces (e.g., three branches).

This article is structured as follows: Section 3 introduces the proposed SFC^+ methodology for generating the 3D flight corridor and discusses the novel occupancy checking algorithm. In Section 4, the results of exemplary numerical simulations are presented, including a comparative study with the SCM algorithm, followed by a detailed examination of the simulation outcomes. Lastly, Section 5 provides the conclusions and summarizes the key findings and contributions of this research endeavor.

3. Novel 3D Flight Corridor Generation

The proposed 3D flight corridor generation process constitutes a vital component of an ongoing research endeavor, shown in Figure 2, aiming to develop mechanisms capable of facilitating the navigation of highly maneuverable UAVs within exceedingly confined and chaotic regions, such as those encountered inside collapsed buildings, mines, and unstructured tunnels. The work seeks to enable these UAVs to adeptly maneuver through narrow corridors, intricate entry and exit points, and dynamically reposition their body orientation, such as hovering while in a pitched attitude, to navigate complex environmental geometries that may also contain other mobile entities, including robots and humans. In order to realize the aforementioned objectives, an *Improved "Teach-Repeat-Replan" (I-TRP)* strategy was formulated. Figure 2 illustrates the proposed methodology, drawing inspiration from the TRP work conducted by Gao, Wang, et al., in which optimal trajectories were devised for navigating intricate environments [14,15]. The architectural framework of the proposed system embodies a combination of offline global and online local path-planning approaches, thus constituting a hybrid methodology consisting of three major phases: (i) teach, (ii) repeat, and (iii) replan (see Figure 2). This approach entails leveraging precomputed global paths along with real-time local path-planning techniques to enable efficient navigation within complex and confined environments. The *Teaching* phase (red box in Figure 2) is achieved by using hand-crafted flight paths provided by a human operator. Such a path captures the preferences and intentions of the human pertaining to its intentions and the environment. Thus, the hand-crafted path does not have to be perfect as it serves as foundational information from which any autonomous navigation and 3D tunnel generation can commence. For the user or computer to generate the hand-crafted path, a map of the environment is needed. Such a map can be derived either from a computationally generated simulated environment or through the extraction of information via a 3D global Simultaneous Localization and Mapping (SLAM) procedure—an offline methodology commonly employed in autonomous vehicle navigation to concurrently establish a map and ascertain vehicle localization within that map, thereby enabling the mapping of unknown environments. Owing to the inherent complexities associated with aerial navigation within intricate spaces, the manually crafted path is susceptible to erratic movements and inaccuracies concerning the aircraft's flight capabilities. To mitigate these challenges, the *Repeating* phase (blue box in Figure 2) undertakes the construction of a

3D aerial corridor around the hand-crafted path. Such a corridor is generated based on the geometric characteristics of the environment and the flying capabilities of the UAV. Subsequently, the algorithm transforms the initially devised path into an optimal and seamless topological equivalent path using a *global planning* algorithm. To further bolster planning efficacy and global navigation, a *local planning* algorithm equipped with real-time collision detection and obstacle avoidance is used. Such local path planning is part of the *Replanning* phase (orange box in Figure 2) comprising the proposed *I-TRP* procedure. The *Replanning* phase generates a posterior optimal dynamically viable path, utilizing terrain data obtained through the UAV's onboard sensors and a 3D local SLAM approach. The result of the *I-TRP* process is a 3D flight corridor and a path that defines the position and orientation of the UAV along a smooth path derived from the initial hand-crafted path. The resulting posterior optimal reference path is then forwarded to the path-following control algorithm module, tasked with generating appropriate guidance commands for execution by the UAV. These guidance commands encompass the desired UAV position, attitude, velocity, and accelerations, which are distinct from the control commands that regulate fin deflections, thrust forces, and thrust moments. The *I-TRP* framework operates iteratively until the UAV successfully reaches its intended destination.

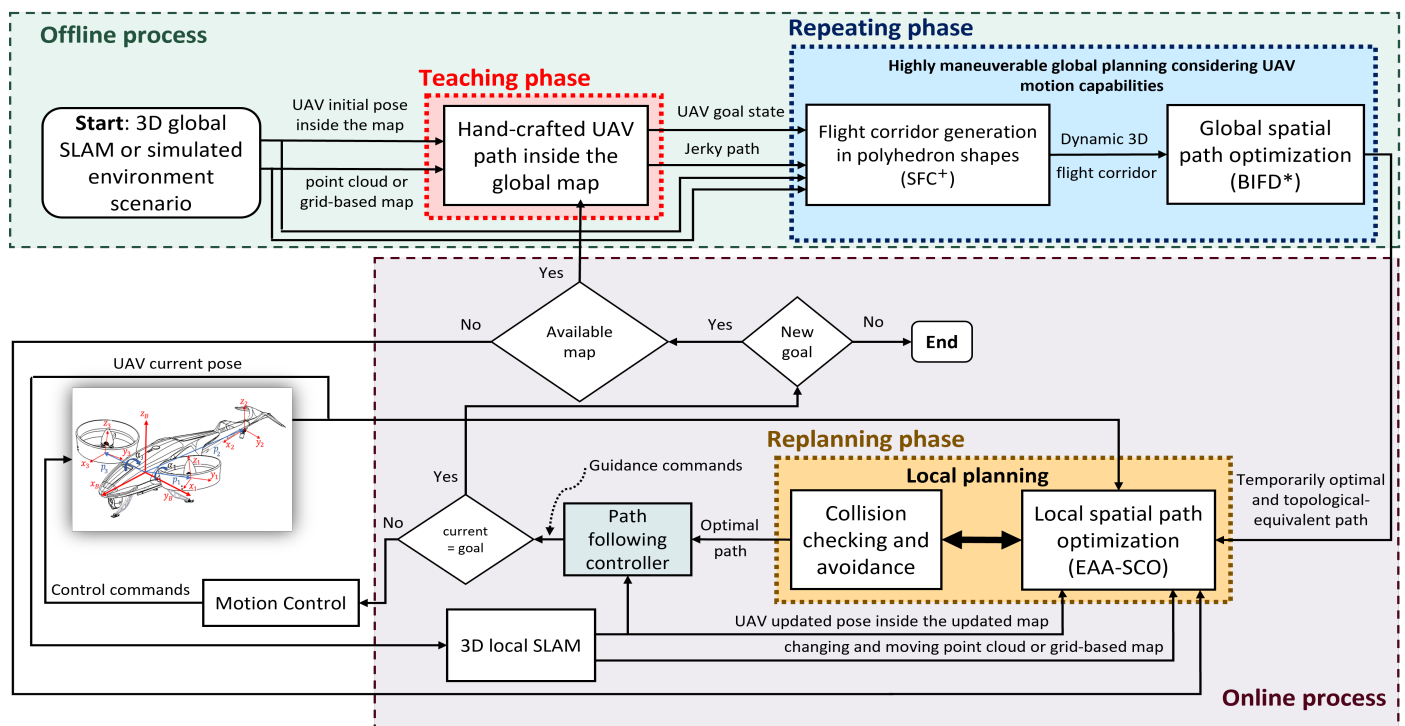


Figure 2. Proposed software architecture for the *I-TRP* autonomous motion planning strategy.

This research paper focuses on the 3D flight corridor generation module, as shown in Figure 3, which is part of the *Repeating* phase shown in Figure 2. For simulation purposes, a global environment map (representing the confined space of interest) is a computer-generated environment with static and dynamic obstacles that can be customized to test different levels of environmental complexities. The 3D corridor generation approach follows a seven-step sequential process (see Figure 3) consisting of:

- (i) Given an environmental map, generate a hand-crafted flight path composed of multiple path segments.
- (ii) Generate suitable maximized ellipsoid geometries containing the hand-crafted path segments while avoiding obstacles.
- (iii) Define a set of hyperplanes for each path segment and their corresponding half-spaces based on the generated ellipsoids.

- (iv) Extract the intersection of the hyperplanes obtained from step (iii) as obstacle-free convex polyhedra surrounding each segment of the path.
- (v) Perform a cuboid bounding surrounding the UAV based on its dimensions to reduce the number of collision checks required during the polyhedra construction process (step (vi)).
- (vi) Add an augmenting occupancy checking process to reduce the search space by identifying regions free from obstacles and replace the previously constructed convex polyhedra with alternate reduced-in-volume convex polyhedra.
- (vii) Generate the 3D Collision-Free Safe Flight Corridor.

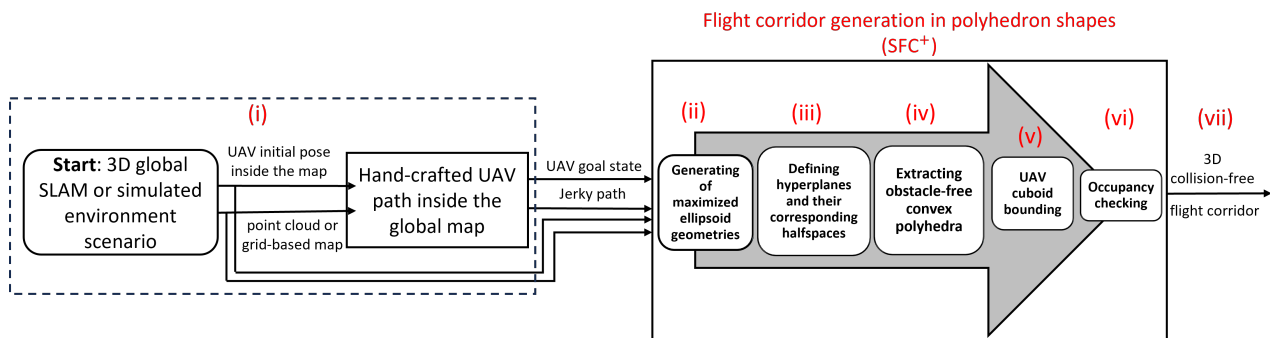
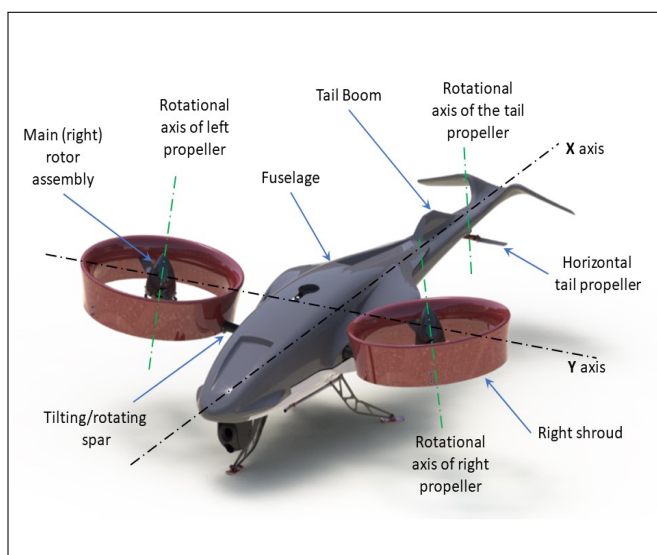
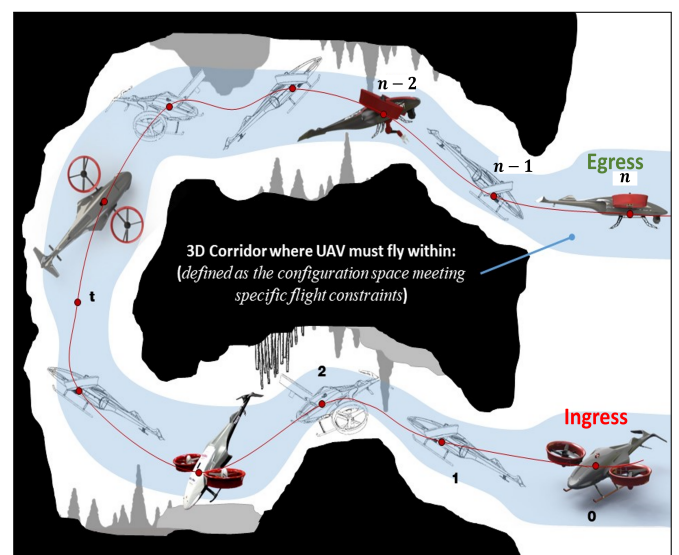


Figure 3. 3D flight corridor generation.

Figure 4b portrays a schematic depiction of the *Navig8* UAV flying inside traditionally impossible spaces. The *Navig8* (Figure 4a) is a highly maneuverable UAV serving as the case study for this research. Such a UAV has been purposefully developed for operation within restricted spaces (e.g., [27,28]). Herein, the definition of confined spaces presented in [29] as environments with an environmental density, or ζ , of greater than 0.1 ($\zeta > 0.1$) is used. Thus, maneuvering inside confined spaces is more complex when compared to moving within cluttered spaces that have a lower environmental density, $0.05 < \zeta \leq 0.1$ (see [29]).



(a) UAV.



(b) UAV flying inside a confined space.

Figure 4. UAV to be used in the proposed research.

3.1. Generation of Convex Polyhedra

The fundamental concept underlying the generation of the proposed flight corridor revolves around constructing a topological flight tunnel (or corridor) based on a hand-

crafted path, denoted as P_{hcr} . The purpose of this path is to capture the way a human pilot would perceive and navigate through the environment, ensuring compatibility with the UAV's maneuvering capabilities. The hand-crafted path is defined as a user-defined sequence of waypoints, represented as $P_{hcr} = (P_1, P_2, \dots, P_N)$, where “ N ” represents the total number of waypoints that form (“ $N - 1$ ”) path segments. To construct the flight corridor, an environmental map is represented as a grid occupancy map. Each i^{th} path segment comprising P_{hcr} is denoted as $\overrightarrow{P_i P_{i+1}}$ where $i = 1, 2, \dots, N - 1$. It is treated as a seed segment for initiating the construction of a convex polyhedron, denoted as CP_i , around segments of interest. Drawing from the work reported in [30], a convex set is defined as a vector space over $\mathbb{R}$ where the line segment connecting any pair of points lies entirely within the set. Hence, each polyhedron CP_i is regarded as a convex hull consisting of collision-free space, which is considered an optimal approach for a UAV motion planning solution. These convex hull polyhedra are then sequentially connected to form a flight corridor (FC), as represented by Equation (2). Each two consecutive polyhedra CP_i and CP_{i+1} need to have an intersecting area/volume that ensures the continuity of the flight corridor. Figure 5 provides an illustrative example of a given path P_{hcr} connecting a start and goal flight points with its corresponding flight corridor, $FC(P_{hcr})$, offering a visual representation of the concept.

The construction of the convex polyhedra around each hand-crafted path segment involves a three-stage process. Firstly, an ellipsoid serving as an initial approximation for the polyhedra shapes is generated around each path segment. Each ellipsoid is created to fully encompass the corresponding path segment. Subsequently, a subset of hyperplanes that are tangent to a sequence of dilated ellipsoids is identified in which these ellipsoids touch any of the surrounding obstacles in the environment (detailed in Section 3.1.1). To further enhance flight safety within the flight corridor generation process, the obstacles comprising the environment are dilated (increased), as visually depicted in Figure 5 (highlighted in gray). This slight increase in the size/volume of the obstacles serves to create an illusion of enlarged or “inflated” obstacles that facilitate the maneuverability of the UAV within the 3D space. These hyperplanes represent the faces of the resulting polyhedra shapes. By finding the tangent hyperplanes, the polyhedra can be formally defined, and their structural characteristics are fully determined (see Figure 5b). To optimize the computational efficiency of the algorithm, a cuboid bounding box is fitted around the path segments. This bounding box serves the purpose of reducing the number of collision checks required during the polyhedra construction process. Instead of checking for collisions with all surrounding obstacle points, the algorithm focuses solely on the points within the cuboid bounding box. This approach significantly reduces the computational burden by limiting the collision checks to a smaller and relevant subset of points. A detailed explanation and discussion of the procedures involved in the decomposition of convex polyhedra are presented in the subsequent subsections, which provide a comprehensive understanding of the steps undertaken to generate the convex polyhedra shapes around the hand-crafted path segments.

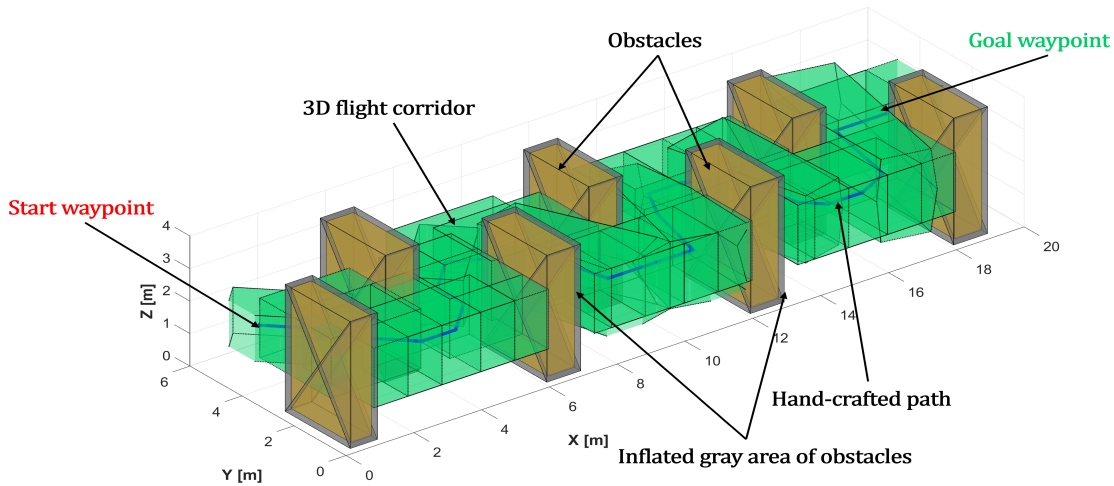
3.1.1. Generation of Ellipsoids

The objective of this step is to identify the empty space surrounding the path segments comprising P_{hcr} via ellipsoids. The aim is to find an ellipsoid that aligns its major axis with the corresponding path segment of interest $\overrightarrow{P_i P_{i+1}}$ while maximizing the volume contained within the ellipsoid and avoiding any obstacles in the environment. An ellipsoid in a 3D space ($\mathbb{R}^3$) is defined by a 3×3 symmetric positive definite matrix “ E ” and a vector “ d ”, where “ E ” represents the deformation of the sphere ($\|\bar{p}\| \leq 1$) and “ d ” represents the center of the ellipsoid as shown in Equation (1). “ E ” is decomposed as $E = R^T S R$ where “ R ” is the rotation matrix aligning the ellipsoid axis to the map axis and $S = \text{diag}(a, b, c)$ is the diagonal scale matrix whose diagonal elements stand for the corresponding lengths of ellipsoid semi-axes. Without loss of generality, it is assumed that $a \geq b, a \geq c$, as illustrated

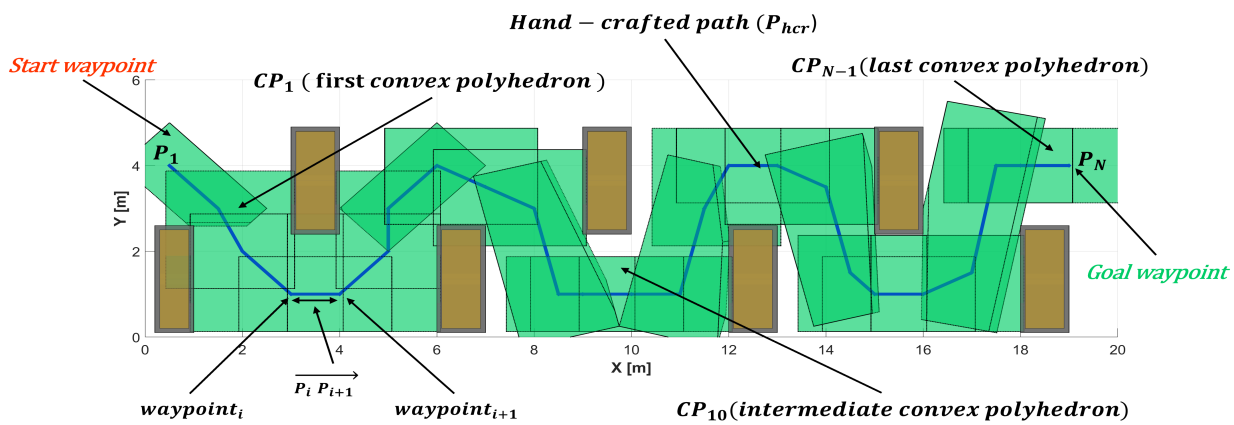
in [31]. The final goal is to find “ E ” and “ d ” given $\overrightarrow{P_i P_{i+1}}$ and “ O ” where “ O ” represents the obstacles around the segment $\overrightarrow{P_i P_{i+1}}$.

$$\tilde{\zeta}(E, d) = \{p = E\bar{p} + d \mid \|\bar{p}\| \leq 1\} \quad (1)$$

$$FC(P_{hcr}) = \{CP_i \mid i = 1, 2, \dots, N-1\}, \quad \text{where : } (N-1) \text{ is the number of } P_{hcr} \text{ segments.} \quad (2)$$



(a) CP_i (green convex hulls) around a path P_{hcr} generated from a set of connected manual waypoints.



(b) Projection of the 3D environment along the xy plane.

Figure 5. An example of generating a flight corridor $FC(P_{hcr})$ composed of connected polyhedra.

To determine a feasible ellipsoid the point cloud data representing the obstacles of interest in the vicinity of $\overrightarrow{P_i P_{i+1}}$ and the line representing $\overrightarrow{P_i P_{i+1}}$ itself are used. The process begins with a spheroid surrounding $\overrightarrow{P_i P_{i+1}}$, as depicted in Figure 6a by the solid blue sphere. By considering $\overrightarrow{P_i P_{i+1}}$ as the semi-major axis, the initial spheroid is gradually shrunk to obtain a maximal ellipsoid, and the ellipsoid does not contain any obstacle points within its boundaries (Figure 6c). This shrinking process involves iteratively identifying the closest obstacle point to the center of the ellipsoid. Eventually, a maximal ellipsoid is achieved with one closest obstacle point lying on its boundary, as illustrated in Figure 6c by the

dashed yellow ellipse. The xy -plane of the ellipsoid is formed based on $\overrightarrow{P_i P_{i+1}}$ and the closest point at the boundary. Depending on the P_i and P_{i+1} waypoints and the objects around such points, the obtained ellipsoid will contain one or more contact points with the obstacles. Regardless of the number of contact points, the obtained ellipsoid will be the biggest ellipsoid. Subsequently, the maximal ellipsoid is dilated along the z -axis. The appropriate length of the third axis in the z -axis direction is determined using a process identical to that describe above (Figure 6) for the ellipsoid's xy -plane along the z -axis. In this case, however, the ellipsoid is initially expanded along the z -axis without changing the ellipsoid in the xy -plane and then gradually shrunk until there are no obstacle points within its boundary. The pseudo-code detailing the complete process is provided in Algorithm 1. Two examples of the obstacle-free path, along with their corresponding ellipsoids, are depicted in Figure 7, illustrating that feasible ellipsoids are obtained that effectively capture the empty space surrounding the path segments. These ellipsoids play a crucial role in constructing the subsequent polyhedra, thereby enabling generation of the flight corridor.

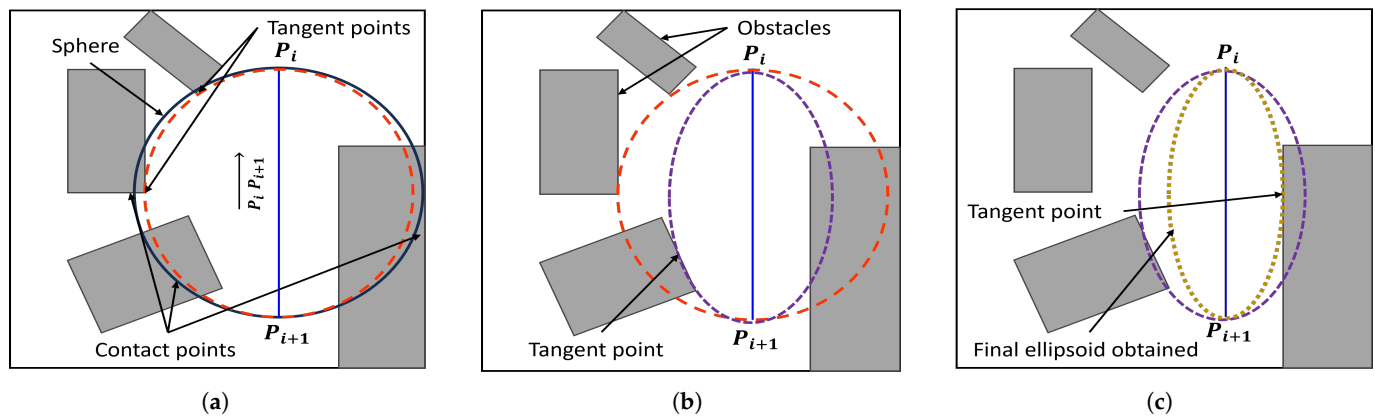


Figure 6. The process of finding an ellipsoid around $\overrightarrow{P_i P_{i+1}}$ by shrinking to form a collision-free one shown in 2D setting: (a) an initial solid blue sphere is surrounding a path segment $\overrightarrow{P_i P_{i+1}}$; (b) the initial sphere is gradually shrinking until it has no obstacle points by repeating the same shrinking process; (c) the final phase of obtaining the maximal collision-free ellipse in size/area (dashed yellow).

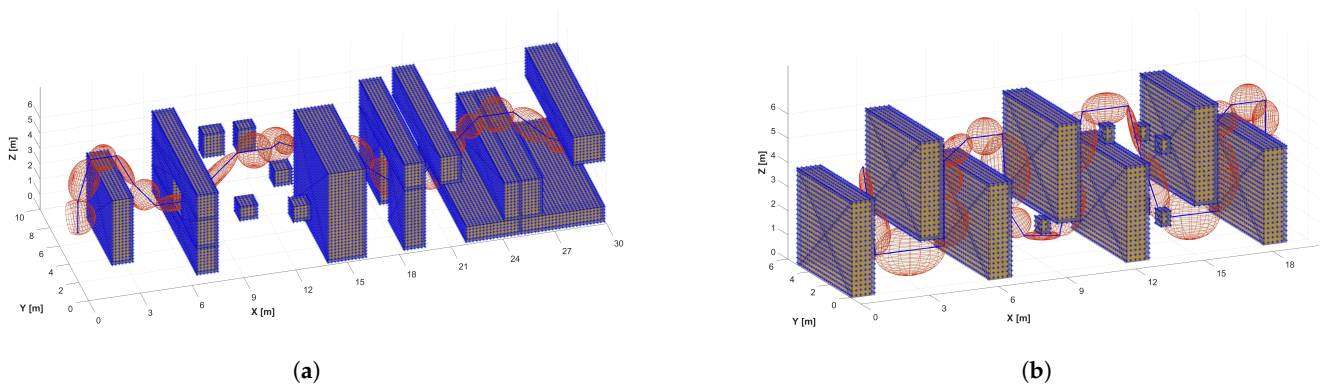


Figure 7. Examples for finding ellipsoids around the provided P_{hcr} segments in two environmental scenarios: (a) $30 \times 10 \times 6$ m environment with 18 convex and concave obstacles, some with opening holes, (b) $18 \times 6 \times 6$ m environment with 7 large and 5 small convex obstacles.

Algorithm 1: Pseudo-code of the proposed SFC^+ flight corridor generation process.

Input : O (Obstacle points), N (No. of P_{hcr} waypoints), Each path segment is denoted by $\overrightarrow{P_i P_{i+1}} \rightarrow L_i$

Output: $\{CP_i, \dots, CP_{i+1}\}$ (Convex Polyhedra forming SFC^+)

```

1 Function ENHANCED_SAFE_FLIGHT_CORRIDOR( $O, N$ ) is
2   for  $i \leq (N - 1)$  do
3     Function ELLIPSOID_GENERATION( $O, L_i$ ) is
4        $\zeta^i(E, d) \leftarrow \text{Define\_Initial\_Ellipsoid}(L_i)$ ;
5        $O_{inside} \leftarrow \text{Remove\_Points\_Outside}(\zeta^i, O)$ ;
6        $j \leftarrow 0$ ;
7       while  $O_{inside} \neq \text{Empty}$  do
8          $p_j^e \leftarrow \text{Closest\_Point}(\zeta^i, O_{inside})$ ;
9          $\zeta^i(E, d) \leftarrow \text{Shrink\_Ellipsoid\_Along\_XY}(\zeta^i, p_j^e)$ ;
10         $O_{inside} \leftarrow \text{Remove\_Points\_Outside}(\zeta^i, O_{inside})$ ;
11         $j \leftarrow (j + 1)$ 
12      end
13       $n_{xy} \leftarrow \text{Find\_XYPlane}(\zeta^i, p_j^e)$ ;
14       $\zeta^i(E, d) \leftarrow \text{Dilate\_Ellipsoid\_Z}(\zeta^i, n_{xy})$ ;
15       $O_{inside} \leftarrow \text{Remove\_Points\_Outside}(\zeta^i, O)$ ;
16       $k \leftarrow 0$ ;
17      while  $O_{inside} \neq \text{Empty}$  do
18         $p_k^e \leftarrow \text{Closest\_Point}(\zeta^i, O_{inside})$ ;
19         $\zeta^i(E, d) \leftarrow \text{Shrink\_Ellipsoid\_Along\_Z}(\zeta^i, p_k^e)$ ;
20         $O_{inside} \leftarrow \text{Remove\_Points\_Outside}(\zeta^i, O_{inside})$ ;
21         $k \leftarrow (k + 1)$ 
22      end
23      return  $\zeta^i(E, d)$ 
24    end
25    Function CONVEX_POLYHEDRON( $\zeta^i, O$ ) is
26       $O_{remaining} \leftarrow O$ ;
27       $j \leftarrow 0$ ;
28      while  $O_{remaining} \neq \text{Empty}$  do
29         $p_j^c \leftarrow \text{Closest\_Point}(\zeta^i, O_{remaining})$ ;
30         $\zeta^i(E, d) \leftarrow \text{Dilate\_Ellipsoid\_Z}(\zeta^i, p_j^c)$ ;
31         $\langle a_j, b_j \rangle \leftarrow \text{Find\_HalfPlane}(\zeta^i, p_j^c)$ ;
32         $j \leftarrow (j + 1)$ 
33      end
34       $CP_i : A \leftarrow [a_0, a_1, \dots], b \leftarrow [b_0, b_1, \dots]$ ;
35      return  $CP_i(A, b)$ 
36    end
37    Function OCCUPANCY_CHECKING( $CP_i, CP_{i+1}, O$ ) is
38      if  $O = \text{Empty}$  then
39         $CP_{i+1}(A, b) \leftarrow CP_{i+1}(A, b)$ ;
40      else  $CP_{i+1}(A, b) \leftarrow CP_i(A, b)$ ;
41      end
42      return  $CP_{i+1}(A, b)$ 
43    end
44  end
45 end

```

3.1.2. Intersection of Hyperplanes

Once a train of connected ellipsoids is generated (Section 3.1.1), such a set is used to generate obstacle-free polyhedra surrounding each segment of the path. Similar to the ellipsoid, the space enclosed within the boundaries of the polyhedron must be devoid of any obstacle points. For this, each ellipsoid comprising the train of connected ellipsoids is gradually dilated identifying any overlapping obstacles and defining a corresponding halfspace after each collision with an obstacle is identified. This process continues iteratively until all such intersected halfspaces collectively form a polyhedron that encompasses the segment of interest $\overrightarrow{P_i P_{i+1}}$ and the corresponding obstacle-free ellipsoid.

To formally define the obstacle-free polyhedron formation, consider an ellipsoid $\xi^0(E, d)$ that is slowly dilated until a (new) point p_0^c is found to touch a new obstacle at the boundary of the ellipsoid. At this boundary point, the tangent plane to the ellipsoid is defined as a halfspace, $H_0 = \{p | a_0^T p < b_0\}$, that contains the ellipsoid. By computing this halfspace, all obstacle points that remain outside the halfspace $O_{remaining}$ are removed. The ellipsoid is then further dilated, and the process is repeated upon encountering another obstacle point p_1^c , at which a new ellipsoid $\xi^1(E, d)$ is called, and a new tangent hyperplane is generated to construct a new halfspace H_1 . This iterative process results in a collection of halfspaces that collectively form a polyhedron CP , as denoted by Equation (3). The terms $\mathbf{p}$, $\mathbf{A}$, and $\mathbf{b}$ in Equation (3) represent a vector of the size 3×1 , representing the halfspace; a matrix of the size $3 \times m$, representing the 3D normal vectors of the hyperplanes; and a vector of the size $m \times 1$, representing the translation constants. The detailed parameters of $\mathbf{p}$, $\mathbf{A}$, and $\mathbf{b}$ are shown in Equation (3). The set of found halfspaces is $H = \{H_0, H_1, H_2, \dots, H_m\}$, where “ m ” is the number of tangent obstacle points found with regard to the segment $\overrightarrow{P_i P_{i+1}}$, and its corresponding ellipsoid represents the polyhedron associated with $\overrightarrow{P_i P_{i+1}}$. These elements are depicted in Figure 8 in which, for simplicity of explanation, a set of 2D images is used; however, the process of interest is 3D in nature. This polyhedron encompasses the segment and the obstacle-free ellipsoid. The pseudo-code for this process can be found in Algorithm 1. By applying this algorithm to each segment of the path, a set of polyhedra is derived, ensuring the path remains clear of obstacles. It should be noted that the hyperplane defining the j^{th} halfspace H_j is tangent to the ellipsoid $\xi^j(E, d)$ at the point p_j^c and is computed as shown in Equations (4) and (5):

$$CP = \bigcap_{j=0}^m H_j = \left\{ \mathbf{p} \mid \mathbf{A}^T \mathbf{p} < \mathbf{b} \right\}, \quad \text{where :} \quad \begin{cases} a_{01} \cdot x_h + a_{02} \cdot y_h + a_{03} \cdot z_h < b_0 \\ a_{11} \cdot x_h + a_{12} \cdot y_h + a_{13} \cdot z_h < b_1 \\ \vdots \\ a_{m1} \cdot x_h + a_{m2} \cdot y_h + a_{m3} \cdot z_h < b_m, \end{cases} \quad (3)$$

$$a_j = \left. \frac{d\xi^j}{dp} \right|_{p=p_j^c} = 2E^{-1}E^T(p_j^c - d), \quad \text{where : } a_j \text{ is the } j^{th} \text{ column of matrix } \mathbf{A}. \quad (4)$$

$$b_j = a_j^T p_j^c, \quad \text{where : } b_j \text{ is the } j^{th} \text{ element of vector } \mathbf{b}. \quad (5)$$

The gradual dilation of the ellipsoid continues until a set of polyhedron volumes is found that form a convex closed volume of obstacle-free space.

3.1.3. Cuboid Bounding

To guarantee that the biggest obstacle-free polyhedron is found, the proposed approach searches through all points in the obstacle set O at least twice when constructing the polyhedron CP_i for each $\overrightarrow{P_i P_{i+1}}$. This process, however, can be computationally expensive. To mitigate this issue, a cuboid bounding box is used around each $\overrightarrow{P_i P_{i+1}}$ segment. By using this, the search for obstacle points is limited to those contained within the cuboid bounding box, thus significantly reducing the number of points that need to be checked. The cuboid bounding box associated with a path segment consisting of six rectangular

surfaces aligned with the segment's axis. The size of the cuboid is defined based on the size/dimension of the UAV targeted to travel within the confined space. Specifically, the length, width, and height of the cuboid are set to double the corresponding dimensions of the UAV, as illustrated in Figure 9. This configuration ensures that the generated flight corridor maintains all necessary safety requirements to facilitate the maneuverability of the UAV within a 3D space. By introducing the cuboid bounding box, the process achieves a more efficient search process by focusing on relevant obstacle points, thereby improving the computational efficiency of the algorithm.

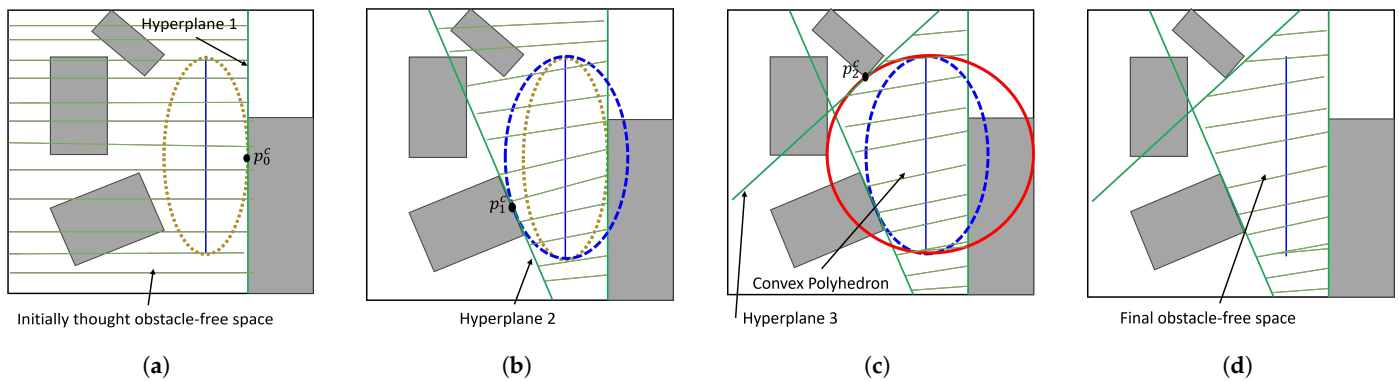


Figure 8. The process of finding a convex polyhedron CP_i around $\overrightarrow{P_i P_{i+1}}$: (a) an initial dashed yellow ellipse touching an obstacle point to generate the first hyperplane; (b) a dashed blue dilated ellipse until touching the second obstacle point to generate the second hyperplane; (c) a third dilated ellipse (solid red) to generate the third hyperplane; (d) the final obstacle-free polyhedron (dashed green).

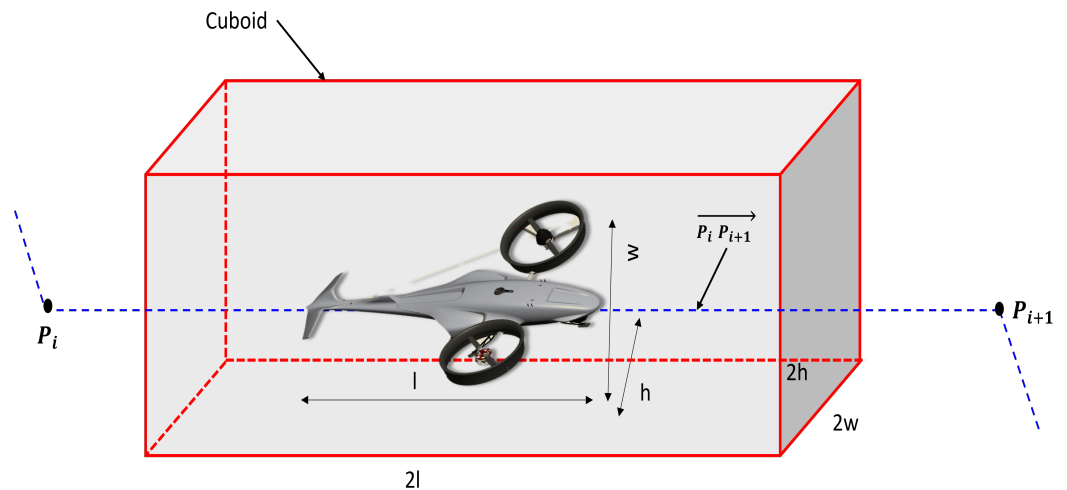


Figure 9. Size of the cuboid based on the UAV's dimensions.

Figure 10 illustrates the set of bounding boxes (in 2D) applied to the environment shown in Figure 8.

3.1.4. Occupancy Checking

As described in the previous section, when a robotic device (e.g., UAV) is targeted to move inside confined spaces, it is of high importance to generate flight corridors that adequately encompass the intended flight path, taking into account potential variations in the UAV's reconfiguration states and maximizing the volume within which the UAV can move. Thus, the proposed flight corridor uses a novel auxiliary *occupancy checking* property specifically tailored for confined spaces, which guarantees that the flight corridor is safe and

generates a corridor size/volume that is enough for the UAV flight within it. The proposed mechanism reduces the search space by identifying regions free from obstacles that do not need to be revised and handles the problem of unbounded corridor boundaries in the case when free space defines the flight space. In these regions, the previously constructed polyhedron geometries are replaced with alternate convex polyhedra, which ensures an appropriate safety margin that considers the potentially changing geometry of the UAV during flight. The pseudo-code outlining the process, shown in Figure 11, is provided in Algorithm 1.

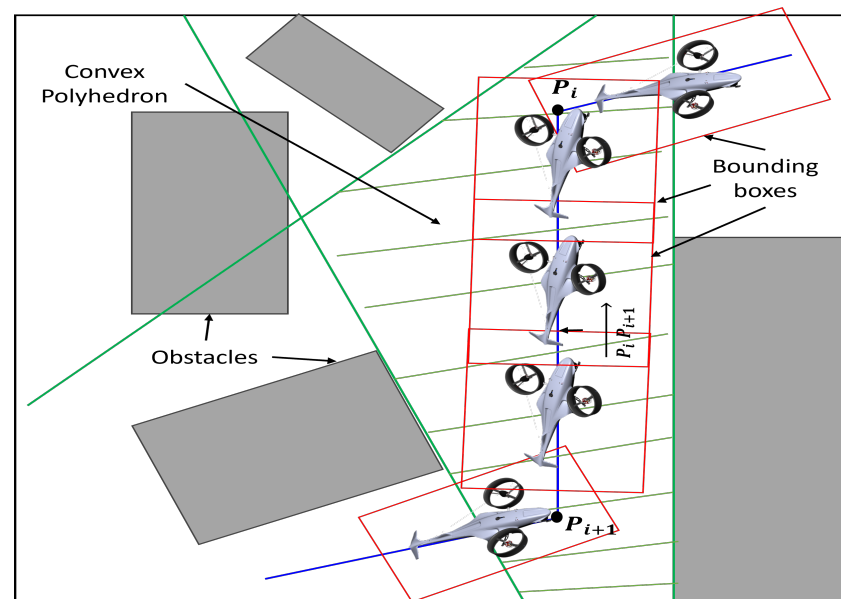


Figure 10. Generation of bounding boxes.

The *occupancy checking* process commences with the continuous examination of the point cloud map that represents the spatial environment for the presence of any obstacle points within the designated flight corridor. Subsequently, upon the absence of detected obstacles within the scanning region of the designated space, a novel function responsible for the generation of alternative polyhedra, P_{alter} , is invoked to replace the original polyhedra. The mathematical formulation employed for constructing these alternative polyhedra within the obstacle-free space is articulated in the following Equation (6). The mathematical representation of polyhedrons adopts the halfspace/hyperplane representation (referred to as the H-representation), which is also known as the inequality representation, by characterizing the polyhedron as the set of solutions that satisfy a finite set of linear inequalities and linear equalities simultaneously. The terms in Equation (6) are defined as follows: $\mathbf{x}$ represents the halfspace solution with a vector of the size 3×1 ; $\mathbf{A}$ represents the 3D normal vectors of the hyperplanes, which in this case exemplifies the six polyhedron faces with a vector of the size 6×3 ; and $\mathbf{b}$ represents the offsets or the translation constants with a vector of the size 6×1 .

$$P_{alter} = \left\{ \mathbf{x} \in \mathbb{R}^3 \mid \mathbf{A}\mathbf{x} \leq \mathbf{b} \right\} \quad (6)$$

Furthermore, Figure 11 illustrates an instance of constructing such alternative polyhedra in an environment devoid of obstacles, surrounding each segment $\overrightarrow{P_i P_{i+1}}$ of the hand-crafted path. Figure 12 demonstrates the visual representation elucidating the distinction between the proposed SFC^+ and the SCM algorithms, particularly in terms of the reduction in the size/volume of the generated flight corridor within an unobstructed space.

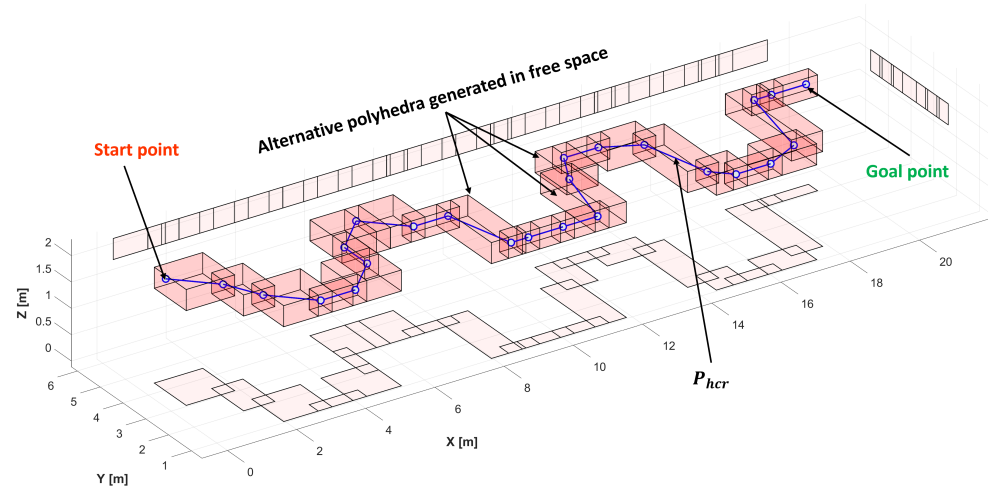
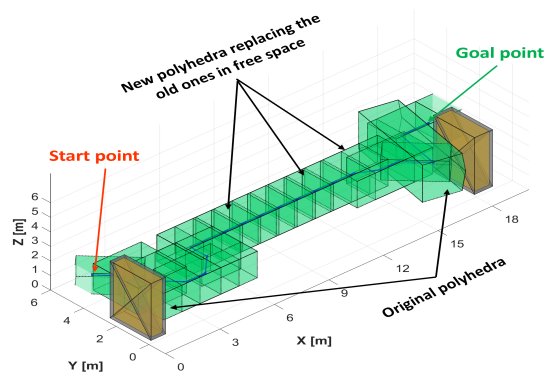
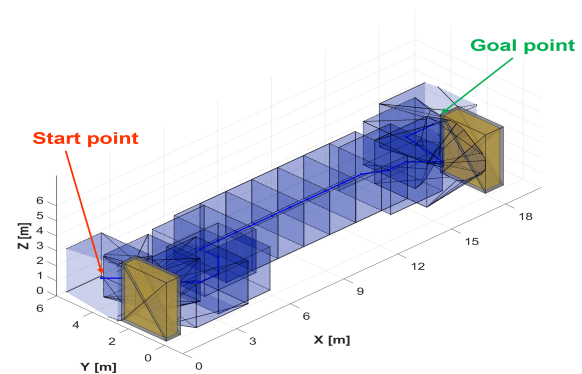


Figure 11. Example of generated polyhedra in an obstacle-free space using *occupancy checking* process.



(a) Proposed SFC^+ , $Vol_{corSFC^+} = 107.7698 \text{ m}^3$



(b) SCM, $Vol_{corSCM} = 224.1295 \text{ m}^3$

Figure 12. Comparison for the generated corridor volume between SFC^+ and SCM.

3.1.5. Final Flight Corridor

The result of the occupancy checking process is a flight corridor that the UAV can use to fly from its current state to a desired final destination. Figure 13 shows the final flight corridor obtained using the collapsed building environment shown in Figure 7a.

3.2. Calculation of Corridor Volume

To assess the effectiveness of the proposed approach's ability to generate a suitable flight corridor for confined spaces that takes into account the physical characteristics of the UAV and the user's flight intentions, a comparative study is conducted, contrasting it with the widely recognized SCM algorithm for flight corridor generation. For this, the difference in the corridor's volume is compared, which in some way represents the search space used for finding the robot's (e.g., UAV's) maneuvering solutions. The computational time and the number of polyhedra needed to generate the corresponding flight corridor are also used to evaluate the effectiveness of the proposed approach. The computational time needed to generate effective solutions for aircraft flight is critical for effective real-time operation of UAVs inside confined spaces, thus careful attention is paid to such an aspect. The comparative analysis with the SCM algorithm demonstrates the superiority of the proposed method in terms of the corridor volume and computational time performance, further supporting its potential for real-world UAV operations in confined environments.

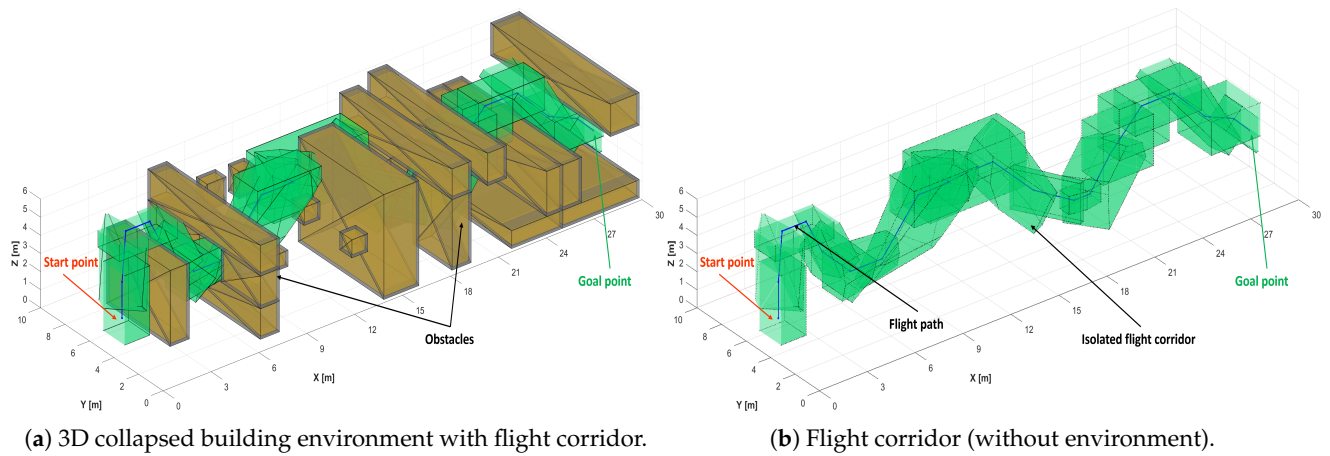


Figure 13. Example of the obtained flight corridor for a given confined environment.

To determine the 3D flight corridor volume for a given environment, the combined volume of intersecting sequential polyhedra is calculated. The process applied to the proposed SFC^+ and the SCM methodology is illustrated in Figure 14, using a somewhat simple environmental map scenario. The translucent green and blue regions in Figure 14 represent the generated 3D flight corridor, while the solid red and cyan volumes indicate the overlapping regions between adjacent convex polyhedra. The calculation of the corridor volume is accomplished using Equation (7), which provides an understanding of the spatial capacity available for UAV navigation within the given flight corridor. The example depicted in Figure 14 provides a visual representation of the corridor volume, highlighting the intricate interplay between the successive polyhedra and their overlapping regions. Determination of the flight corridor enables planners and operators to assess the feasibility and suitability of a given corridor for specific mission requirements that provide valuable insights into the available space for UAV navigation, aiding in the flight and control decision-making process for successful mission execution.

$$\text{Corridor Volume } (Vol_{cor}) = \sum_{i=1}^{N-1} CP_{ivol} - \sum_{i=1}^{N-1} (CP_{ivol} \cap CP_{i+1ivol}) \quad (7)$$

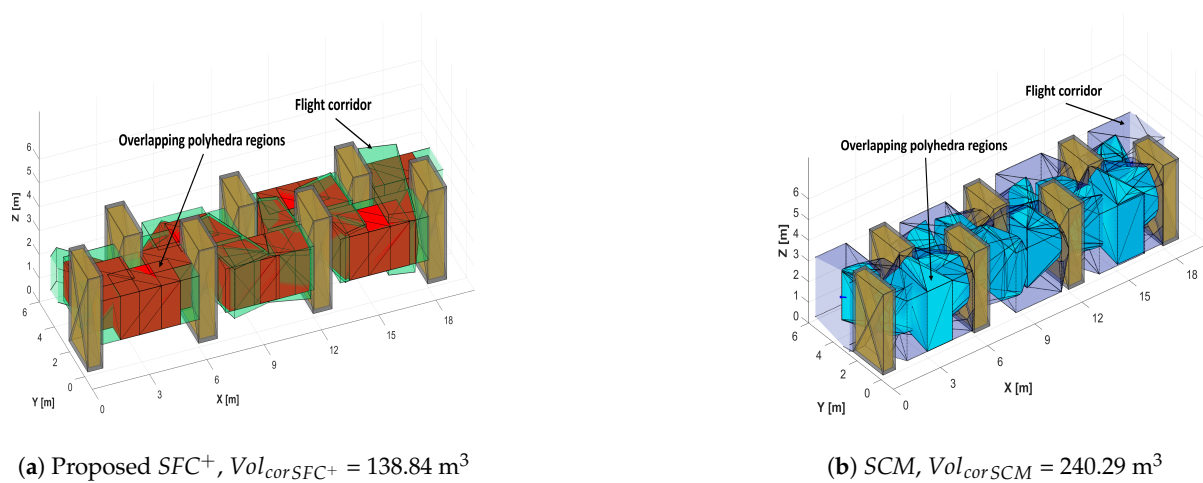


Figure 14. 3D flight corridor with intersected polyhedra regions.

4. Simulation Results and Discussion

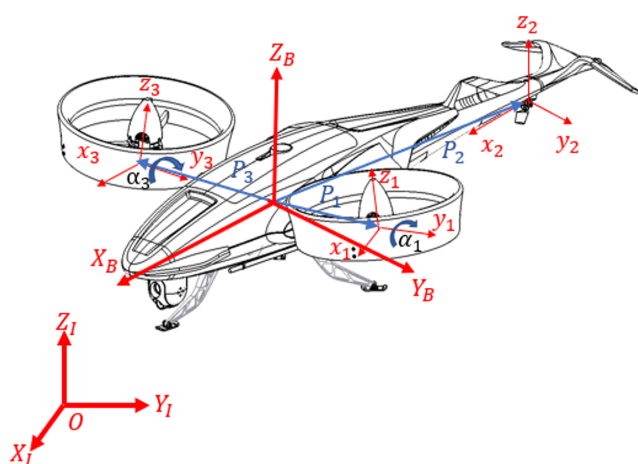
To evaluate the efficiency of the proposed SFC^+ algorithm in generating corridors in geometrically complex confined spaces for 3D flight, diverse environments with varying obstacle densities were employed. The selected environment scenarios were designed to

replicate complex, hazardous, and unstructured spaces encountered in real-life situations in which data collection is problematic. Furthermore, access to such spaces by traditional means (human entry) is risky with time constraints in which the use of UAVs provides a safe alternative.

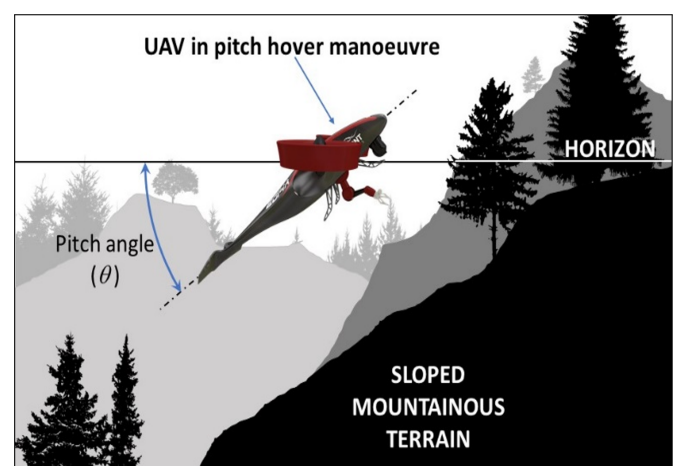
4.1. Selected UAV

In the context of the simulation, an aerial vehicle of notable versatility denoted as Navig8 (Navigate) (Figure 15a), which was purposefully engineered for operations conducted within structurally and non-structurally constrained environments, is employed. The Navig8 UAV represents the culmination of the fifth generation of a vertical takeoff and landing (VTOL) aircraft renowned for its exceptional maneuverability, a creation attributed to Dr. A. Ramirez-Serrano, with an accompanying patent pending. Notably, the Navig8 UAV embodies a scalable design featuring a twin-shrouded propeller configuration, facilitating VTOL capabilities that are optimally suited for engagements within the confines of intricate, structured, or unstructured environments. These environments encompass, but are not limited to, locations characterized by collapsed structures and spaces beneath a dense tree canopy, operating at low altitudes or in close proximity to critical infrastructure.

Consequently, the Navig8 UAV attains an elevated level of maneuvering proficiency, enabling the successful execution of intricate missions demanding heightened agility, such as the challenging pitched hover maneuver, which remains beyond the purview of currently available aircraft (as illustrated in the accompanying Figure 15b). Furthermore, the vehicle exhibits proficiency in executing other acrobatic maneuvers, exemplified by its capability to effectuate landings on dynamically unstable surfaces akin to those encountered on seafaring vessels navigating waters classified on the Beaufort scale at a 7–8 range, a quantifiable empirical scale spanning from 0 to 12 that associates wind speed with observed conditions at sea or on land (where 0 signifies tranquil waters characterized by wave heights of 0 m and 12 denotes hurricane-force conditions). In terms of its operational utility, the Navig8 UAV can be readily outfitted with standard sensor payloads, such as electro-optical/infrared (EO/IR) cameras and light detection and ranging (LIDAR) systems. Additionally, it accommodates mission-specific sensors, exemplified by gas detectors and environmental quality monitoring apparatuses, thereby enabling the generation of comprehensive environmental maps and facilitating real-time video streaming capabilities.



(a) Navig8 UAV model.



(b) Navig8 pitched hover maneuver.

Figure 15. Selected UAV (Navig8) for the simulation.

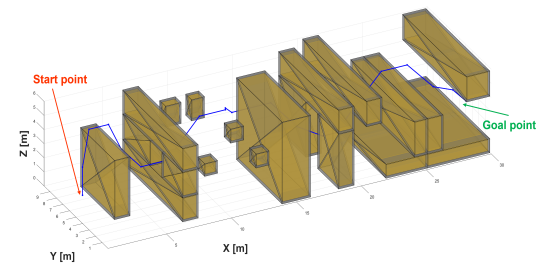
4.2. Testing Scenarios

To test and evaluate the proposed SFC^+ 3D corridor generation in confined spaces, numerous simulation tests were used. Here, however, the results of three representative

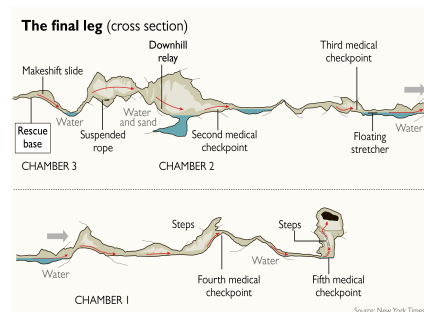
environments are provided (Figure 16): (a) a collapsed building (Figure 16a), (b) a cave (Figure 16c), and (c) an obstructed mine corridor (Figure 16e). Figure 16 also provides the 3D corresponding environments used in the simulations (Figure 16b, Figure 16d, and Figure 16f, respectively). Table 1 provides the overall characteristics of each environment including its environmental density, ζ , which describes its degree of confined space [29], and the coordinates of the entry and exit points. For each of these terrains, the degree of the confined space was also changed to test the generation of 3D flight corridors in similar terrains with different cluttered or confined properties, as defined in [29]. In [29], confined spaces are defined as environments with an environmental density, ζ , of greater than 0.1 ($\zeta > 0.1$). Thus, maneuvering inside confined spaces is more complex when compared to moving within cluttered spaces that have a lower environmental density, $0.05 < \zeta \leq 0.1$. Figure 17 demonstrates the visual difference between the confined and cluttered spaces. All numerical simulations are executed within the MATLAB environment, utilizing an Intel(R) Core (TM) i5-10210U CPU @ 1.60 GHz, 2.10 GHz with 12 GB of installed RAM and a 64-bit operating system.



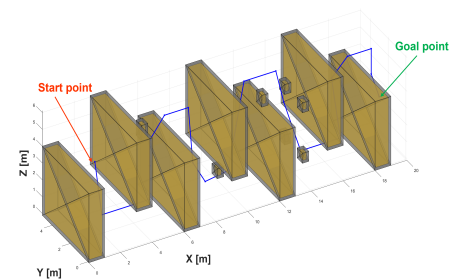
(a)



(b)



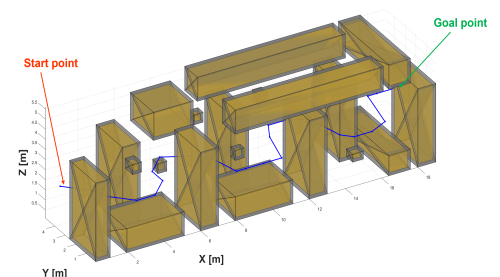
(c)



(d)



(e)

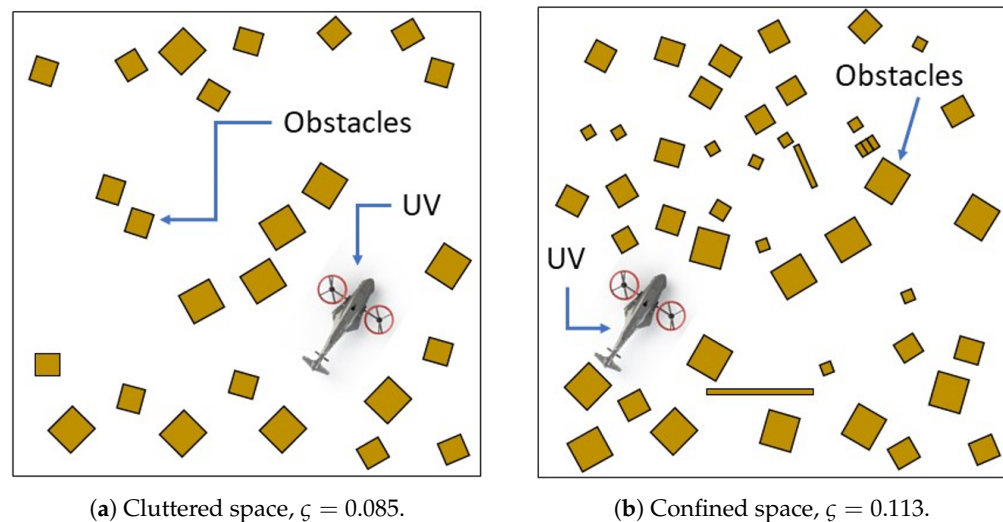


(f)

Figure 16. Simulated map scenarios and their counterparts in real life: (a) collapsed building [32]; (b) simulated collapsed building; (c) steep cave in Thailand [33]; (d) simulated steep cave; (e) obstructed mine corridor (photo from Exyn Technologies); (f) simulated obstructed corridor.

Table 1. Characteristics of the environments used.

Environment	Obstacle Density (ζ)	No. of Obstacles	Overall Size of the Space ($w \times h \times d$, m)	Coordinates of Entry Point (m)	Coordinates of Exit Points (m)
Collapsed building	$\zeta = 0.1645$ (confined space)	20	$30 \times 10 \times 6$	[0.5,4,2]	[30,4,2]
	$\zeta = 0.1233$ (confined space)	10			
	$\zeta = 0.081$ (cluttered space)	9			
Steep cave	$\zeta = 0.18567$ (confined space)	14	$20 \times 5 \times 6$	[1,1,5]	[20,3,2]
	$\zeta = 0.13325$ (confined space)	5			
	$\zeta = 0.08125$ (cluttered space)	3			
Obstructed mine corridor	$\zeta = 0.1455$ (confined space)	21	$20 \times 5 \times 6$	[0.5,4,2]	[19,4,2]
	$\zeta = 0.1188$ (confined space)	19			
	$\zeta = 0.077$ (cluttered space)	17			

**Figure 17.** Example showing the visual difference between 2D cluttered and confined spaces.

The results of the test performed in the three mentioned environments are shown in Tables 2 and 3 and Figures 18–23. Figures 18–23, however, only show the results obtained in confined spaces and not those obtained in cluttered spaces (which have a lesser obstacle density and are considered simpler). Table 2 provides a comparative assessment of the computational time and corridor volume generated by the SFC^+ algorithm in contrast to the SCM algorithm across diverse map scenarios. The findings demonstrate the superior computational efficiency of the SFC^+ algorithm, resulting in up to a 2.5-fold reduction in processing time. This result is particularly evident in collapsed buildings and cave scenarios. The color grading in Table 2, ranging from light red to regular red, highlights a result that the computational time performance of the SFC^+ approach improves as the confined space becomes denser. It is important to note, however, that in the mine corridor scenario, the computational time difference remains relatively constant across various terrain complexities/densities. These results showcase the effectiveness of the

SFC^+ approach for rapid corridor generation capabilities within confined environments, positioning it as a highly promising solution for real-time UAV operations.

Table 2. Performance comparison between the proposed SFC^+ and SCM for 3D flight corridor generation based on a hand-crafted path.

	Obstacle Density (ζ)	Map Resolution (Voxel Size, cm)	Computational Time (t_c , s)		Corridor Volume (m^3)		No. of Generated Polyhedra	
			Proposed SFC^+	SCM	Proposed SFC^+	SCM	Proposed SFC^+	SCM
Collapsed building map scenario	$\zeta = 0.1645$	$20 \times 20 \times 20$	2.5818	3.5284	237.6235	500.5325	21	27
		$10 \times 10 \times 10$	2.6651	3.712	237.6235	500.5325	21	27
	$\zeta = 0.1233$	$20 \times 20 \times 20$	1.8453	2.5013	257.3771	515.8238	21	25
		$10 \times 10 \times 10$	1.862	2.5227	257.3771	515.8238	21	25
	$\zeta = 0.081$	$20 \times 20 \times 20$	1.5007	2.0604	277.559	530.5415	21	24
		$10 \times 10 \times 10$	1.5351	1.9972	277.559	530.5415	21	24
Cave map scenario	$\zeta = 0.18567$	$20 \times 20 \times 20$	1.3274	2.8754	221.0281	306.3768	21	33
		$10 \times 10 \times 10$	1.3054	2.9787	221.0281	306.3768	21	33
	$\zeta = 0.13325$	$20 \times 20 \times 20$	0.90729	1.9332	265.5472	351.4842	21	28
		$10 \times 10 \times 10$	0.87448	1.9204	265.5472	351.4842	21	28
	$\zeta = 0.08125$	$20 \times 20 \times 20$	0.62353	1.0092	257.136	365.0643	21	27
		$10 \times 10 \times 10$	0.5778	1.0391	257.136	365.0643	21	27
Obstructed mining map scenario	$\zeta = 0.1455$	$20 \times 20 \times 20$	1.196	2.0492	126.6911	211.6088	24	18
		$10 \times 10 \times 10$	1.1476	2.0704	126.6911	211.6088	24	18
	$\zeta = 0.1188$	$20 \times 20 \times 20$	1.1009	1.9819	126.6911	206.3995	24	19
		$10 \times 10 \times 10$	1.0287	1.9481	126.6911	206.3995	24	19
	$\zeta = 0.077$	$20 \times 20 \times 20$	0.71836	1.5541	138.8408	240.288	24	19
		$10 \times 10 \times 10$	0.70418	1.6772	138.8408	240.288	24	19

Figures 18–20 provide the geometrical shape of the 3D corridor generated for the environments tested. From such results (figures), one can compute diverse aspects of the corridor including its volume, geometrical features, convexity, etc. These depictions offer valuable insights into the structural characteristics of the flight corridors. The figures vividly demonstrate the SFC^+ algorithm's capability to produce smooth and collision-free corridors. Figure 21 presents the relationship between the computational time and the generated corridor volume against multiple environmental obstacle's densities. From Figure 21, Figure 22 can be generated. Figure 22 shows that there is a gradual increase regarding the difference in computational time curve values between the two algorithms depicting a high computational efficiency favoring the SFC^+ algorithm in most cases. Additionally, Table 2 shows that there is a substantial disparity in corridor volume between the algorithms, which is also shown in Figure 22. The SFC^+ algorithm generates corridors with reduced volume, optimizing the search area for path-planning algorithms and enhancing computational performance. Notably, the number of generated polyhedra differs between the algorithms, and a lower count corresponds to improved computational time performance, as indicated in Table 2. The proposed algorithm's polyhedra count is contingent upon the number of hand-crafted path segments, which can be fine-tuned to optimize performance in conjunction with other parameters.

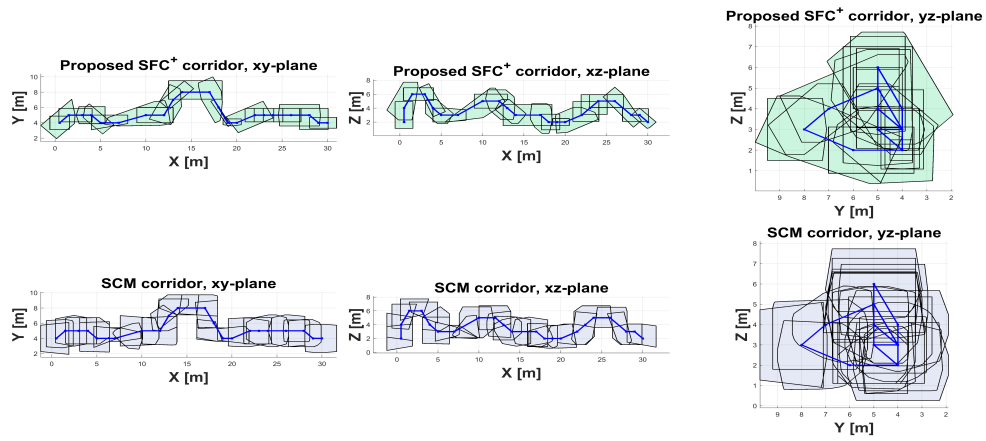
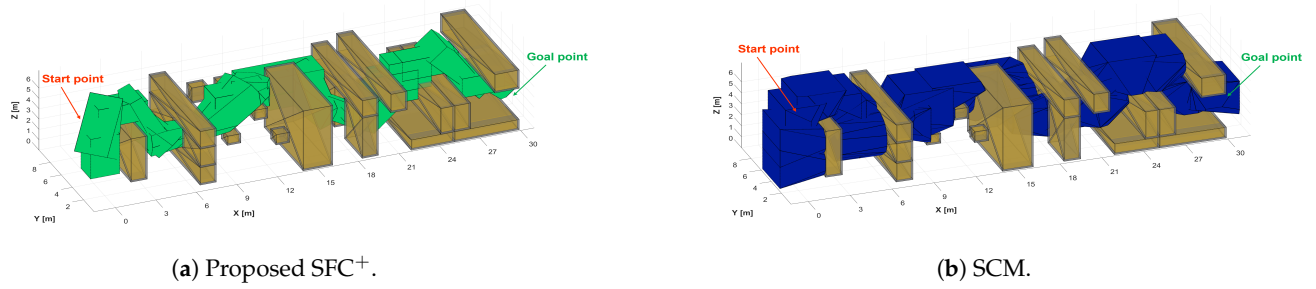


Figure 18. Results from the 3D collapsed building environment ($\zeta = 0.1645$).

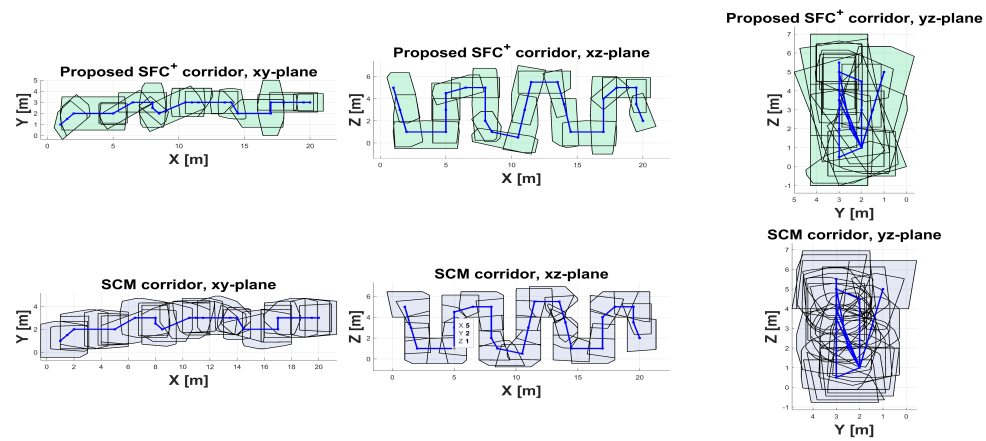
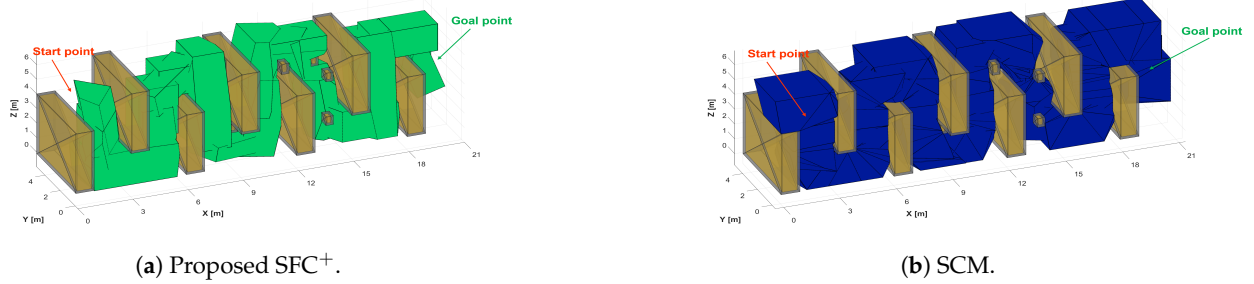


Figure 19. Results from the 3D cave environment ($\zeta = 0.18567$).

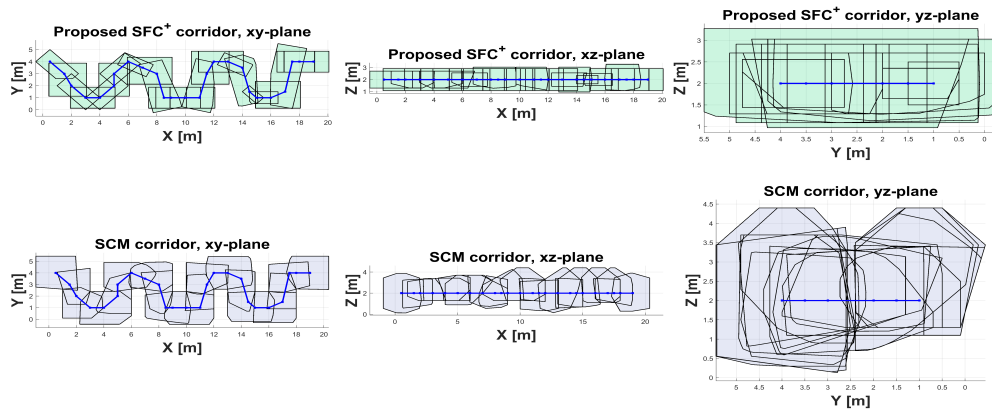
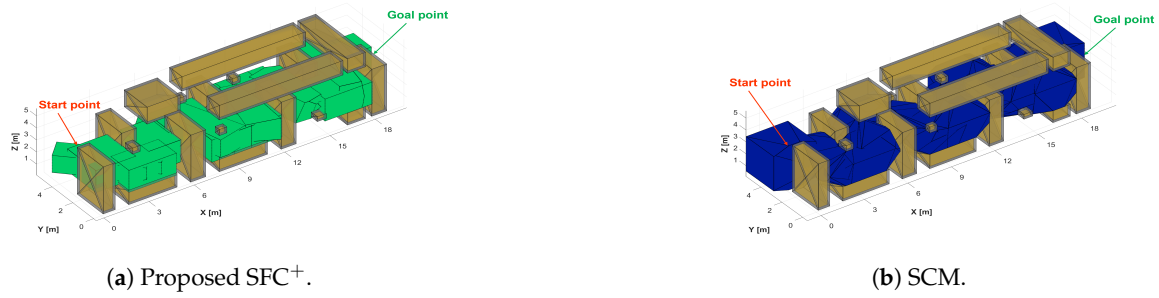
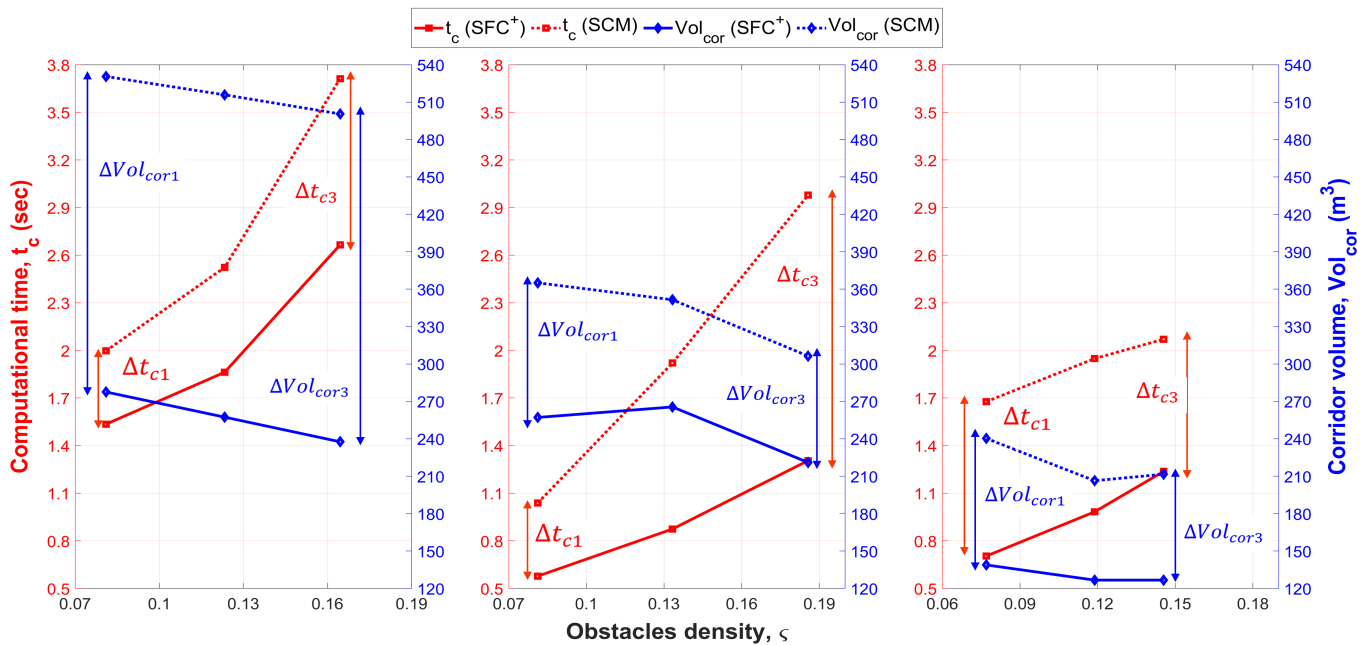


Figure 20. Results from the 3D obstructed mine corridor environment ($\zeta = 0.1455$).



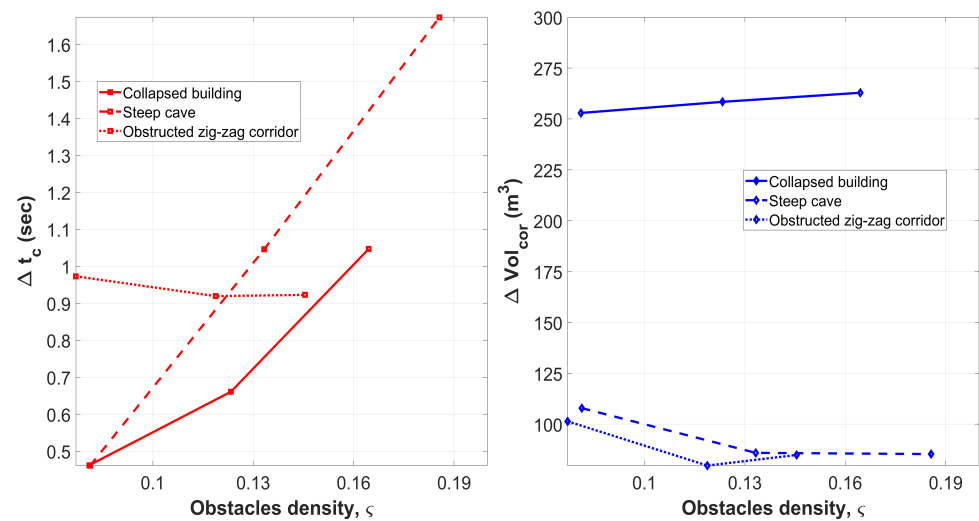


Figure 22. Performance comparison (Δt_c , ΔVol_{cor}) for three different map scenarios against obstacle density (ζ): **left:** computational time difference, **right:** corridor volume difference.

Table 3 presents a similar comparison of parameters with the distinction that the generated corridors are based on a post-processed hand-crafted path where such a post-processing step reduces the number of utilized path segments by eliminating the potentially unnecessary redundant waypoints. The results from such new hand-crafted paths indicate a reduced number of generated polyhedra for the proposed SFC^+ algorithm. Figure 23 shows an example of generating a flight corridor within a given environment based on a post-processed hand-crafted path. This result aligns with expectations leading to significantly improved computational time with up to threefold faster performance. However, the findings underscore the trade-off between computational performance and corridor characteristics. While the SFC^+ algorithm demonstrates superior computational efficiency, it is important to consider the impact on the shape and volume of the generated corridors. Therefore, a thoughtful optimization process for the post-processing algorithm is crucial to ensure an optimal balance between these parameters.

Table 3. Performance comparison between the proposed SFC^+ and SCM for 3D flight corridor generation based on a post-processed hand-crafted path.

	Obstacle Density (ζ)	Map Resolution (Voxel Size, cm)	Computational Time (t_c , s)		Corridor Volume (m^3)		No. of Generated Polyhedra	
			Proposed SFC^+	SCM	Proposed SFC^+	SCM	Proposed SFC^+	SCM
Collapsed building map scenario	$\zeta = 0.1645$	$20 \times 20 \times 20$	2.1929	2.7444	231.7023	358.8962	7	24
		$10 \times 10 \times 10$	1.9182	2.9326	309.0498	423.968	8	23
Cave map scenario	$\zeta = 0.18567$	$20 \times 20 \times 20$	1.0666	2.7866	370.3862	307.041	13	32
		$10 \times 10 \times 10$	1.0577	2.6848	362.4562	288.697	13	30
Obstructed mining environment	$\zeta = 0.1455$	$20 \times 20 \times 20$	0.66375	1.806	113.088	177.5656	7	18
		$10 \times 10 \times 10$	0.70554	1.6912	93.0473	172.254	6	19

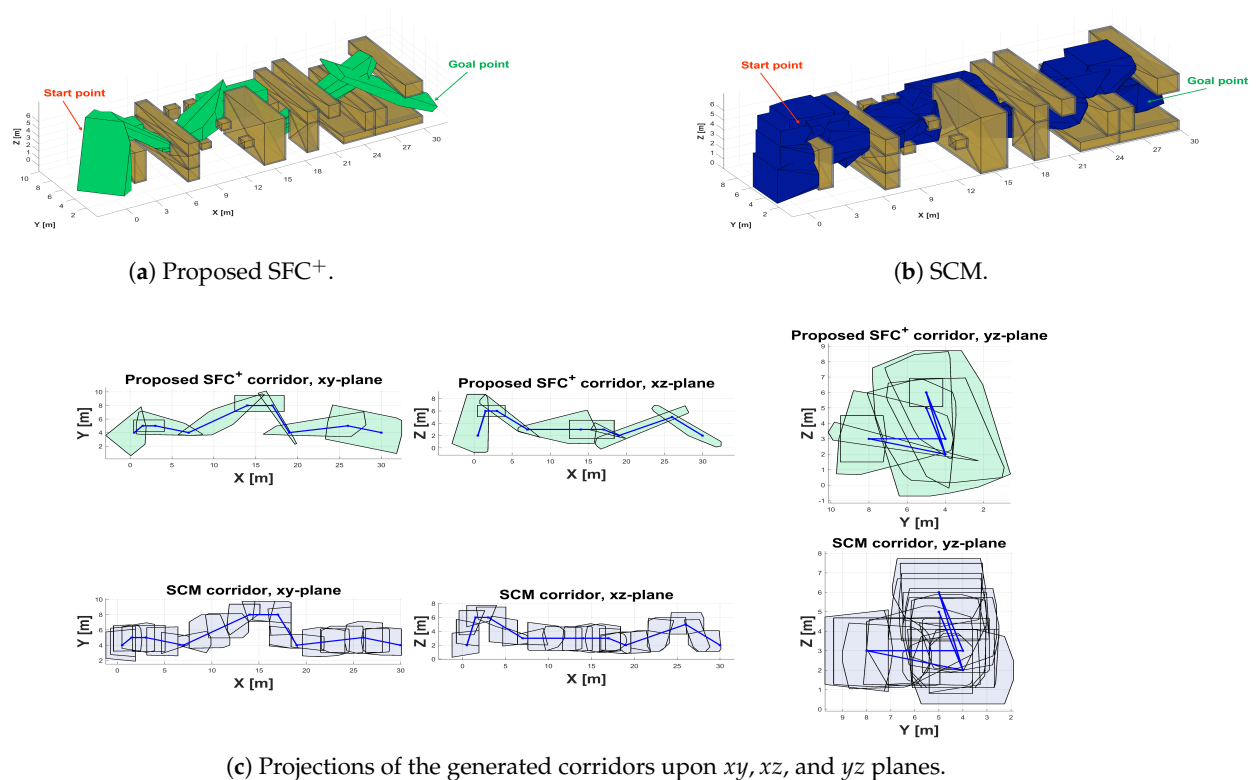


Figure 23. Results from a 3D collapsed building environment ($\zeta = 0.1645$) based on a post-processed hand-crafted path.

In summary, the results illustrate the capabilities of the algorithm to generate smooth and collision-free flight corridors with respect to a hand-crafted path, showcasing its potential effectiveness in being used by robotic systems to maneuver within confined environments. The proposed methodology adeptly accommodates the safe coexistence of multiple UAVs within a shared constrained environment by comprehensively defining the constraints that pertain to the hand-crafted path and environmental mapping, as well as the operational constraints and capabilities specific to each UAV. Furthermore, this approach exhibits the flexibility to adapt seamlessly to diverse types of UAVs deployed within a range of confined spaces. These findings collectively emphasize the significance of the *SFC*⁺ algorithm as a reliable solution for real-time UAV operations in challenging, diverse confined scenarios characterized by dense obstacles and geometric complexities. Nonetheless, the application of this methodology in real-world scenarios necessitates careful attention to a multitude of concerns and constraints related to the SLAM process, which is instrumental in mapping uncharted terrains via the vehicle's onboard sensors. These limitations stem from various sources, encompassing sensor uncertainties, altitude-related inaccuracies, computational resource constraints, and potential hardware malfunctions.

5. Conclusions

The proposed *SFC*⁺ algorithm is an effective solution for generating 3D flight corridors within confined environments. The results and comparative analysis demonstrate a significant improvement when compared to previously developed mechanisms in terms of computational efficiency, corridor volume, and polyhedra count. The results show the proposed algorithm consistently outperforms the previously developed state-of-the-art algorithms, exhibiting up to a 2.5-fold improvement in computational time, particularly when used in geometrically complex environments, such as those encountered inside collapsed buildings. The visual depictions of the generated corridors showcase the algorithm's ability to generate corridors suitable for robot navigation. The analysis of the

generated corridors, along with the trade-off between computational performance and corridor shape/volume, highlighted the importance of fine-tuning the post-processing algorithm to strike an optimal balance. Furthermore, the algorithm's capability to adapt to various map resolutions and obstacle densities enhances its applicability in real-world scenarios.

In conclusion, the SFC^+ algorithm presents a valuable contribution to the field of UAV operations in confined environments. Its computational efficiency, generation of collision-free flight corridors, and adaptability to diverse scenarios position it as a promising solution for enabling safe and efficient UAV navigation in complex confined spaces.

Author Contributions: Conceptualization, S.M. and A.R.-S.; methodology, S.M.; software, S.M.; validation, S.M. and A.R.-S.; formal analysis, S.M. and A.R.-S.; investigation, S.M. and A.R.-S.; resources, S.M. and A.R.-S.; writing—original draft preparation, S.M.; writing—review and editing, S.M. and A.R.-S.; visualization, S.M.; supervision, A.R.-S.; project administration, A.R.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is unavailable due to privacy.

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

Abbreviations

The following abbreviations are used in this manuscript:

CP	Convex Polyhedron
IRIS	Iterative Regional Inflation by Semi-Definite Programming
I-TRP	Improved “Teach-Repeat-Replan”
SCM	Star Convex Method
SFC^+	Enhanced Safe Flight Corridor
SLAM	Simultaneous Localization and Mapping
UAV	Unmanned Aerial Vehicle
USAR	Urban Search and Rescue
VTOL	Vertical Take-off and Landing

References

- Shahmoradi, J.; Talebi, E.; Roghanchi, P.; Hassanalian, M. A comprehensive review of applications of drone technology in the mining industry. *Drones* **2020**, *4*, 34.
- Ren, H.; Zhao, Y.; Xiao, W.; Hu, Z. A review of UAV monitoring in mining areas: Current status and future perspectives. *Int. J. Coal Sci. Technol.* **2019**, *6*, 320–333.
- Półka, M.; Ptak, S.; Kuziora, Ł. The use of UAV's for search and rescue operations. *Procedia Eng.* **2017**, *192*, 748–752.
- Neubauer, M.; Günther, G.; Füllhas, K. *Structural Design Aspects and Criteria for Military UAV*; Technical Report; European Aeronautic Defence and Space (EADS): Munich, Germany, 2007.
- Gupta, S.G.; Ghonge, D.M.; Jawandhiya, P.M. Review of unmanned aircraft system (UAS). *Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET)* **2013**, *2*, 1646–1658.
- Innocente, M.S.; Grasso, P. Self-organising swarms of firefighting drones: Harnessing the power of collective intelligence in decentralised multi-robot systems. *J. Comput. Sci.* **2019**, *34*, 80–101.
- CSIRO Head Office; Westwick-Farrow Pty Ltd. Researchers Deploy Autonomous Drone to Improve Operations for Mining Industry. November 2017. Available online: <https://www.processonline.com.au/content/business/news/researchers-deploy-autonomous-drone-to-improve-operations-for-mining-industry-1267356273> (accessed on 24 September 2023); .
- Coldewey, D. Subterranean Drone Mapping Startup Emesent Raises \$2.5 M to Autonomously Delve the Deep. TechCrunch; November 2018. Available online: <https://techcrunch.com/2018/11/05/subterranean-drone-mapping-startup-emesent-raises-2-5m-to-autonomously-delve-the-deep/> (accessed on 24 September 2023).
- Reporter, S. Mapping the Underground with Drones, Legged Robots. Aspermont Media Ltd. October 2018. Available online: <https://www.miningmagazine.com/innovation/news/1347934/mapping-the-underground-with-drones-legged-robots> (accessed on 24 September 2023).
- Liang, H.; Lee, S.C.; Bae, W.; Kim, J.; Seo, S. Towards UAVs in Construction: Advancements, Challenges, and Future Directions for Monitoring and Inspection. *Drones* **2023**, *7*, 202.

11. Liu, S.; Watterson, M.; Mohta, K.; Sun, K.; Bhattacharya, S.; Taylor, C.J.; Kumar, V. Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1688–1695.
12. Zinage, V.; Arul, S.H.; Manocha, D.; Ghosh, S. 3D-Online Generalized Sensed Shape Expansion: A Probabilistically Complete Motion Planner in Obstacle-Cluttered Unknown Environments. *IEEE Robot. Autom. Lett.* **2023**, *8*, 3334–3341.
13. Chen, J.; Liu, T.; Shen, S. Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1476–1483.
14. Gao, F.; Wang, L.; Zhou, B.; Zhou, X.; Pan, J.; Shen, S. Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments. *IEEE Trans. Robot.* **2020**, *36*, 1526–1545.
15. Gao, F.; Wang, L.; Wang, K.; Wu, W.; Zhou, B.; Han, L.; Shen, S. Optimal trajectory generation for quadrotor teach-and-repeat. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1493–1500.
16. Lien, J.M.; Amato, N.M. Approximate convex decomposition of polygons. In Proceedings of the Twentieth Annual Symposium on Computational Geometry, Brooklyn, NY, USA, 9–11 June 2004; pp. 17–26.
17. Liu, H.; Liu, W.; Latecki, L.J. Convex shape decomposition. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 97–104.
18. Eidenbenz, S.J.; Widmayer, P. An approximation algorithm for minimum convex cover with logarithmic performance guarantee. *SIAM J. Comput.* **2003**, *32*, 654–670.
19. Feng, H.Y.; Pavlidis, T. Decomposition of polygons into simpler components: Feature generation for syntactic pattern recognition. *IEEE Trans. Comput.* **1975**, *100*, 636–650.
20. Deits, R.; Tedrake, R. Computing large convex regions of obstacle-free space through semidefinite programming. In Proceedings of the Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics, Istanbul, Turkey, 3–5 August 2015; pp. 109–124.
21. Deits, R.; Tedrake, R. Efficient mixed-integer planning for UAVs in cluttered environments. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Washington, DC, USA, 26–30 May 2015; pp. 42–49.
22. Tan, L.; Hu, M.; Lin, H. Agent-based simulation of building evacuation: Combining human behavior with predictable spatial accessibility in a fire emergency. *Inf. Sci.* **2015**, *295*, 53–66.
23. Mamou, K.; Ghorbel, F. A simple and efficient approach for 3D mesh approximate convex decomposition. In Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; pp. 3501–3504.
24. Sarmientoy, A.; Murrieta-Cidz, R.; Hutchinson, S. A sample-based convex cover for rapidly finding an object in a 3-D environment. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 3486–3491.
25. Savin, S. An algorithm for generating convex obstacle-free regions based on stereographic projection. In Proceedings of the 2017 International Siberian Conference on Control and Communications (SIBCON), Astana, Kazakhstan, 29–30 June 2017; pp. 1–6.
26. Zhong, X.; Wu, Y.; Wang, D.; Wang, Q.; Xu, C.; Gao, F. Generating large convex polytopes directly on point clouds. *arXiv* **2020**, arxiv:2010.08744.
27. Jansen, F.; Ramirez-Serrano, A. Extended MPC Strategy for Maneuvering Unmanned Vehicles in Restricted 3D Environments. In Proceedings of the Unmanned Systems Canada Conference, Montreal, QC, Canada, 12 November 2011.
28. Hosseini, Z.; Ramirez-Serrano, A.; Martinuzzi, R.J. Ground/wall effects on a tilting ducted fan. *Int. J. Micro Air Veh.* **2011**, *3*, 119–141.
29. Jansen, F. Manoeuvring Unmanned Vehicles through Confined 3D Environments Using Model Predictive Control. Master's Thesis, University of Calgary, Calgary, AB, Canada, 2011.
30. Lay, S.R. *Convex Sets and Their Applications*; Courier Corporation: Chelmsford, MA, USA, 2007.
31. Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
32. Cherney, M. Drones Are Now Operating Underground: Mining Companies Look to Automation to Help Companies Dig Out More ore and Save Lives. Kathryn Report Company. November 2017. Available online: <http://www.kathrynsreport.com/2017/11/drones-are-now-operating-underground.html> (accessed on 24 September 2023).
33. Hannah Beech, R.C.P.; Suhartono, M. Still Can't Believe It Worked': The Story of the Thailand Cave Rescue. *The New York Times*, 12 July 2018.

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