

## Research Article

# Filtered Medial Surface Based Approach for 3D Collision-Free Path Planning Problem

**Karima Benzaid** <sup>1</sup>, **Romain Marie**,<sup>2</sup> **Noura Mansouri**,<sup>1</sup> and **Ouidad Labbani-Igbida**<sup>2</sup>

<sup>1</sup>LARC Laboratory, University Frères Mentouri Constantine 1, Constantine, Algeria

<sup>2</sup>X-Lim Institute, UMR CNRS 7252, University of Limoges, Limoges, France

Correspondence should be addressed to Karima Benzaid; kari\_benzaid@yahoo.fr

Received 25 September 2017; Revised 14 February 2018; Accepted 1 March 2018; Published 3 June 2018

Academic Editor: Farrokh Janabi-Sharifi

Copyright © 2018 Karima Benzaid et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper introduces an original 3D path planning approach for Unmanned Aerial Vehicle (UAV) applications. More specifically, the core idea is to generate a smooth and collision-free path with respect to the vehicle dimension. Given a 3D grid representation of the environment, the Generalized Voronoi Graph (GVG) is first approximated using a filtered medial surface (FMS) algorithm on the corresponding navigable space. Based on an efficient pruning criterion, the produced FMS excludes GVG portions corresponding to narrow passages unfitting safe UAV navigation constraints, and thus it defines a set of guaranteed safe trajectories within the environment. Given a set of starting and destination coordinates, an adapted A-star algorithm is then applied to compute the shortest path on the FMS. Finally, an optimization process ensures the smoothness of the final path by fitting a set of 3D Bézier curves to the initial path. For a comparative study, the A-star algorithm is applied directly on the input environment representation and relevant comparative criteria are defined to assert the proposed approach using simulation results.

## 1. Introduction

This paper deals with the issue of 3D path planning for UAVs in a known environment with stationary obstacles. The problem is to define an efficient method for finding the safest path from a starting point to a target, taking into account the dimension of the UAV.

To deal with this problem, many methods are proposed in the literature. Sampling based algorithms [1], graph based search algorithms [2, 3], mathematic model based algorithms [4, 5], and bioinspired algorithms [6–8] are the fundamental families of path planning algorithms [9]. Most of the path planning solutions were originally developed for 2D problems and then extended to the third dimension, thus increasing the complexity of the approach and its computational cost.

Among the most known sampling based algorithms, we can cite the Rapidly Exploring Random Tree (RRT) [10] and the Probabilistic Road-Maps (PRM) [11] algorithms. These methods are initially designed for 2D path planning problems based on the sampling of the configuration space.

The drawback of this kind of algorithms is that they return nonoptimal solutions [12]. To improve the quality of the solution, many variants of these sampling based algorithms have been proposed [12–16]. For aerial vehicles path planning problem, some adapted versions [17–20] are applied. Few methods based on Voronoi diagram have been used to deal with the problem of 3D path planning [21].

The A-star is the most popular graph based search algorithm. Developed by Hart et al. [22], it is based on Dijkstra's algorithm [23] and has been widely implemented in robotics and video games. The conventional A-star is used to find the shortest (possible) path connecting edges of 8 neighbors in 2D grids and 26 neighbors in 3D grids. However, this algorithm cannot find the path in any neighbor connection angle. This implies a suboptimal solution represented as a path with continuous heading changes. With the aim of overcoming this limit and/or improving results in terms of time computation, memory requirement, and path length, some variants of the A-star algorithm have been developed [24–26]. For 3D applications, we can cite [2, 3, 27, 28]. Even though the solution was improved, the planned path is not

unruffled enough and needs more refinements to make it smooth for aerial vehicles tracking.

In this paper, we propose a novel approach based on the skeletonization of the navigable space along with an adapted A-star path planner for aerial vehicles. The environment is represented as a discrete 3D occupancy grid supposedly given (its construction is out of scope here). Based on a pruning parameter, the skeletonization algorithm is able to discard noisy skeleton branches and thus exclude narrow passages with respect to the UAV dimensions. This constraint is of major importance but is rarely considered in the literature.

The key idea is to extract the *filtered medial surface* of the grid representation according to the dimension of the aerial vehicle. In the next step, the use of the A-star algorithm aims to find the shortest path on the FMS. Since the returned path is not smooth due to the continuous heading changes of the A-star path, a smoothing method is finally applied.

This may appear similar to the idea adopted in [29]. However, in this cited paper, the authors propose a 2D solution for a 3D path planning problem. The method is based on a 2D skeleton represented as a thin line. The 3rd dimension is not considered at any time.

The main contributions can be summarized in three essential points:

- (i) We introduce a 3D skeletonization algorithm for direct applications in aerial robotics context based on a 2D shape representation algorithm.
- (ii) In order to eliminate oscillations and tight turns in the generated path, we propose a new safety corridor based smoothing approach using Bézier curves.
- (iii) We perform a comparative study of the proposed approach by applying the A-star algorithm directly on an inflated occupancy grid and defining a set of quantitative criteria.

The paper is organized as follows. In Section 2, we present the two steps constituting the proposed solution. The first is the extraction of the FMS on the navigable space, and the second is the computation of the shortest path on the FMS. In Section 3, we introduce a new Bézier curve based method to smooth the initial planned path. In Section 4, our solution is compared to the A-star solution applied on the 3D grid representation. We finally conclude the paper and introduce the future work.

## 2. Path Planning on the Environments GVG

This section is dedicated to the problem of accurate and real-time path planning on the Generalized Voronoi Graph (GVG), i.e., the set of points in the environment at equal distance from their two closest obstacles. Given a 3D occupancy grid  $\mathcal{F} \cup \mathcal{O}$  delimiting the navigable space  $\mathcal{F}$  from the obstacles  $\mathcal{O}$ , it is approximated here by the medial surface of  $\mathcal{F}$ , filtered with respect to the robot dimension.

*2.1. Skeletonization and Filtering of the Navigable Space.* Let  $\mathcal{X}_{\mathbb{R}}$  be a subset of  $\mathbb{R}^n$  and  $\mathcal{X}_{\mathbb{R}}^c$  be its complement. Initially defined by Blum [30], the real medial axis  $S_{\mathbb{R}}$  of  $\mathcal{X}_{\mathbb{R}}$  (also

referred to as skeleton) is the set of points  $\mathbf{x} \in \mathcal{X}_{\mathbb{R}}$  at equal distance from their two closest neighbors in  $\mathcal{X}_{\mathbb{R}}^c$ . As pointed out in [31], this definition is conceptually equivalent to the GVG of an environment and is thus used here as an accurate approximation. In the specific case of 3D shapes, it generates a set of surfaces and is thus also referred to as medial surface.

For each shape point  $\mathbf{x} \in \mathcal{X}_{\mathbb{R}}$ , let  $\mathcal{P}_{\mathbf{x}} = \operatorname{argmin}_{\mathbf{y} \in \mathcal{X}_{\mathbb{R}}^c} d(\mathbf{x}, \mathbf{y})$  denote the set of its closest points on  $\mathcal{X}_{\mathbb{R}}^c$ , referred to as projection of  $\mathbf{x}$ . The corresponding Euclidean distance is denoted as  $\mathcal{D}_{\mathbf{x}} = d(\mathbf{x}, \mathcal{P}_{\mathbf{x}})$ . Accordingly, the skeleton  $S_{\mathbb{R}}(\mathcal{X}_{\mathbb{R}})$  can then be defined by

$$S_{\mathbb{R}}(\mathcal{X}_{\mathbb{R}}) = \{\mathbf{x} \in \mathcal{X}_{\mathbb{R}}, |\mathcal{P}_{\mathbf{x}}| \geq 2\}. \quad (1)$$

To efficiently approximate and filter the environment GVG, two important concerns have to be considered. First, the navigable space is expressed in this work in a discrete domain (the occupancy grid  $\mathcal{F} \cup \mathcal{O}$ ). Applying (1) directly on  $\mathcal{F}$  would fail, since, in most cases,  $|\mathcal{P}_{\mathbf{x}}| = 1$  even when a given cell  $\mathbf{x}$  actually belongs to the environment GVG.

Therefore, the definition of the discrete medial axis has to be slightly modified. Although several solutions were proposed over the years, this work is based upon the Integer Medial Axis (IMA) [32]. Let  $\mathcal{P}_{\mathbf{x}}^*$  denote the reduced projection of  $\mathbf{x}$ , corresponding to the first element in  $\mathcal{P}_{\mathbf{x}}$  with respect to a lexical ordering. Given a digital shape  $\mathcal{X}_{\mathbb{Z}} \subset \mathbb{Z}^n$  and its complement  $\mathcal{X}_{\mathbb{Z}}^c$ , the discrete skeleton  $S_{\mathbb{Z}}$  of  $\mathcal{X}_{\mathbb{Z}}$  can then be expressed as

$$S_{\mathbb{Z}}(\mathcal{X}_{\mathbb{Z}}) = \{\mathbf{x} \in \mathcal{X}_{\mathbb{Z}}, \exists \mathbf{y} \in N_{\mathbf{x}} \mid d(\mathcal{P}_{\mathbf{x}}^*, \mathcal{P}_{\mathbf{y}}^*) > 1, d(\mathbf{m}, \mathcal{P}_{\mathbf{x}}^*) \geq d(\mathbf{m}, \mathcal{P}_{\mathbf{y}}^*)\}, \quad (2)$$

where  $N_{\mathbf{x}} = \{\mathbf{y} \in \mathcal{X}_{\mathbb{Z}}, d(\mathbf{x}, \mathbf{y}) = 1\}$  is the set of direct neighbors of  $\mathbf{x}$  and  $\mathbf{m}$  is the midpoint of line segment  $[\mathbf{x}, \mathbf{y}]$ . The first condition measures the (Euclidean) distance between the projections of  $\mathbf{x}$  and  $\mathbf{y}$  to remove skeleton points related to aliasing, while the second condition is used to produce a skeleton as thin as possible, since it only selects the best approximation to the real skeleton. Note that it is theoretically proven [32] that this definition efficiently approximates the medial axis of the corresponding shape in  $\mathbb{R}^n$  while removing skeleton points induced by discretization.

The second major concern is the question of boundary noise: Skeletons are known to be very sensitive to small shape deformations and tend to produce spurious branches. The IMA efficiently approximates Blum's real medial axis but is not designed to filter noisy branches. This issue is addressed here using the delta medial axis (DMA) [33], a real-time pruning strategy able to identify and discard spurious skeleton branches in real time. The algorithm was initially designed for 2D shapes but can straightforwardly be extended to higher dimensions.

The core concepts of the algorithm are illustrated in Figure 1.

Given  $\mathbf{x} \in \mathcal{X}_{\mathbb{Z}}$ ,  $\mathbf{y} \in N_{\mathbf{x}}$ , let  $r_{\mathbf{x}, \mathbf{y}}$  denote the radius of the largest disk disjoint from  $\mathcal{X}_{\mathbb{Z}}^c$  and centered along  $[\mathcal{P}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}]$ :

$$r_{\mathbf{x}, \mathbf{y}} = \sup \{\mathcal{D}_{\mathbf{z}} \mid \mathbf{z} \in [\mathcal{P}_{\mathbf{x}}, \mathcal{P}_{\mathbf{y}}]\}. \quad (3)$$

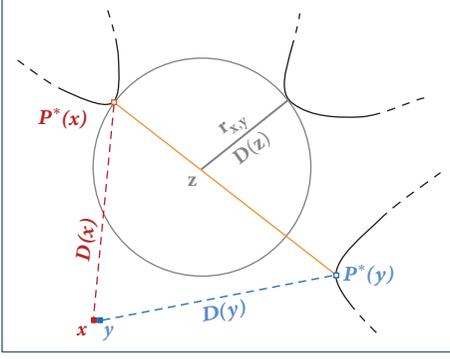


FIGURE 1: Illustrative representation of the concepts involved in DMA computation.

The pruning criterion  $\delta^*(\mathbf{x})$  is then defined as the maximal value of  $r_{\mathbf{x},\mathbf{y}}$  among its direct neighbors  $\mathbf{y} \in N_{\mathbf{x}}$ :

$$\delta^*(\mathbf{x}) = \sup \{r_{\mathbf{x},\mathbf{y}} \mid \mathbf{y} \in N_{\mathbf{x}}, d(\mathbf{m}, \mathcal{P}_{\mathbf{x}}) \geq d(\mathbf{m}, \mathcal{P}_{\mathbf{y}})\}. \quad (4)$$

Note that the second condition is similar to the IMA. It contributes to generating a thin skeleton, and thus a better approximation to the GVG. The DMA is then defined as follows.

*Definition 1* (delta medial axis). Let  $\delta \in \mathbb{R}_+^*$  and  $\mathcal{X}_{\mathbb{Z}} \subset \mathbb{Z}^n$ . The delta medial axis  $MA_{\delta}(\mathcal{X}_{\mathbb{Z}})$  of  $\mathcal{X}_{\mathbb{Z}}$  is the set of points  $\mathbf{x} \in \mathcal{X}_{\mathbb{Z}}$  with a maximal delta value  $\delta^*(\mathbf{x})$  of at least  $\delta$ :

$$MA_{\delta}(\mathcal{X}_{\mathbb{Z}}) = \{\mathbf{x} \in \mathcal{X}_{\mathbb{Z}} \mid \delta_M(\mathbf{x}) \geq \delta\}. \quad (5)$$

Intuitively, the DMA discards skeleton branches generated by shape deformations smaller than parameter  $\delta$ . Based on the robot dimension, it can thus be tuned to retain only skeleton branches (thus GVG navigable paths) large enough to ensure safe navigation.

Applied on the input environment representation  $\mathcal{F} \cup \mathcal{O}$ , the GVG is thus finally approximated and filtered for a given UAV size  $\delta$ , by computing the corresponding FMS  $MA_{\delta}(\mathcal{F})$ . Examples for different  $\delta$  values are depicted in Figure 2.

In order to illustrate how the FMS extracted by skeletonization depends on the UAV size, we have considered the same environment representation for two different UAV sizes. As it can be seen in Figure 2, the FMS narrows when  $\delta$  increases. This means that there are more candidate paths to reach the destination from an initial position when the UAV size is small.

**2.2. Shortest Path Computation.** The FMS defines a subspace from the navigable space that guarantees safe paths according to the dimension of the UAV. The A-star algorithm is then used to compute a 3D refined path on the FMS. It allows finding the path with the smallest cost from a set of candidate paths leading to a destination. The A-star is based on a cost function  $F(n)$ :

$$F(n) = G(n) + H(n), \quad (6)$$

where  $n$  is the current node,  $G(n)$  denotes the moving cost from the starting point (node)  $P_S$  to the current node  $n$ , and  $H(n)$  denotes the moving cost from the current node  $n$  to the target point (node)  $P_T$ .

Given the coordinates of  $P_S$  and  $P_T$ , the aim here is to compute the shortest path on the FMS. The cost function is then the Euclidean distance in the 3D space. A tolerable distance  $D_t$  has to be defined to consider two points in the FMS connected. In a 3D occupancy grid, the A-star algorithm uses searching restricted to 26-connectivity (giving the closest cells in 3D environment) or in every angle for some variants of the A-star. Here, the search forwards on the FMS based on  $D_t$ . For a current node, neighbors are determined according to their distance to the current node. They will be examined if their distance to the current node is less than or equal to  $D_t$ .

The generated path is a set of ordered waypoints defined on the FMS from  $P_S$  to  $P_T$ . We consider the two previous cases of UAV size  $\delta$  (20 cm and 40 cm). Figure 3 shows the shortest path found for each case, with the vertical position  $z$ . One can see that the planned path is suboptimized with oscillations, which requires a postprocessing step to refine it.

### 3. Smoothing the Planned Path

The generated path that the UAV is supposed to track must be elegant and unruffled without oscillations, but the path computed on the FMS is not necessarily smooth. To solve this problem, we propose to add an additional processing step based on the use of 3D Bézier curves.

**3.1. Bézier Curves.** Bézier curves are parametric curves frequently used in vehicle design, fonts representation, games, and 3D animations. Consider  $P_0, P_1, \dots, P_n$  ( $n \geq 1$ ),  $n + 1$  points. The Bézier curve of  $n$  degrees associated with these points is the curve  $C$  defined by  $M(t)$  with  $t \in [0, 1]$ :

$$M(t) = B_0(t)P_0 + B_1(t)P_1 + \dots + B_n(t)P_n \quad (7)$$

$P_i$  ( $i = 1 : n$ ) are called the control points of the Bézier curve and define the control polygon.  $B_i$  ( $0 \leq i \leq n$ ) coefficients are the Bernstein polynomials of degree  $n$  given as follows:

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}, \quad i = 0, \dots, n \quad (8)$$

(with  $C_n^i = \frac{n!}{i!(n-i)!}$ ).

In a 3D environment,  $P_i$  is the point  $(x_i, y_i, z_i)$ . This means that the point  $M(t)$  has as coordinates

$$\begin{aligned} x(t) &= \sum_{i=0}^n B_i(t) x_i, \\ y(t) &= \sum_{i=0}^n B_i(t) y_i, \\ z(t) &= \sum_{i=0}^n B_i(t) z_i. \end{aligned} \quad (9)$$

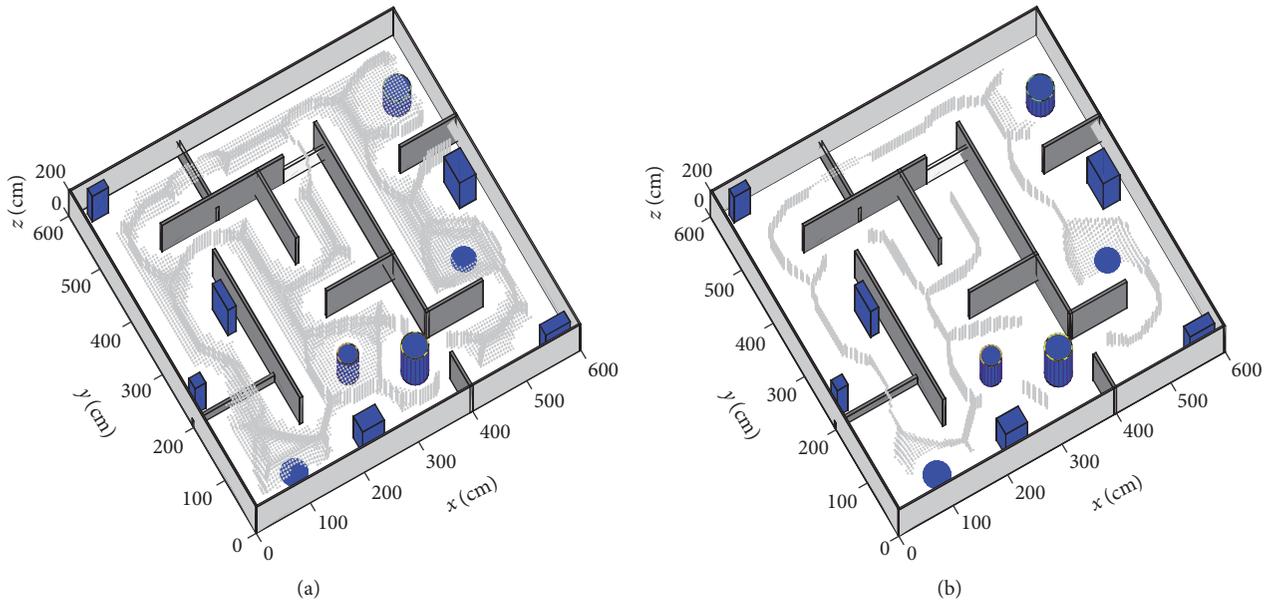


FIGURE 2: Two FMS of the environment based on different UAV sizes ((a)  $\delta = 20$  cm, (b)  $\delta = 40$  cm).

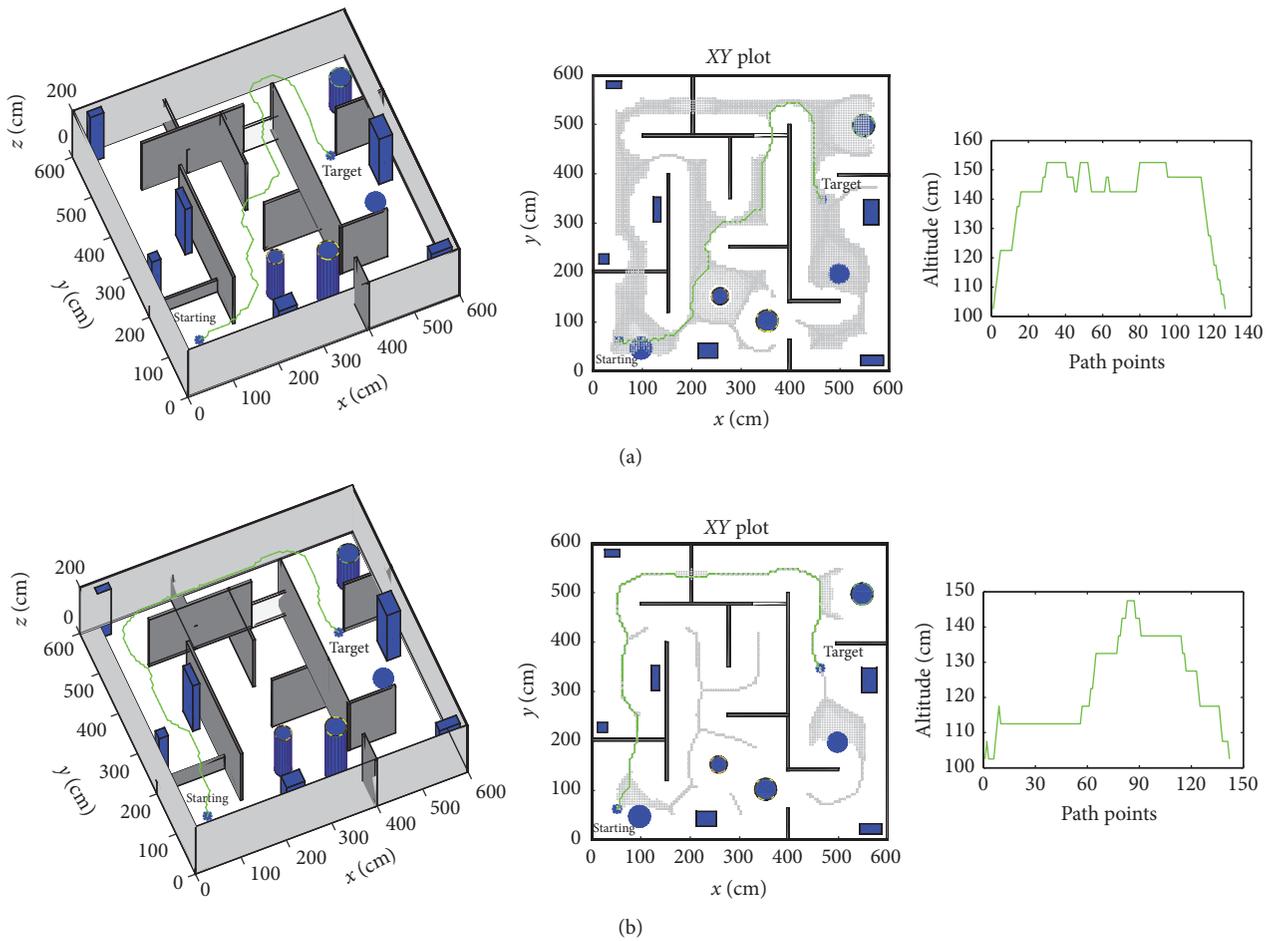


FIGURE 3: Shortest path computed on the FMS based on different UAV sizes (20 cm (a), 40 cm (b)) with the altitude (right column) plotted for each case.

Because they are elegant and easy to use and can describe any shape, Bézier curves are appropriate to solve the smoothing problem of a set of ordered data. When it is about a path between two specific points  $P_S$  and  $P_T$ , one more reason to use Bézier curves is their *endpoint interpolation property*. This means that the first and last control points (denoted, resp., by  $P_S$  and  $P_T$ ) are the endpoints of the curve. To form a complex smooth path, cubic Bézier curves can be smoothly connected thanks to the *endpoint tangent property*. This property means that the cubic Bézier curve  $C$  is tangent to  $[P_0, P_1]$  ( $[P_2, P_3]$ ) in  $P_0$  ( $P_3$ ) and  $\dot{M}(0) = 3(P_1 - P_0)$  ( $\dot{M}(1) = 3(P_3 - P_2)$ ). We will give more details on this point thereafter.

**3.2. Bézier Curves Smoothing Strategy.** The issue of smoothing of the obtained path is formulated as an optimization problem with constraints. The objective is to compute an approximated path to the real one which allows the UAV to make the least maneuvers. Note that there is a compromise between the feasibility of the path and its optimality. By smoothing, the path optimality in terms of safety and path length may degrade. The idea we propose here is to define a safety corridor around the original path as follows:

$$(Q_j - p(t_j)) \leq (D_j - \delta), \quad (10)$$

where  $Q_j$  are the original data,  $p(t_j)$  are values of approximated data computed by interpolation, and  $D_j$  is the Euclidean distance to the nearest obstacle in each original path point.

The smoothed path has to be in the safety corridor. For that, we have to determine (by solving the optimization problem) the control points defining the shape of the smoothed path formed by connected patches of Bézier curves. The endpoint interpolation property of Bézier curves ensures that the path starts in  $P_S$  and ends in  $P_T$ . The objective function to be minimized is the mean of the squared distance between real data and smoothed data. Suppose that there are  $m$  waypoints  $Q_j$  ( $j = 1 : m$ ) in the original data. For smoothed data  $p(t_j)$ , we note that  $0 \leq t_j \leq 1$  with uniform division, so  $Q_1 = p(0) = P_S$  and  $Q_m = p(1) = P_T$ . The objective function is given by

$$\min_{p(t_j)} \left( \frac{1}{m-2} \sqrt{\sum_{j=2}^{m-1} (Q_j - p(t_j))^2} \right). \quad (11)$$

The original data can be divided into several segments by giving the initial set of break points. Each segment will be approximated by a cubic Bézier curve (patch) by identifying the middle control points. If there are  $l$  segments (i.e.,  $l - 1$  break points), by solving the problem of optimization (see Figure 4),  $3l - 1$  control points must be determined between  $P_S$  and  $P_T$  to define  $l$  Bézier curves. It is important to ensure that the control points joining Bézier curves are not fixed and must be determined as well.

The  $C^1$ -continuity between patches of Bézier curve can be translated to constraints of the optimization. Let  $P_0^k, P_1^k, P_2^k$ , and  $P_3^k$  denote the control points of the  $k$ th Bézier curve.  $k$ th

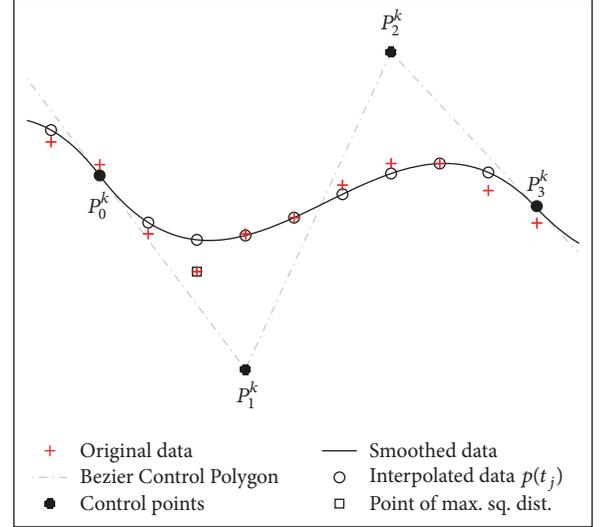


FIGURE 4: Smoothing strategy illustration for the  $k$ th segment.

and  $(k + 1)$ th cubic Bézier curves are  $C^1$  continuous at the joining point if

$$\begin{aligned} P_3^k &= P_0^{k+1}, \\ P_3^k - P_2^k &= P_1^{k+1} - P_0^{k+1}. \end{aligned} \quad (12)$$

We note that it is possible to add the constraints of UAV heading in the starting and target points, basing on the endpoint tangent property. But this is not considered in this work.

In the optimization process, we consider the case where no solution is found while respecting the constraint that the smoothed path must not be outside the safety corridor. In this case, we split the segment which leaves the safety corridor in two segments and we add their joining point to the break points list. This gives one more segment (consequently, one more Bézier curve to be defined) by replacing one segment with two new segments. We try again to find a solution of the optimization problem defining an approximated path in the safety corridor. The split step is repeated until this is done. Figure 5 shows the refined path after using the Bézier curves based approach. We can see that the path planned by the proposed method is optimal in terms of safety and smoothness.

Table 1 illustrates the results of the gap between original and smoothed paths. Intuitively, this gap can reach its maximum value where the safety corridor width is the maximum. We note that the smoothing method satisfies the condition on the maximum permitted smoothing gap. The smoothing gap mean does not exceed.

## 4. Results and Discussions

To provide a comparative study of the proposed approach, we apply the A-star directly on the 3D occupancy grid to find the shortest possible path. For that, we supply an inflated occupancy grid (OG) that takes into account the different

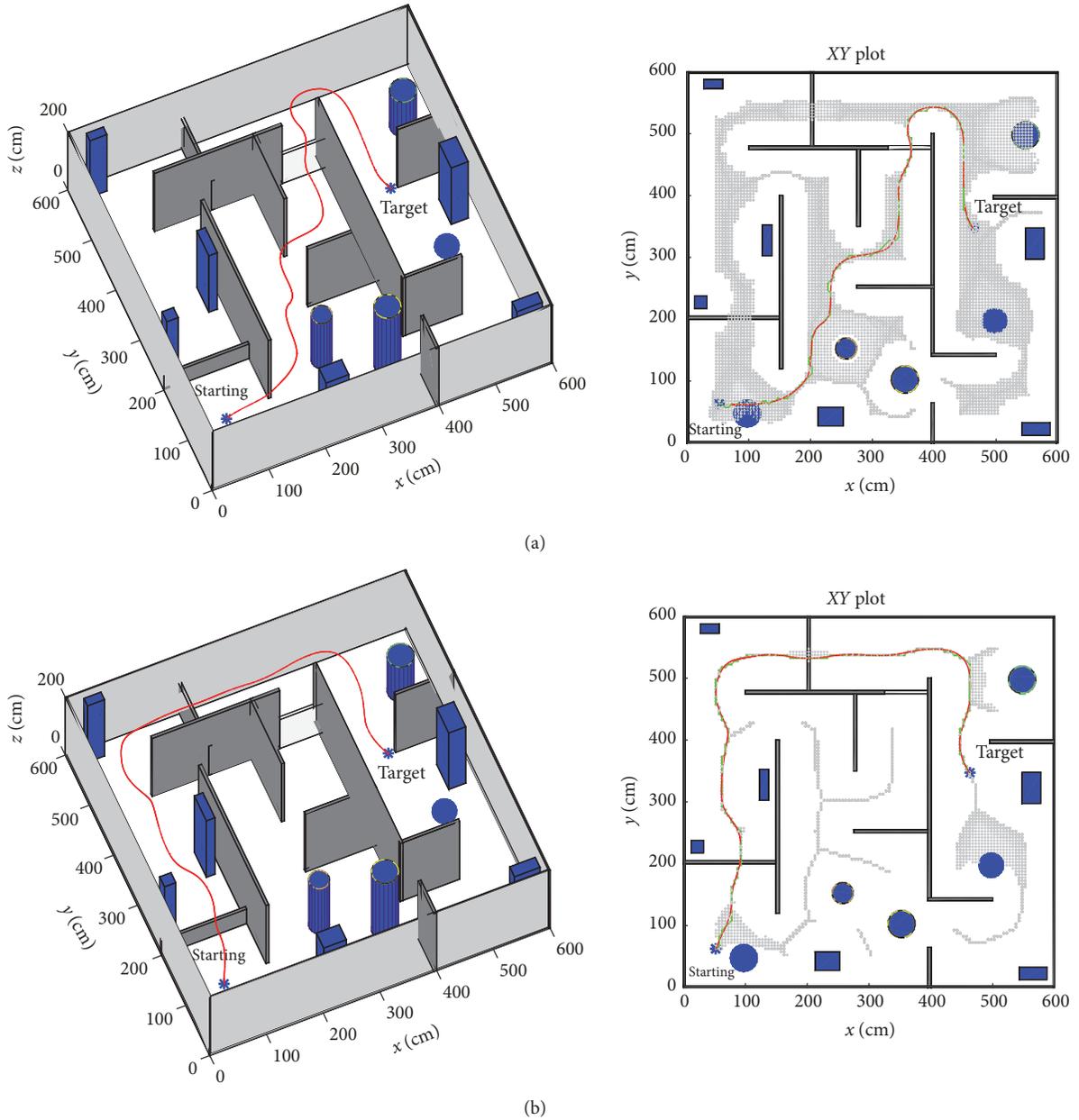


FIGURE 5: Final path for different UAV sizes (20 cm (a), 40 cm (b)).

TABLE I: Smoothing results.

	<i>case 1</i> $\delta = 20$ cm	<i>case 2</i> $\delta = 40$ cm
Maximum allowed smoothing gap (cm)	42.59	32.58
Maximum smoothing gap (cm)	8.13	8.98
Smoothing gap mean (cm)	2.70	2.55

UAV sizes considered before ( $\delta = 20$  cm and  $\delta = 40$  cm). The results are shown in Figure 6 compared with the solutions depicted in Figure 5.

To efficiently analyze the results, we define a set of quantitative criteria for the comparative study. The comparison results are shown in Table 2.

The input data (see Table 2) of the search algorithm depends on the type of the approach applied and the UAV dimension. When directly using the A-star on the occupancy grid (without skeletonization), the input data is the navigable space in the inflated occupancy grid. Using the proposed approach, it is the FMS of the navigable space in the original grid. The observations drawn from Table 2 are as follows:

- (i) The proposed approach offers a major decrease in the volume of the input data in both cases (90.15% and 87.21% of decrease, resp.) for an increase of,

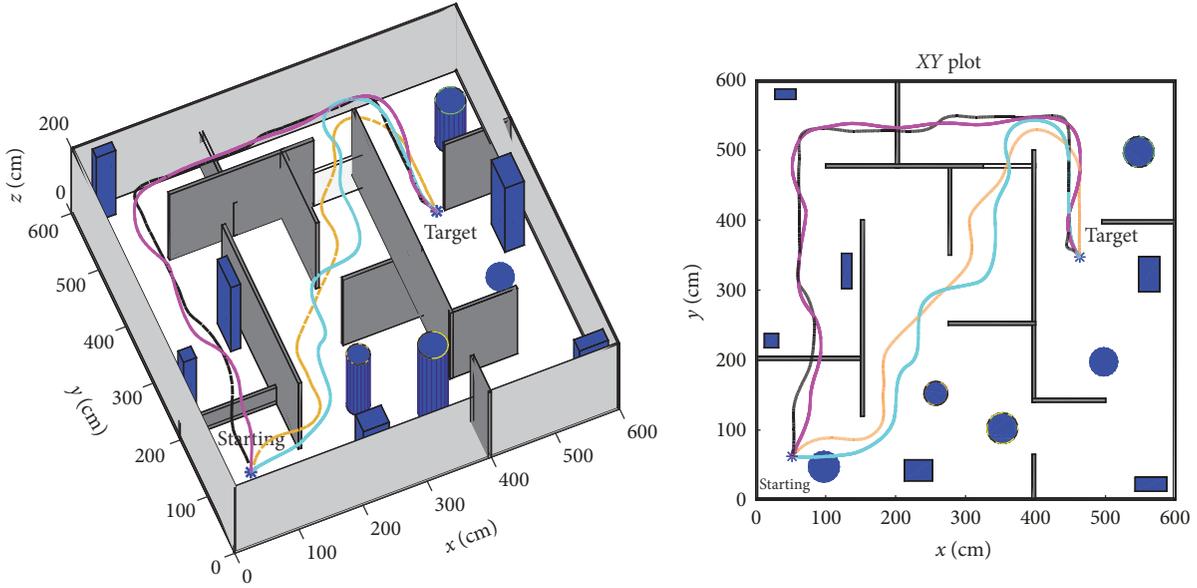


FIGURE 6: Comparison between the proposed approach based on different UAV sizes (20 cm (cyan line), 40 cm (magenta line)) and the A-star applied on the occupancy grid taking into account different UAV sizes (20 cm (orange line), 40 cm (black line)).

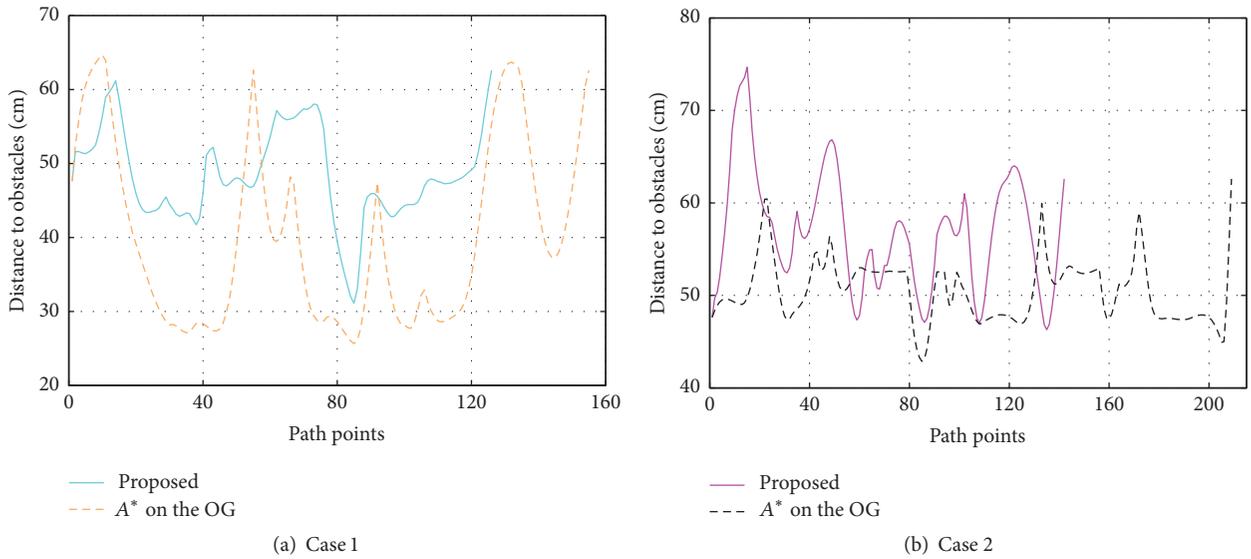


FIGURE 7: Evolution of the distance to obstacles ((a)  $\delta = 20$  cm, (b)  $\delta = 40$  cm).

TABLE 2: Quantitative comparison between the A-star applied on the OG and the proposed approach based on two different UAV sizes ( $\delta = 20$  cm (case 1),  $\delta = 40$  cm (case 2)).

	A-star on the OG		Proposed approach	
	case 1	case 2	case 1	case 2
Input data number	236644	61841	23324	7913
Path length (m)	9.03	11.08	9.95	11.41
Number of path points	155	209	126	142
Smallest dist. to obstacles (cm)	25.63	42.86	31.13	46.28
Mean of dist. to obstacles (cm)	40.18	50.51	48.42	56.88

respectively, 10.18% and 2.97% in the path length. The reduction in the research space is thanks to the skeletonization step. This is of great benefit when addressing real-time applications.

- (ii) For all cases, the smallest distance to obstacles is greater than the UAV size, which confirms that all planned paths are collision-free. The proposed approach gives more distance to obstacles supplying better path in terms of safety than the A-star algorithm applied directly on the OG. The evolution of the distance to obstacles along the path that the UAV is supposed to track is shown in Figure 7.

We perceive that the planned path provided by the proposed approach is not the optimum with regard to its length. However, it is more optimum with regard to the safety criterion than the path generated using the *A*-star directly on the occupancy grid.

The vehicle dynamics have been detailed in [34]. In this previous work, we have presented a generalized dynamical model of miniature multirotor vehicles. These UAVs are known for their high maneuverability. A control approach has been proposed and applied to track (user-designed) trajectories for an example of quadrotor, hexarotor, and octorotor UAVs.

As the smoothing step is performed to compute an approximate path requiring the least maneuvers of UAV without affecting its safety, we define two new quantitative parameters to better evaluate the proposed solution:

- (i)  $\Delta Z_{\max}$ : maximal change value of the altitude.
- (ii)  $\Delta \psi_{\max}$ : maximal change value of the heading angle.

For the two UAV sizes considered, the following numbers are obtained:

- (i) *In case 1* ( $\delta = 20$  cm):  $\Delta Z_{\max} = 08.78$  cm,  $\Delta \psi_{\max} = 28.43$  deg.
- (ii) *In case 2* ( $\delta = 40$  cm):  $\Delta Z_{\max} = 05.41$  cm,  $\Delta \psi_{\max} = 23.84$  deg.

These parameter values confirm that there are no aggressive changes in altitude and heading angle between two waypoints in the obtained path.

We note that, for all simulated cases, we consider the particular case where starting and destination points are on the FMS. This is not necessarily the case. If one of the initial and target points or both of them are not on the FMS, the idea is to reach (from a starting point) the closest point on the FMS and/or give up the FMS at the closest point to the destination. These additional portions at the beginning and/or the end of the path can be generated using 3D Bézier curves. An optimization process can be applied to guarantee the smoothness of the path at the connecting points.

## 5. Conclusion and Future Works

In this paper, we present an efficient solution for 3D path planning problem in aerial robotics. It is based on the extraction of the FMS by a 3D skeletonization algorithm. Including a pruning parameter  $\delta$ , the solution is given based on the UAV dimension discarding small branches. In the second step, we used the *A*-star algorithm to find the shortest path on the FMS. For smoothing, we proposed a method based on the use of Bézier curves.

In order to evaluate the performance of our approach, we compared the results to the *A*-star algorithm applied directly on the 3D occupancy grid. The quantitative criteria of comparison reveal that the proposed approach meets interesting performance criteria. Reducing interestingly the dimensionality of the searching space (by more than 87.21%) reduces memory requirement and is beneficial in terms of

computational cost. It is also proved that this approach attaches greater importance to the path safety compared to the path length, so it provides the shortest safest path.

Currently, the proposed approach uses the conventional *A*-star for the planning of the shortest path. An improved version of the *A*-star is obviously required especially when the ratio *UAV size/environment dimensionality* is smaller. In future works, we intend to consider curvature constraints of the planned path in the cost function. Furthermore, we propose dealing with the dynamic constraints of the UAV in order to find feasible trajectories within its maneuverability ability.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] S. M. LaValle, "Sampling-based motion planning," in *Planning Algorithms*, chapter 5, pp. 185–248, Cambridge University Press, 2006.
- [2] L. De Filippis, G. Guglieri, and F. Quagliotti, "Path planning strategies for UAVS in 3D environments," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1–4, pp. 247–264, 2012.
- [3] J. Carsten, D. Ferguson, and A. Stentz, "3D field D\*: improved path planning and replanning in three dimensions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3381–3386, October 2006.
- [4] B. Miller, K. Stepanyan, A. Miller, and M. Andreev, "3D path planning in a threat environment," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, vol. 6, pp. 6864–6869, IEEE, Orlando, Fla, USA, December 2011.
- [5] A. Chamseddine, Y. Zhang, C. A. Rabbath, C. Join, and D. Theilliol, "Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 2832–2848, 2012.
- [6] H. Duan and P. Li, *Bio-Inspired Computation in Unmanned Aerial Vehicles*, Springer, Berlin, Germany, 1st edition, 2014.
- [7] H. Ergezer and K. Leblebicioglu, "Path planning for UAVs for maximum information collection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 1, pp. 502–520, 2013.
- [8] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013.
- [9] L. Yang, J. Qi, J. Xiao, and X. Yong, "A literature review of UAV 3D path planning," in *Proceedings of the 11th World Congress on Intelligent Control and Automation (WCICA '14)*, pp. 2376–2381, IEEE, Shenyang, China, July 2014.
- [10] S. M. LaValle, "Rapidly-exploring random trees: a new tool for path planning," Tech. Rep., Computer Science Department, Iowa State University, 1998.
- [11] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

- [12] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [13] R. Bohlin and L. E. Kavraki, "Path planning using lazy PRM," in *Proceedings of the International Conference on Robotics and Automation*, pp. 521–528, April 2000.
- [14] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.
- [15] K. E. Bekris, B. Y. Chen, A. M. Ladd, E. Plaku, and L. E. Kavraki, "Multiple query probabilistic roadmap planning using single query planning primitives," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 656–661, October 2003.
- [16] D. Aarno, D. Kragic, and H. I. Christensen, "Artificial potential biased probabilistic roadmap method," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 461–466, April 2004.
- [17] P. O. Pettersson and P. Doherty, "Probabilistic roadmap based path planning for an autonomous unmanned helicopter," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 17, no. 4, pp. 395–405, 2006.
- [18] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees," *Journal of Intelligent & Robotic Systems*, vol. 71, no. 2, pp. 231–253, 2013.
- [19] F. Yan, Y.-S. Liu, and J.-Z. Xiao, "Path planning in complex 3D environments using a probabilistic roadmap method," *International Journal of Automation and Computing*, vol. 10, no. 6, pp. 525–533, 2013.
- [20] X. Wu, C. Guo, Y. Li, W. Chen, and Y. Wang, "The research of RRT route planning algorithm for UAV that based on kinematic equation," *International Journal of Control and Automation*, vol. 8, no. 1, pp. 287–296, 2015.
- [21] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," *Aerospace Science and Technology*, vol. 16, no. 1, pp. 47–55, 2012.
- [22] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [23] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [24] S. Koenig and M. Likhachev, "D\* Lite," in *Proceedings of the 18th National Conference on Artificial Intelligence*, pp. 476–483, August 2002.
- [25] A. Nash, S. Koenig, and M. Likhachev, "Incremental Phi\*: incremental any-angle path planning on grids," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 1824–1830, IEEE, July 2009.
- [26] A. K. Gurujji, H. Agarwal, and D. K. Parsediya, "Time-efficient A\* algorithm for robot path planning," *Procedia Technology*, vol. 23, pp. 144–149, 2016.
- [27] A. Nash, S. Koenig, and C. Tovey, "Any-angle path planning and path length analysis in 3D," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 147–154, July 2010.
- [28] Z. Qi, Z. Shao, Y. S. Ping, L. M. Hiot, and Y. K. Leong, "An improved heuristic algorithm for UAV path planning in 3D environment," in *Proceedings of the 2nd International Conference on Intelligent Human-Machine Systems and Cybernetics*, pp. 258–261, Nanjing, China, August 2010.
- [29] X. Liu, C. Zhou, M. Ding, and C. Cai, "Skeleton-based fast path planning for UAV," in *Remote Sensing and GIS Data Processing and Other Applications*, October 2009.
- [30] H. Blum, "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Form*, pp. 362–380, MIT Press, 1967.
- [31] R. Ogniewicz and M. Ilg, "Voronoi skeletons: theory and applications," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 63–69, Champaign, Ill, USA, 1992.
- [32] W. H. Hesselink and J. B. T. M. Roerdink, "Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 12, pp. 2204–2217, 2008.
- [33] R. Marie, O. Labbani-Igbida, and E. M. Mouaddib, "The delta medial axis: a fast and robust algorithm for filtered skeleton extraction," *Pattern Recognition*, vol. 56, pp. 26–39, 2016.
- [34] K. Benzaid, N. Mansouri, and O. Labbani-Igbida, "A generalized dynamical model and control approach applied to multirotor aerial systems," in *Proceedings of the 8th International Conference on Modelling, Identification and Control*, pp. 225–230, November 2016.

