algorithms

Article

An Effective Algorithm for Finding Shortest Paths in Tubular Spaces

Dang-Viet-Anh Nguyen [1,*], Jérôme Szewczyk [2] and Kanty Rabenorosoa [1]

1 FEMTO-ST Institute, University Bourgogne Franche-Comté, CNRS, 25000 Besançon, France; rkanty@femto-st.fr
2 Institute for Intelligent Systems and Robotics (ISIR), Sorbonne University, CNRS, INSERM, 75005 Paris, France; szewczyk@isir.upmc.fr
* Correspondence: dang.nguyen@femto-st.fr

Abstract: We propose a novel algorithm to determine the Euclidean shortest path (ESP) from a given point (source) to another point (destination) inside a tubular space. The method is based on the observation data of a virtual particle (VP) assumed to move along this path. In the first step, the geometric properties of the shortest path inside the considered space are presented and proven. Utilizing these properties, the desired ESP can be segmented into three partitions depending on the visibility of the VP. Our algorithm will check which partition the VP belongs to and calculate the correct direction of its movement, and thus the shortest path will be traced. The proposed method is then compared to Dijkstra's algorithm, considering different types of tubular spaces. In all cases, the solution provided by the proposed algorithm is smoother, shorter, and has a higher accuracy with a faster calculation speed than that obtained by Dijkstra's method.

Keywords: Euclidean shortest path; tubular space; reactive algorithm; visibility; oriented drilling process; Dijkstra's algorithm

check for updates

Citation: Nguyen, D.-V.-A.; Szewczyk, J.; Rabenorosoa, K. An Effective Algorithm for Finding Shortest Paths in Tubular Spaces. *Algorithms* **2022**, *15*, 79. https://doi.org/10.3390/a15030079

Academic Editor: Lijun Chang

Received: 2 February 2022
Accepted: 24 February 2022
Published: 25 February 2022

1. Introduction

Finding the shortest path in the presence of obstacles, referred to as the Euclidean shortest path problem, is one of the fundamental problems in path planning [1]. This problem arises in many industrial applications. The idea of using a flying robot such as an unmanned aerial vehicle (UAV) to navigate through a tunnel-like environment can be found in the inspection of dam penstocks [2–4], chimneys [5], ventilation systems [6], onshore oil and gas industry [7], narrow sewers [8], and other hazardous deep tunnels [9,10]. In addition, many marine applications also require navigating through underwater tunnel-like environments with autonomous underwater vehicles (AUVs). These include, for instance, the inspection of different kinds of underwater structures such as offshore oil platforms [11], flooded spring tunnels [12,13], water delivery tunnels [14], etc. In these applications, shortest path planning that minimizes the total distance travelled by the vehicles plays an important role in optimizing the energy consumption, thus extending the operation time without recharging their batteries [15,16]. It may also reduce the travelling time and will be very useful for search and rescue missions during disaster events in underground tunnels [6,17].

Another example of the studied problem is to determine the location of a non-elastic chord between two points within a tube. Indeed, this problem can be found in controlling the deformation of slender tube-like robots actuated with at least one internal tendon [18,19]. Calculating the tendon load effect on the tube wall requires determining the tendon location. As it is only attached at the tip, pre-positioned at the base, and the rest freely locates inside the innermost tube, its location results in the shortest path connecting two points at the base and at the tip.

1.1. Related Works

In general 3D space, the problem of finding the ESP between two given points that does not intersect with any given obstacles is known to be NP-hard [20]: special cases of this problem have been studied as follows. The authors in [21] gave a polynomial time algorithm to calculate ESP for cases in which the number of obstacles is 'small' and all of them are convex. Another algorithm was proposed in [22] with the assumption that all obstacles are vertical buildings with k different heights. More recently, Ref. [23] presented algorithms for solving approximate ESPs amid convex obstacles. Other approximation algorithms for ESP calculations are detailed in [24]. These studies share common features in that a collection of finite obstacles are given as forbidden zones in space and the ESP will be found in the space surrounding these obstacles.

In the studied problem, the obstacle is the entire space outside the tube and the ESP must pass through the inner zone of the tubular space. A similar problem can be found in [25] that computes the minimal path in tubular space. The minimal path is typically solved based on the Fast Marching Method (FMM) which only considers grid nodes as the possible vertices of the minimal paths. However, the paths detected by the FMM have been proven to be not always the exact ESPs [1]. Several approximation algorithms also exist for finding the ESP between two points in 3D space bounded by a closed surface such as cube-curves [26] or a simple polyhedron [27] using the rubberband algorithm. This method is suitable for solving various ESPs in 3D space. Even so, there is a non-trivial gap in the geometric shape between the cube-curve and the tubular space. Polyhedron seems like a better choice to represent a tubular space. However, the characteristic geometrical properties of tubular spaces should be considered for a dedicated algorithm.

The study of a tubular surface with Bishop frame was proposed in [28]. The authors provided some characterizations about the special curves lying on this surface (e.g., geodesic and asymptotic curves). However, the problem studied herein requires considering the interior space instead of just the boundary surface. In addition, the geometrical properties of the ESP inside the tubular space also need to be investigated. The authors in [29] described a simple geometric structure of ESPs where they consist of curved paths on the obstacles connected by straight line segments (see Theorem 1). In this work, we develop the geometric structure of ESPs presented in [29] by considering the characteristic properties of tubular spaces.

In practice, the navigation problem can be classified into planning-based and reactive algorithms [30]. Planning-based approaches require a global map representation of the environment (e.g., a graph or a network) before searching. Prior knowledge of the tubular space enables the generation of a weighted graph where the weight of each edge (or arc for the directed graph) is associated with its length [24]. Numerous algorithms are used for the shortest path calculation in graph theory (see Chapters 24 and 25 in [31]). A well-known graph-based algorithm among them is Dijkstra's algorithm [32] in which the shortest path connects vertices in the graph. Unlike the planning-based approaches, a reactive method allows directly generating motion decisions during the movement based on observed data [30]. The reactive shortest path navigation was presented in [33] for an in-plane problem. Such a problem was also found in 3D space where the obtained path is interpolated with a spline curve [34]. In this work, we propose an algorithm based on the observed information of a virtual particle that can be used as a reactive method for the shortest path navigation inside the tubular space.

1.2. Contributions

In this paper, we propose a novel algorithm to find the shortest path within a tubular space that connects two points at the tube ends. Our contributions include: (1) the description of the ESP geometrical structure inside tubular spaces with mathematical proof; (2) the proposition of a novel algorithm for finding the shortest path in tubular spaces based on the observed data; and (3) the numerical validation and comparison results with Dijkstra's algorithm by considering various types of tubular spaces. As a result, the solution obtained by using the proposed algorithm is shorter, smoother and faster than that provided by Dijkstra's method.

The remainder of this paper is organized as follows. Section 2 formulates the problem. The basics of Dijkstra's algorithm are described in Section 3. Then, we present the proposed algorithm in Section 4. Our computational results are provided in Section 5. After that, Section 6 includes some brief discussions. Section 7 concludes the paper.

2. ESP in Tubular Space

2.1. Problem Description

Euclidean geometry is the geometry in daily life [1] where the distance between two points $\boldsymbol{p} = (x_p, y_p, z_p)^T$ and $\boldsymbol{q} = (x_q, y_q, z_q)^T$ in 3D space is defined as follows:

$$d_e(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2 + (z_p - z_q)^2} \tag{1}$$

From the discrete point of view, a path (α) from the source $\mathfrak{P}$ to the destination $\mathfrak{Q}$ is a finite sequence of nodes $\boldsymbol{x}_i$, starting at $\mathfrak{P}$ and ending at $\mathfrak{Q}$. We obtain the length of the path as in Equation (2):

$$L(\alpha) = \sum_{i=0}^{n-1} d_e(\boldsymbol{x_i}, \boldsymbol{x_{i+1}}), \quad \boldsymbol{x}_0 = \mathfrak{P}, \boldsymbol{x}_n = \mathfrak{Q} \tag{2}$$

Then, the ESP is the path connecting $\mathfrak{P}$ and $\mathfrak{Q}$, which has the minimum length and has to be through a given tubular space. The mathematical definition of tubular space is given as follows [35]:

Definition 1. *Let $\boldsymbol{c}(s) : I \rightarrow \mathbb{R}^3$ be a smooth, regular space curve. A tubular surface $\partial\Omega$ associated to $\boldsymbol{c}(s)$, of radius ρ, is, by definition, the envelope of the family of spheres of radius ρ, with the center on the curve.*

Definition 2. *The storage space of the tube Ω is the 3D space enclosed by the lateral wall ($\partial\Omega$) and the two ending cross-sections of the tube.*

In Definition 1, s is the arc length parameter of the centerline curve. In this work, we consider the tubular surface $\partial\Omega$ to be regular (Figure 1). The condition underlying the regularity of a tube is given in detail in [28]. By $\kappa(s)$, we denote the curvature of the centerline curve $\boldsymbol{c}(s)$. In order to avoid singularities as well as self-overlapping, the following condition is required:

$$\kappa(s) < \rho^{-1}, \forall s \in [0, L] \tag{3}$$

where L is the length of $\boldsymbol{c}(s)$.

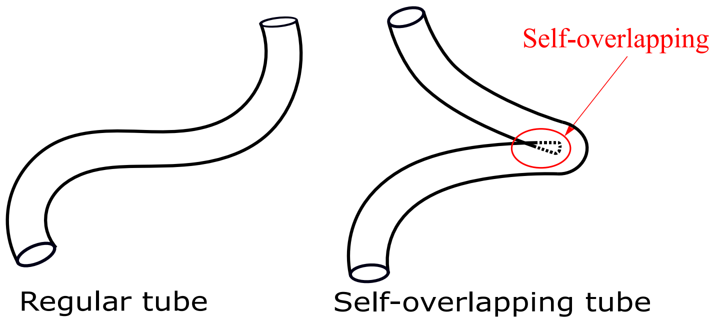


Figure 1. The regular and self-overlapping tube. The regular tube ensures the correctness of the directed graph in the following section.

2.2. Directed Graph

We discretize Ω into a series of meshed circular disks corresponding to the cross-sections perpendicular to $\boldsymbol{c}(s)$. By $S_0, \ldots, S_{N+1}$, we denote the meshed circular disks where S_0 contains the starting point (source) $\mathfrak{P}$ and S_{N+1} includes the destination $\mathfrak{Q}$. The distance between two consecutive disks along the centerline curve is $h = \frac{L}{N+1}$. As the shortest path from the source to the destination must obviously pass through each cross-section at only one point, we have the weighted directed graph $G(V, A)$, as shown in Figure 2. This directed graph is defined by a finite set V of *vertices* and a set A of *arcs* between those vertices [1]. All vertices of the graph (except $\mathfrak{P}$ and $\mathfrak{Q}$) are located at the nodes of the meshed disks $S_1, \ldots, S_N$. We define two vertices as adjacent if they are connected by one arc. Then, every two adjacent nodes in the graph are located on two consecutive disks. The source $\mathfrak{P}$ is connected with all nodes of disk S_1. Each node of disk S_i is connected by one arc to every node of disk S_{i+1} for all $i \in \{1, \ldots, N-1\}$. Eventually, every node of disk S_N is directly connected to the destination $\mathfrak{Q}$.

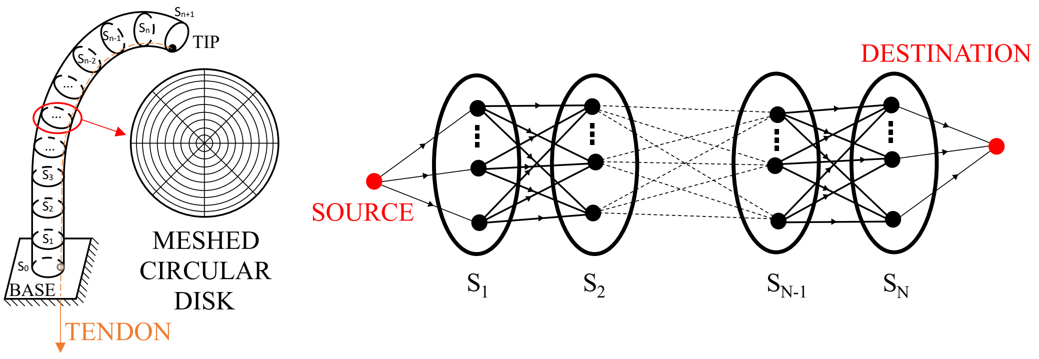


Figure 2. Discrete approach for the ESP problem. (**Left**) Inner space of the tube transformed into a series of meshed circular disks; and (**Right**) the directed graph.

3. Basics of Dijkstra's Algorithm

The concept of this method, based on the lemma about the relationship between the global minimum and local minimum, was first presented by E.W. Dijkstra in 1959 (see [32], Problem 2). Despite also being based on Lemma 1, if applied to the directed graph, Algorithm 1 has a run time in $O(|A|)$ instead of $O(|V|log|V| + |A|)$, as is the case for the conventional Dijkstra's algorithm used for a weighted graph [1]. In the literature, the A^* algorithm [36] is a good method for searching the shortest path in a weighted graph as it improves the calculation speed of Dijkstra's algorithm for many cases. However, choosing an effective admissible heuristic function for this algorithm based on the geometry of the tubular space is beyond the scope of this paper. In addition, the complexity $O(|A|)$ is already a good run time for this problem, so we do not consider the A^* algorithm in this work.

Lemma 1 (Dijkstra algorithm, 1959)**.** *"If $\boldsymbol{r}$ is a node on the minimal path from $\boldsymbol{p}$ to $\boldsymbol{q}$, the knowledge of the latter implies the knowledge of the minimal path from $\boldsymbol{p}$ to $\boldsymbol{r}$".*

This lemma can easily be proven by contradiction. The shortest path from the source $\mathfrak{P}$ to the destination $\mathfrak{Q}$ will be traced by extending all extendable paths by one edge to a node not yet visited on this path until $\mathfrak{Q}$ is reached. Consequently, extending all the extendable paths from the source by one arc in the directed graph starts to turn the examined disk into its adjacent disk towards the destination. Upon examining a disk, the shortest path between the source and every node on the previous disks has been identified, so we do not need to revisit these points.

Set $\boldsymbol{r}_{[i][j]} (i \in \{1, \ldots, N\}, j \in \{1, \ldots, M\})$ as the node jth on the meshed disk S_i. In addition, we denote $D(\boldsymbol{x})$ as the minimum length from node $\mathfrak{P}$ to node $\boldsymbol{x}$, $\mathrm{w}(\boldsymbol{x}, \boldsymbol{y})$ as the length of the arc connecting two adjacent nodes $\boldsymbol{x}$ and $\boldsymbol{y}$, and $\mathcal{L}_{[i][j]}$ as the list of nodes on the shortest path from source $\mathfrak{P}$ to node $\boldsymbol{r}_{[i][j]}$ (for node $\mathfrak{Q}$, we utilize $\mathcal{L}_{[\mathfrak{Q}]}$). We obtain Dijkstra's algorithm applied for this ESP problem as given in Algorithm 1.

Algorithm 1: Dijkstra's algorithm.

```
    Input: $\mathfrak{P}, \mathfrak{Q}$, and $\boldsymbol{r}_{[i][j]}, \forall i \in \{1, \ldots, N\}, \forall j \in \{1, \ldots, M\}$.
    Output: $\mathcal{L}_{[\mathfrak{Q}]}$.
    // Initialisation
 1  $D(\mathfrak{Q}) \leftarrow +\infty, D(\boldsymbol{r}_{[i][j]}) \leftarrow +\infty, \forall i \in \{1, \ldots, N\}, \forall j \in \{1, \ldots, M\}$;
    // From the source to $S_1$
 2  for $j \leftarrow 1$ to $M$ do
 3      $D(\boldsymbol{r}_{[1][j]}) \leftarrow \mathrm{w}(\mathfrak{P}, \boldsymbol{r}_{[1][j]})$;
 4      $\mathcal{L}_{[1][j]} \leftarrow \{\mathfrak{P}, \boldsymbol{r}_{[1][j]}\}$;
 5  end
    // Between $S_1$ and $S_N$
 6  for $i = 2$ to $N$ do
 7      for $j = 1$ to $M$ do
 8          for $k = 1$ to $M$ do
 9              $D(\boldsymbol{r}_{[i][j]}) \leftarrow min\{D(\boldsymbol{r}_{[i][j]}), D(\boldsymbol{r}_{[i-1][k]}) + \mathrm{w}(\boldsymbol{r}_{[i-1][k]}, \boldsymbol{r}_{[i][j]})\}$
10              If $D(\boldsymbol{r}_{[i][j]})$ is replaced, put a label $K^* \leftarrow k$.
11          end
12          $\mathcal{L}_{[i][j]} \leftarrow \left[\mathcal{L}_{[i-1][K^*]}, \boldsymbol{r}_{[i][j]}\right]$ ;        // add $\boldsymbol{r}_{[i][j]}$ to the list $\mathcal{L}_{[i-1][K^*]}$
13      end
14  end
    // From $S_N$ to the destination
15  for $k = 1$ to $M$ do
16      $D(\mathfrak{Q}) \leftarrow min\{D(\mathfrak{Q}), D(\boldsymbol{r}_{[N][k]}) + \mathrm{w}(\boldsymbol{r}_{[N][k]}, \mathfrak{Q})\}$.
17      If $D(\mathfrak{Q})$ is replaced, put a label $K^* \leftarrow k$.
18  end
19  return $\mathcal{L}_{[\mathfrak{Q}]} \leftarrow [\mathcal{L}_{[N][K^*]}, \ \mathfrak{Q}]$
```

During the operation, all currently visited nodes always belong to the same cross-section. Thus, all possible paths will reach the destination at the same time. When the destination is reached, there will no longer be extendable paths in the directed graph and we can point out the shortest path. For that reason, the algorithm becomes a breadth-first search algorithm [1]. Thus, the time complexity of Algorithm 1 is $O(|A|)$ with $|A|$ as the number of arcs in the directed graph. As a conventional method, the solution by Dijkstra's algorithm is given as a series of vertices of the graph G. Then, the obtained solution path is generally a polyline. To increase the accuracy of the result as well as make it smoother, the mesh of the discretized cross-section must be finer. However, increasing the number of nodes on the mesh results in significantly slowing down the calculation speed. We then proposed a new method that takes advantage of the geometrical properties of tubular spaces to improve the searching solution.

4. The ESP Searching Algorithm Based on Visibility

The algorithm we propose hereafter is based on a visible tube portion that can be "seen" by the VP moving along the shortest path it is searching. The ESP will be gradually established determining the correct moving direction of the particle from the source $\mathfrak{P}$ to the destination $\mathfrak{Q}$. For convenience, we firstly define some concepts used in this section.

4.1. Geometric Properties of the ESP in Tubular Space

Definition 3. *For every point* $\mathbf{X} \in S_i \subset \Omega, \ \ i \neq N+1$, *the cross-section* S_i *divides* Ω *into two sub-spaces, and a direction from* $\mathbf{X}$ *is said to be positive (+) if it is towards the sub-space containing the destination.*

According to Definition 3, any point in Ω (not belonging to S_{N+1}) will have an infinite number of positive directions (see Figure 3). Obviously, during the movement, the correct direction of the particle is always a positive direction.

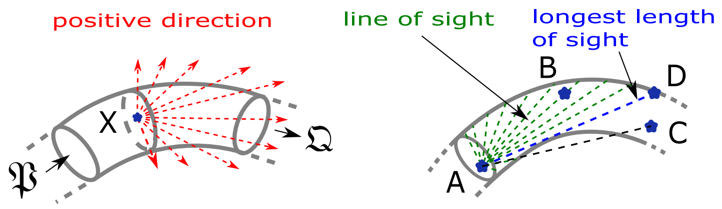


Figure 3. (**Left**) The dashed red rays describe the positive directions. (**Right**) $\boldsymbol{A}$ can see $\boldsymbol{B}$ and $\boldsymbol{D}$ because the line segments $\overline{\boldsymbol{AB}}$ and $\overline{\boldsymbol{AD}}$ are totally contained by Ω. Furthermore, by this definition, $\boldsymbol{A}$ cannot see $\boldsymbol{C}$. The green and blue dashed lines terminating at the boundary $\partial\Omega$ illustrate the line of sights. Among them, the blue one is the longest length of sight, an important concept used in the following method.

Definition 4. *Two points* $\mathbf{X}, \ \mathbf{Y} \in \Omega$ *are said to see each other if the line segment joining them* $\overline{\mathbf{XY}}$ *is totally contained by* Ω.

Definition 5. *A cross-section* $S \subset \Omega$ *is visible from a point* $\boldsymbol{X} \in \Omega$ *if there exists a point* $\boldsymbol{Y} \in S$ *that can be seen by* $\boldsymbol{X}$.

Lemma 2. *If a point* $\boldsymbol{X}$ *inside the tube can see a cross-section* S *of the tube, the area part in* S *that can be seen by* $\boldsymbol{X}$ *must be a convex set (as illustrated in Figure 4).*

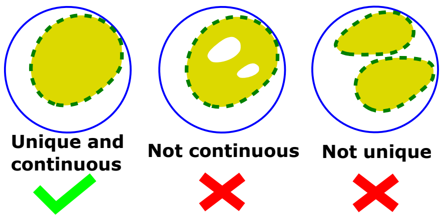


Figure 4. The visible area of a cross-section S_i is described by the yellow zone(s) which must be unique and continuous.

The proof of this lemma is detailed in Appendix A.1.

Definition 6. *From a point* $\mathbf{X} \in \Omega$, *the "**length of sight**" corresponding to a positive direction is the distance between* $\mathbf{X}$ *and the farthest point in* $\partial\Omega$ *that can be seen by* $\mathbf{X}$ *along the positive direction. The line segment corresponding to this length is called the **line of sight**.*

We then have the geometric structure of the shortest path between two points (which are not in each other's line of sight) in a tubular space.

Theorem 1. *By* $\boldsymbol{f}$, *we denote the shortest path between two arbitrary points* $\mathbf{X}$ *and* $\mathbf{Y}$ *inside the tubular space* Ω. *If* $\mathbf{X}$ *cannot see* $\mathbf{Y}$, *then there exist curved parts of* $\boldsymbol{f}$ *lying on the inner lateral wall of the tube* $\partial\Omega$. *Outside these parts,* $\boldsymbol{f}$ *consists of a union of straight line segments which are tangent to the boundary surface* $\partial\Omega$.

Theorem 2. *The curved parts of* $\boldsymbol{f}$ *are geodesic paths on* $\partial\Omega$*. Moreover, they are on the surface of the negative curvature.*

Proof. The proof of Theorem 1 was presented in [29]. To prove Theorem 2, we employed Lemma 1. As the curved parts of $\boldsymbol{f}$ are also the shortest paths connecting their ends, they must be geodesic paths on the boundary surface $\partial\Omega$ [37]. Moreover, if a curved path of $\boldsymbol{f}$ exists that is outside the surface of the negative curvature, we always find on this path two neighboring points that can see each other (see Figure 5). As the straight line segment joining these points is shorter than the geodesic path between them, then $\boldsymbol{f}$ is not the shortest one connecting $\boldsymbol{X}$ and $\boldsymbol{Y}$. This contradicts the definition of $\boldsymbol{f}$ (Q.E.D.). □

These two theorems lead us to two important corollaries.

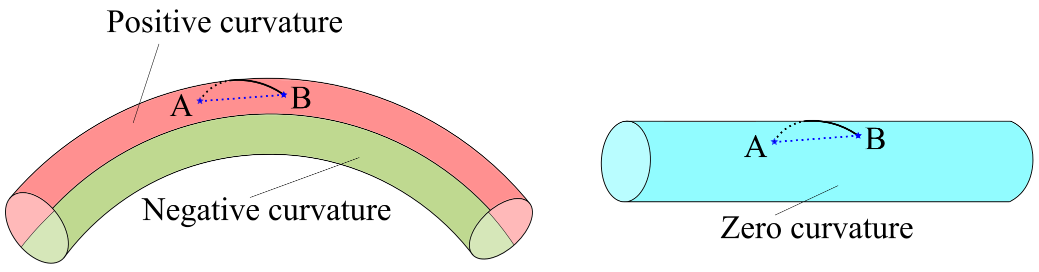


Figure 5. Curved segments lying on surfaces of positive and zero curvatures where A and B can see each other.

Corollary 1. *Let* $\boldsymbol{X}$ *be a point on the ESP* $\boldsymbol{p}(s)$ *that can see a point* $\boldsymbol{Y}$ *so that the ray* $\boldsymbol{XY}$ *is not the direction* $\dot{\boldsymbol{p}}(s_X)$ *of the ESP at* $\boldsymbol{X}$*. Let* (α) *be an arbitrary plane containing* $\boldsymbol{XY}$*. If the angle between* $\dot{\boldsymbol{p}}(s_X)$ *and* (α) *is not zero, then the direction* $\dot{\boldsymbol{p}}(s)$ *at any point on the ESP segment between the cross-sections containing* $\boldsymbol{X}$ *and* $\boldsymbol{Y}$ *will always point away from* (α)*.*

Proof. From $\boldsymbol{X}$, the particle moves away from (α) (the angle between $\dot{\boldsymbol{p}}(s_X)$ and (α) is not zero). Using Theorem 1, we obtain that the particle only changes its direction at points on the geodesic paths. As these curves must be on the surface of negative curvature (Theorem 2), where vector $\ddot{\boldsymbol{p}}(s)$ points out of the tube, thus the $\dot{\boldsymbol{p}}(s)$ will always point away from the plane (α) (see Figure 6). □

Corollary 2. *If* $\boldsymbol{X}$ *on the ESP can see a cross-section* S *of the tubular space* Ω *via a positive direction, the correct direction at* $\boldsymbol{X}$ $(\dot{\boldsymbol{p}}(s_X))$ *must be towards a point* $\boldsymbol{Y}$ *in the visible area of* S *by* $\boldsymbol{X}$*.*

Proof. The above corollary can be proven by contradiction. By $\sigma_X(S)$, we denote the visible area of the cross-section S by $\boldsymbol{X}$. Let $\boldsymbol{Y}$ be the intersection of the straight line containing $\dot{\boldsymbol{p}}(s_X)$ and the plane containing S (denoted by $\beta(S)$). We need to prove that $\boldsymbol{Y} \in \sigma_X(S)$. We consider the following hypothesis of contradiction:

$$if\ \boldsymbol{Y} \notin \sigma_X(S) \Rightarrow \overline{\boldsymbol{XY}} \not\subset \Omega \tag{4}$$

In other words, the ray $\boldsymbol{XY}$ passes through the boundary $\partial\Omega$. Let $\boldsymbol{M}$ be the passing point that is closest to $\boldsymbol{X}$. In $\beta(S)$ and through $\boldsymbol{Y}$, we draw an arbitrary straight line that intersects with the visible area $\sigma_X(S)$. Let $\boldsymbol{W}$ be the intersection point that is closest to $\boldsymbol{Y}$. Then, $\boldsymbol{W}$ must be on the boundary of $\sigma_X(S)$ (denoted by $\partial\sigma_X(S)$). In addition, there is a total of two relative positions of $\boldsymbol{W}$: $\boldsymbol{W} \in \partial\Omega$ and $\boldsymbol{W} \notin \partial\Omega$ (see Figure 7). In the following, we define a plane (α) and a closed surface (C) for these two mentioned cases:

- $(\boldsymbol{W} \notin \partial\Omega)$. (α) is the plane that contains $\boldsymbol{XW}$ and the tangent at $\boldsymbol{W}$ of $\sigma_X(S)$. If $\overline{\boldsymbol{XW}}$ is not tangent to $\partial\Omega$, we can always find in $\beta(S)$ a circle with center $\boldsymbol{W}$ and radius ϵ small enough so that the entire circle can be seen by $\boldsymbol{X}$ (as there is no obstacle between $\boldsymbol{X}$ and this circle). Then, there exist points outside $\sigma_X(S)$ (which is part of the circle) that can be seen by $\boldsymbol{X}$. This contradicts the definition of $\sigma_X(S)$. Thus, $\overline{\boldsymbol{XW}}$ is tangent to

$\partial\Omega$. Let $\boldsymbol{T}$ be the tangent point that is closest to $\boldsymbol{X}$ and $P_T(\partial\Omega)$ be the tangent plane of $\partial\Omega$ at $\boldsymbol{T}$. If $P_T(\partial\Omega) \not\equiv (\alpha)$, $P_T(\partial\Omega)$ will divide $\sigma_X(S)$ into two subsets. However, as the line of sight of a visible point in $\sigma_X(S)$ must pass through the cross-section of the tube at $\boldsymbol{T}$, only then can one subset of $\sigma_X(S)$ be observed by $\boldsymbol{X}$. This contradicts the definition of $\sigma_X(S)$. Thus, $P_T(\partial\Omega) \equiv (\alpha)$. We can then define a closed surface (C) enclosed by (α), the cross-section at $\boldsymbol{X}$, and part of $\partial\Omega$ which contains $\boldsymbol{M}$ (see Figure 7 Left).

- $(\boldsymbol{W} \in \partial\Omega)$. (α) is the plane that contains $\boldsymbol{XW}$ and the tangent at $\boldsymbol{W}$ of S. Then, (C) is the closed surface containing $\boldsymbol{M}$ and enclosed by $\partial\Omega$, (α), and the cross-section of the tube at $\boldsymbol{X}$ (see Figure 7 Right).

By using the definition of (C), we can deduce from $\boldsymbol{X}$ that the VP will go into the inner space of (C). Since the destination $\mathfrak{Q}$ is outside (C), the particle must pass the boundary of (C) somewhere on (α). However, by applying Corollary 1, the direction vector $\dot{\boldsymbol{p}}(s)$ will always point away from (α), thus the particle cannot return to (α) for a passing point. Therefore, the hypothesis (4) cannot be true, then $\boldsymbol{Y} \in \sigma_X(S)$ (Q.E.D.). $\square$

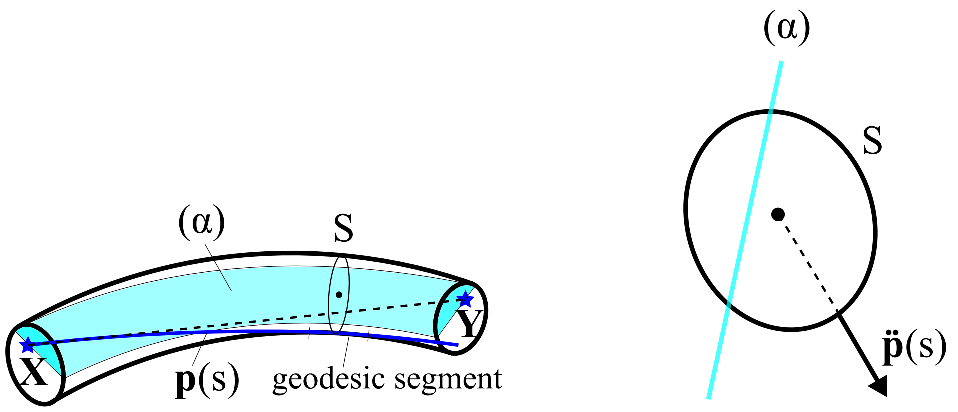


Figure 6. (**Left**) Tube portion between the cross-sections containing $\boldsymbol{X}$ and $\boldsymbol{Y}$ which can see each other. (α) is an arbitrary plane containing $\boldsymbol{XY}$ but not containing $\dot{\boldsymbol{p}}(s_X)$. The ESP $\boldsymbol{p}(s)$ only changes its direction $\dot{\boldsymbol{p}}(s)$ on its geodesic segment(s). S is an arbitrary cross-section of the tube where the geodesic segment crosses. (**Right**) On the projection view plane that is perpendicular to (α) ((α) degenerates to a straight line), as $\ddot{\boldsymbol{p}}(s)$ points outside the envelope of S, it also points away from (α). As $\dot{\boldsymbol{p}}(s_X)$ points away from (α), by mathematical induction, $\dot{\boldsymbol{p}}(s)$ will point away from (α), $\forall s \in [s_X, s_Y]$.

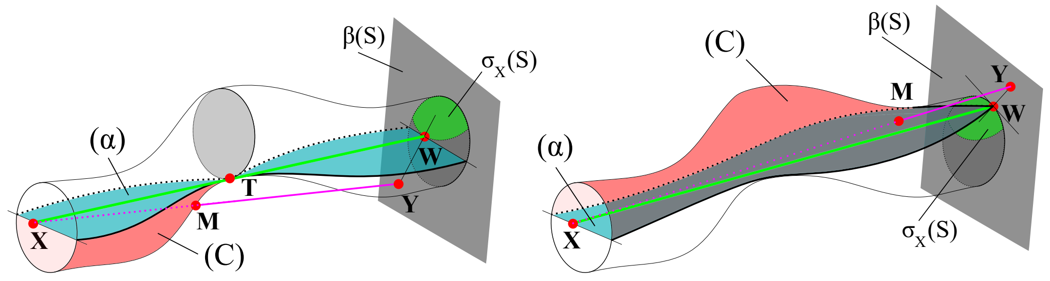


Figure 7. Point $\boldsymbol{X}$ can see cross-section S. $\boldsymbol{Y}$ is the intersection of the direction of the ESP $\dot{\boldsymbol{p}}(S_X)$ and the plane containing S. In $\beta(S)$ and through $\boldsymbol{Y}$, we draw an arbitrary straight line that intersects the visible area $\sigma_X(S)$. Let $\boldsymbol{W}$ be the intersection point that is closest to $\boldsymbol{Y}$. (**Left**) $\boldsymbol{W}$ is in the inner zone of S; (**Right**) $\boldsymbol{W}$ is on the boundary of S.

Employing the above lemma, theorems and corollaries lead us to an important result regarding the partitions of the ESP inside a tubular space, as shown in Remark 1.

Remark 1. *For any type of tubular space, the shortest path $\boldsymbol{p}(s)$ can be segmented into three partitions:*

- *Partition 1 (**P1**) : Includes points that can see the destination $\mathfrak{Q}$. The direction $\dot{\boldsymbol{p}}(s)$ at any point in this partition is always towards $\mathfrak{Q}$.*

- *Partition 2 (**P2**) : Includes points that can see the ending cross-section S_{end}, but cannot see $\boldsymbol{\Omega}$. The direction $\dot{\mathbf{p}}(s)$ at any point $\boldsymbol{X}$ in this partition is always towards a visible point $\boldsymbol{Y}$ in the ending cross-section such that the angle between $\boldsymbol{XY}$ and $\boldsymbol{X\Omega}$ is the smallest one.*
- *Partition 3 (**P3**) : Includes points that cannot see the ending cross-section S_{end}. The direction of $\dot{\mathbf{p}}(s)$ at any point in this partition is the positive direction corresponding to the longest length of sight.*

The three partitions of the searching shortest path inside the tube are described in Figure 8. The proof of this remark is given in Appendix A.2. It is important to note that a tube does not necessarily contain all three partitions. For instance, a straight tube only contains partition 1 regardless of the positions of the source and the destination.

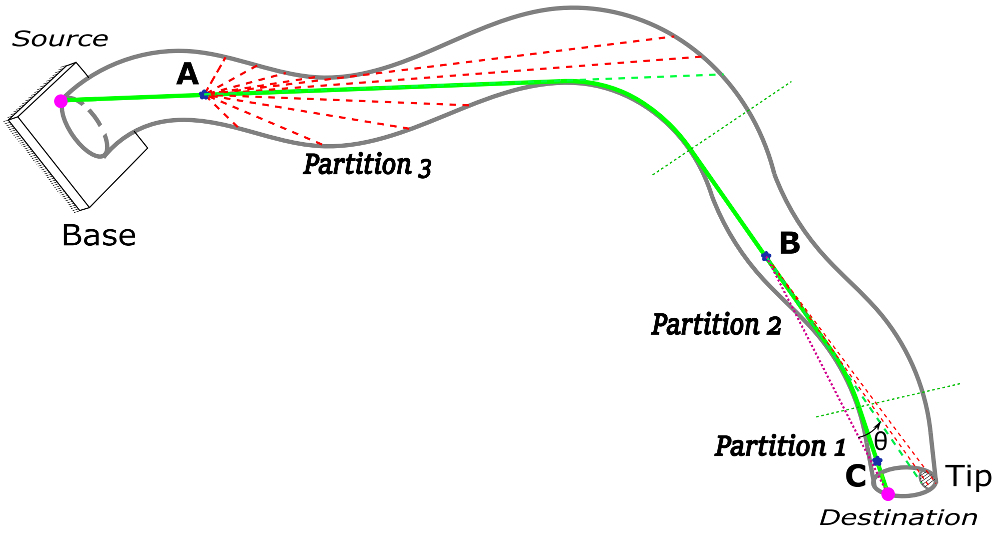


Figure 8. Three partitions of the shortest path correspond to three sections of the tube. At $\boldsymbol{A}$ belonging to **P3**, the VP cannot see the ending cross-section S_{end}. The correct direction corresponds to the longest length of sight. At $\boldsymbol{B}$ belonging to **P2**, S_{end} can see been, but not $\boldsymbol{\Omega}$. The correct direction is towards the visible point $\boldsymbol{Y}$ in S_{end} so that the angle θ between $\boldsymbol{BY}$ and $\boldsymbol{B\Omega}$ is the smallest one. At $\boldsymbol{C}$ in **P1**, the particle can see $\boldsymbol{\Omega}$. The correct direction is towards $\boldsymbol{\Omega}$.

4.2. The Proposed Algorithm

The principle of this method is based on Remark 1. Ensuring that two points can see each other in the discrete approach is equivalent to proving that the line segment joining those points must cross all the meshed circular disks between them. In the Algorithm 2, we use $\mathbf{C}_i$ to denote the intersection point between the searching shortest path and the cross-section $S_i,\ i \in \{0,\ldots,N+1\}$. The objective of Algorithm 2 is then equivalent to finding the series of $\mathbf{C}_i$. It is important to note that $\mathbf{C}_i$ is not necessarily a node of $G(V,A)$. Indeed, the algorithm determines the correct direction of $\mathbf{C}_{i-1}$, thereby determining the position of $\mathbf{C}_i$ as the intersection point between this direction and the next cross-section S_i. The correct direction of $\mathbf{C}_{i-1}$ was found using Remark 1 by checking which partition $\mathbf{C}_{i-1}$ belongs to (with precedence from **P1** to **P3**). The value **1** of $flag$ marks that $\mathbf{C}_{i-1}$ can see the ending cross-section S_{N+1}. The **Oriented Drilling Process** is an algorithm employed for Partition 3, which returns to the next value of the $\mathbf{C}_i$ series by the intersection point between the longest-length-of-sight direction and the next cross-section.

Algorithm 2: Proposed method.

```
   Input: $\mathfrak{P}, \mathfrak{Q}$, and $r_{[i][j]}, \forall i \in \{1, \ldots, N\}, \forall j \in \{1, \ldots, M\}$.
   Output: $C_i, \forall i \in \{0, \ldots, N+1\}$.
   // Initialisation
1  $C_0 \leftarrow \mathfrak{P}, C_{N+1} \leftarrow \mathfrak{Q}, C_i \leftarrow \emptyset, \forall i \in \{1, \ldots, N\}$;
2  $flag \leftarrow 0$ ;                    // Marking if the $C_{i-1}$ can see $S_{N+1}$ or not
   // Loop Process
3  for $i \leftarrow 1$ to $N+1$ do
4      if $C_{i-1}$ can see $C_{N+1}$ then
           // $C_{i-1}$ is belong to Partition 1
5          $C_k \leftarrow \overline{C_{i-1}C_{N+1}} \cap S_k, \forall k \in \{i, \ldots, N\}$;
6          $flag \leftarrow 1$;
7          break;
8      else
9          $\theta \leftarrow +\infty$ ;       // Angle between the correct direction and $\overrightarrow{C_{i-1}.\mathfrak{Q}}$
10         for $j \leftarrow 1$ to $M$ do
11             $C_{distal} \leftarrow r_{[N+1][j]}$ ;       // Temporary examined vertex of $S_{N+1}$
12             if $C_{i-1}$ can see $C_{Distal}$ then
13                 $flag \leftarrow 1$ ;                   // $C_{i-1}$ is belong to Partition 2
14                 $C_{temp} \leftarrow \overline{C_{i-1}C_{distal}} \cap S_i$ ;      // Possible value for $C_i$
15                 $\theta_{temp} = Angle\left(\overrightarrow{C_{i-1}.C_{distal}}, \overrightarrow{C_{i-1}.\mathfrak{Q}}\right)$ ;    // Possible value for $\theta$
16                 if $\theta > \theta_{temp}$ then
17                     $\theta \leftarrow \theta_{temp}$;
18                     $C_i \leftarrow C_{temp}$;
19                 end
20             end
21         end
22         if $flag = 0$ then
               // $C_{i-1}$ is belong to Partition 3
23             $C_i$ = Oriented Drilling Process($C_{i-1}$);
24         end
25     end
26 end
```

4.3. Oriented Drilling Process

To effectively determine the longest length of sight from an arbitrary point $\boldsymbol{X}$ inside the tube, we employ Lemma 2. The operation scheme of finding the longest-length-of-sight direction illustrated in Figure 9 comprises a series of expanding and deepening processes. Without loss of generality, we assume that $\boldsymbol{C}_{i-1}$ can see a point $\boldsymbol{T}$ in a forward section S_j. By employing Lemma 2, we discretely expand the examined direction from the direction passing through $\boldsymbol{T}$ to others passing through its nearby nodes on the same mesh S_j until discovering a point $\boldsymbol{T}_{new}$ in a farther disk S_k $(k > j)$. As a deepening process, we then update $\boldsymbol{T}$ by $\boldsymbol{T}_{new}$ and also the examined cross-section S_j by S_k. The operation is then repeated until the expanding process is over. The condition to stop the expanding process is when the boundary of the visited area in S_j just comprises the invisible nodes and the boundary points of S_j. Finally, we compare the length of sight corresponding to all visible points in the farthest visible cross-section and find the longest one. The next correct point $\boldsymbol{C}_i$ is the intersection point between this line of sight and the next cross-section S_i.

One advantage of this method is the fact it represents a significant improvement in computation time as we do not need to visit all the nodes of the graph. For the expanding process, on the examined disk S_j, we just need to expand the investigated nodes until the current exact point $\boldsymbol{C}_{i-1}$ can see farther; then we jump further into the more in-depth cross-section. Otherwise, if there is no new disk observed by $\boldsymbol{C}_{i-1}$, we stop the process and indicate the next correct point of the ESP $\boldsymbol{C}_i$. Moreover, as this is an inheritable algorithm, in the next searching process, we can directly use the previous correct direction as the initial examined orientation. Thus, we skip the disks that were examined in the previous loop. For several circumstances such as points on the straight-line segments of the shortest path (the ESP consists of straight-line and geodesic curve segments, see Theorems 1 and 2), the next searching process can stop right after choosing this initial examined orientation. That is also the reason why we call this method Oriented Drilling. Imagine that every time we find the correct direction for point $\boldsymbol{C}_{i-1}$ such as when we drill a hole in that direction. For the next searching process, as there was already a hole, the searching is simplified. The whole process becomes an adjustment the direction of the drill so that it can drill deeper. Consequently, the drilling direction will be oriented closer and closer to the deepest drill hole (the longest length of sight).

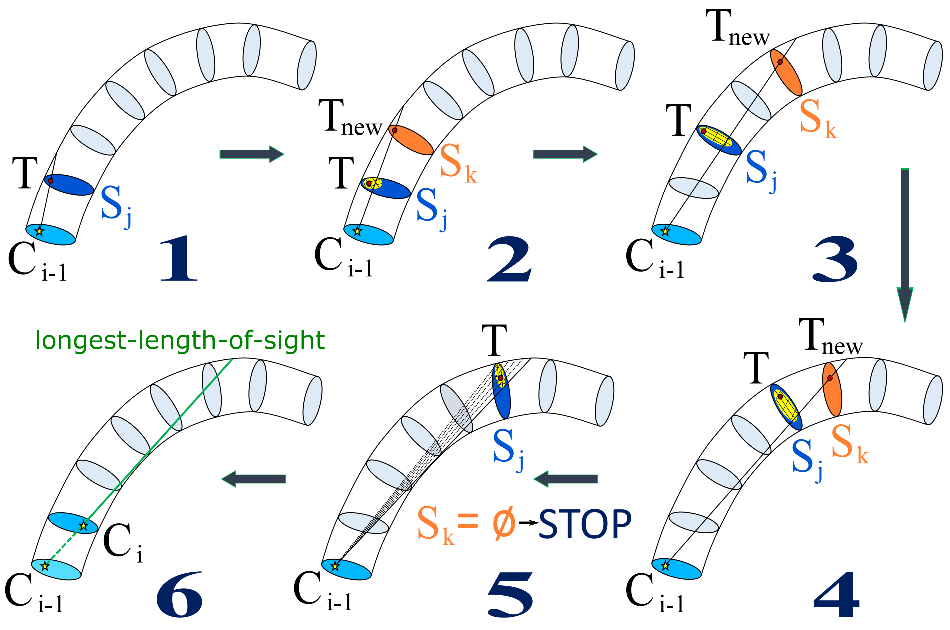


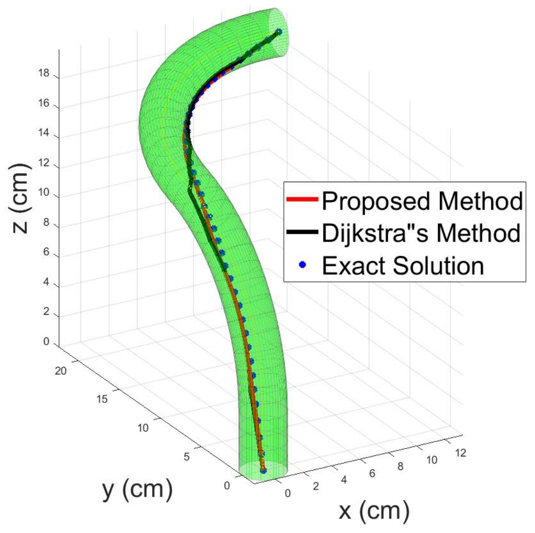
Figure 9. The oriented drilling process : (1) $\boldsymbol{C}_{i-1}$ can see $\boldsymbol{T}$ in section S_j, (2) expand the examined direction in the vicinity of $\boldsymbol{T}$ until seeing $\boldsymbol{T}_{new}$ in section $S_k (k > j)$, (3) update $\boldsymbol{T}$ by $\boldsymbol{T}_{new}$, S_j by S_k and repeat step 2 for the new $\boldsymbol{T}$ and S_j, (4) repeat step 3, (5) the expanding process is over and we do not find any farther section $S_k = \emptyset$, and we then compare all the length of sight passing through the visible area in S_j to obtain the direction corresponding to the longest length of sight, and (6) initialize the next correct point of the shortest path $\boldsymbol{C}_i$ as the intersection point between the correct direction and cross-section S_i.

5. Computational Results

In this section, we will compare the efficiency of the proposed algorithm with Dijkstra's one. There are several criteria for this comparison result: the length of the obtained ESP, the computation speed, the smoothness, and the position error of the solution. The experiments were ran on a machine with an Intel Core i5-8400 CPU @ 2.80 GHz processor. It has a six-core CPU and the available RAM was 16 GB. All algorithms were implemented in Matlab. The code is available online at https://github.com/nguyengiathuongphai/ESP_Tubular.git (accessed date: 18 Feburary 2022).

5.1. Computation Time

We firstly implemented them considering a tube with the centerline in 3D space consisting of a 4 cm straight length and two curved segments belonging to two perpendicular planes. The radii of both curves are 12 cm and their lengths are 16 cm and 20 cm, respectively, as detailed in Figure 10. The inner diameter of the tube is 3 cm. We chose the discretization step h = 2 mm (N = 199). Each meshed disk is made by dividing the cross-section into 25 concentric circles ($N_\rho = 25$) whose circumference is divided into four equal arcs ($N_\theta = 4$).



L.P = 36.63 (mm), L.D = 37.24 (mm), L.E = 36.59 (mm)

T.P = 759 (ms), T.D = 4103 (ms)

Figure 10. Tubular space with two curved segments in space. By **L.P**, and **T.P**, we denote the path length and the computation time for the solution obtained by the proposed method. Similarly, **L.D**, and **T.D** for Dijkstra's algorithm. The exact solution is determined by using Dijkstra's method with $N_\rho = 144$ and $N_\theta = 64$.

As shown in Figure 10, the proposed method enables obtaining a shorter and smoother solution than Dijkstra's method with the same mesh (a detailed analysis will be provided in the next sub-section). Another advantage of the proposed algorithm compared to Dijkstra's method is the computation speed as a large number of unimportant vertices and arcs can be ignored in the process (see Figure 10). As the time complexity of the proposed method has a huge variation depending on the specific shape of the tubular space, the computation time (instead of the theoretical time complexity) will be considered for the comparison result. Table 1 shows how the computation times of the two methods depend on the number of nodes in the meshed circular disks. As we can see, the computation time of Dijkstra's method will increase by a factor of 4 if M is doubled (M is the number of nodes in a meshed disk). This is consistent with the time complexity $O(|A|)$ of Dijkstra's algorithm ($|A| = 2M + (N-1)M^2$). For the proposed method, this increasing rate is less than two.

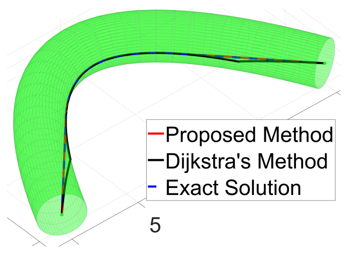
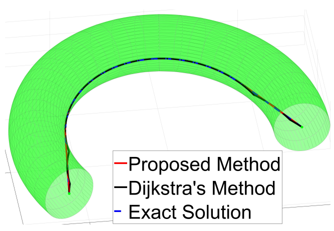
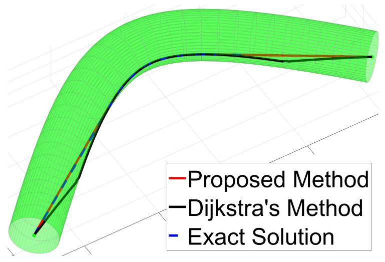
Table 1. Comparison result in computation time of the two methods with different meshes.

$M = 100$	$N_\rho = 25, N_\theta = 4$ T.P = 0.76, T.D = 4.10	
$M = 200$	$N_\rho = 50, N_\theta = 4$ T.P = 0.96, T.D = 16.66	$N_\rho = 25, N_\theta = 8$ T.P = 1.05, T.D = 15.77
$M = 400$	$N_\rho = 100, N_\theta = 4$ T.P = 1.63, T.D = 64.67	$N_\rho = 25, N_\theta = 16$ T.P = 1.51, T.D = 60.19
$M = 800$	$N_\rho = 200, N_\theta = 4$ T.P = 3.03, T.D = 259.23	$N_\rho = 25, N_\theta = 32$ T.P = 2.01, T.D = 219.82

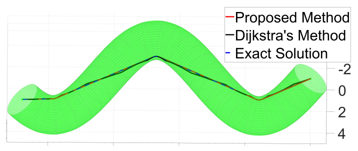
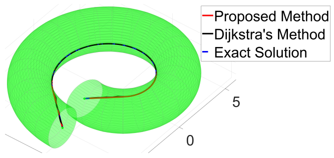
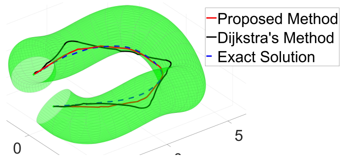
5.2. Accuracy and Smoothness

In the following, we extend the comparison results between the two algorithms for different types of tubular spaces as shown in Table 2. Depending on the properties of the centerline, we have two main classes of the tubular spaces: in plane centerline (parabolic, elliptical, hyperbolic, sinusoidal, and evolvent of a circle); and in space centerline (wave-shaped torus on a sphere, helical, spiral, and complex shape). Each meshed disk is chosen with $N_\rho = 25$ and $N_\theta = 4$. In all these cases, the proposed algorithm always gives shorter, smoother and faster results than Dijkstra's algorithm with the same mesh. Unlike conventional graph-based methods (e.g., Dijkstra's searching algorithm) in which the shortest path is made up of the graph nodes, the proposed method allows finding each correct point on the ESP by determining the intersection point between the exact moving direction (line of sight) and the next cross-section. This intersection point is not necessarily a node of the mesh and leads to a smoother and shorter solution than that by Dijkstra's algorithm. The smoothness of this path is important, especially in mechanical applications when the derivatives of the path with respect to the arc length s of the tube is required such as using the coupled Cosserat rod and string model [38] to find the deformation of a flexible tendon drive robot in the case that the tendon locates freely inside the tube [18].

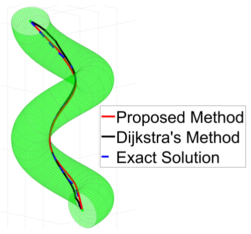
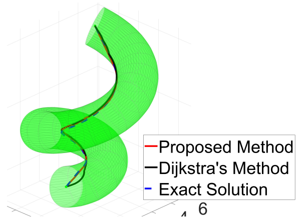
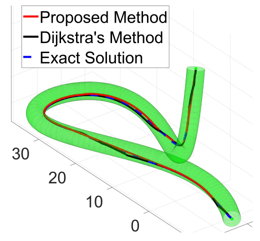
Table 2. Compare the proposed method and Dijkstra's method with many tubular surfaces.

1. Plane Parabolic Centerline	2. Plane Elliptical Centerline	3. Plane Hyperbolic Centerline
		

L.P = 27.22, L.D = 27.29, L.E = 27.21 (cm)
T.P = 0.68, T.D = 2.57 (s)
	L.P = 25.70, L.D = 25.74, L.E = 25.68 (cm)
	T.P = 1.02, T.D = 2.65 (s)
		L.P = 27.97, L.D = 28.10, L.E = 27.96 (cm)
		T.P = 0.73, T.D = 2.64 (s)

4. Plane Sinusoidal Centerline	5. Plane Evolvent of a Circle	6. Wave-Shaped Torus on a Sphere
 Proposed Method Dijkstra's Method Exact Solution -2 0 2 4	 Proposed Method Dijkstra's Method Exact Solution 5 0	 Proposed Method Dijkstra's Method Exact Solution 0 5

L.P = 24.41, L.D = 24.53, L.E = 24.39
T.P = 1.36 (s), T.D = 2.66
	L.P = 21.92, L.D = 21.94, L.E = 21.91
	T.P = 1.16 (s), T.D = 2.57
		L.P = 22.29, L.D = 25.12, L.E = 21.96
		T.P = 1.26 (s), T.D = 2.74

7. Tubular Helical Surface	8. Tubular Spiral Surface	9. Complex Shape Tubular Surface
		 Proposed Method Dijkstra's Method Exact Solution 30 20 10 0

L.P = 20.05, L.D = 20.31, L.E = 20.01
T.P = 1.00 (s), T.D = 2.74
	L.P = 18.87, L.D = 19.02, L.E = 18.71
	T.P = 1.17 (s), T.D = 2.88
		L.P = 110.96, L.D = 111.22, L.E = 110.92
		T.P = 5.90 (s), T.D = 11.29

In addition to the length and the smoothness of the obtained ESP, its location inside the tube is also very important. For example, in the mechanical problem just mentioned above, the tendon location is directly related to the deformation direction of the tube. Thus, the position error of the obtained ESP to the exact solution needs to be investigated. We consider the ESPs given by Dijkstra's and the proposed algorithms to be a series of points located on the cross-sections of the tube. Then, the position error of each point is the distance between itself and the exact solution within the containing cross-section. Let ϵ_i^D and ϵ_i^P be the position errors within cross-section S_i of the solution by Dijkstra's algorithm and by the proposed method, respectively. In this test, we expect to consider the relative errors instead of the absolute ones. As the obtained paths must be inside the tubular space, to limit the relative errors by 100%, we compare the absolute position error to the inner diameter of the tube d. The root mean square error ($RMSE$) and the maximum error (E_{max}) of Dijkstra's solution are given in Equations (5) and (6) (the same for the proposed method just by replacing super index D by P):

$$RMSE^D = \frac{1}{d}\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\epsilon_i^D\right)^2} \tag{5}$$

$$E_{max}^D = \min_{i\in\{1,\ldots,N\}}\left(\frac{\epsilon_i^D}{d}\right) \tag{6}$$

Here, we do not consider the two ending cross-sections (S_0 and S_{N+1}) as the position error is obviously zero at the source and the destination. As shown in Table 3, the proposed method always provides smaller $RMSE$ and E_{max} than those obtained by Dijkstra's algorithm for all of the tubes. Concretely, the average values among these tubes of $RMSE$ and of E_{max} for the proposed solution are, respectively, 0.319% and 1.427% and approximately six times smaller than those given by Dijkstra's algorithm (2.133% and 8.753%, respectively). As the path obtained by Dijkstra's method must pass through nodes of the weighted graph, its position errors significantly depend on the meshing. These errors can be reduced if we increase the granularity of the mesh, but it will also increase the computation time. For the proposed method, the location of the obtained path is not forced to be the nodes of the graph that leads to smaller position errors.

Table 3. Root mean square and maximum position errors of the ESP obtained by the two algorithm. The tube number is as given in Table 2.

					RMSE					
Tube	1	2	3	4	5	6	7	8	9	Avg.
Dijkstra's algorithm	0.506%	0.032%	1.922%	0.118%	0.007%	12.487%	1.655%	1.334%	1.140%	2.133%
Proposed algorithm	0.002%	0.002%	0.004%	0.009%	0.001%	1.003%	0.510%	0.368%	0.974%	0.319%
					Maximum Error					
Tube	1	2	3	4	5	6	7	8	9	Avg.
Dijkstra's algorithm	4.343%	0.628%	11.812%	1.136%	0.226%	28.121%	8.556%	11.942%	12.017%	8.753%
Proposed algorithm	0.012%	0.016%	0.026%	0.105%	0.015%	4.774%	2.077%	2.039%	3.782%	1.427%

To extend the comparison results for different granularities of the meshed disks, Tube 6 (wave-shaped torus on a sphere) and Tube 7 (helical tubular surface) will be considered. The mesh is modified using different N_ρ and N_θ as shown in Table 4. In these tests, the path length given by the proposed method are shorter than that proposed by Dijkstra's algorithm for most cases. As a consequence, the proposed method also provides solutions with smaller position errors in comparison with Dijkstra's algorithm. Even so, for the mesh with high granularity ($N_\rho = 20, N_\theta = 256$), Dijkstra's algorithm can give a more accurate result than that obtained by the proposed method but it requires a much longer computation time. The accuracy of Dijkstra's solution increases with the increase in N_ρ and N_θ while this is not always true with the proposed method. The reason is that a sparse grid can effectively constrain the movement of the VP in the correct direction in some cases. The proposed solution may become more wavy with a higher granular mesh. However, the length of the obtained paths only fluctuate in a narrow range ($\pm$0.5% for Tube 6) and ($\pm$0.1% for Tube 7). Eventually, when the number of nodes on the mesh is higher, the increasing rate in the computation time of Dijkstra's algorithm is superior to that of the proposed method.

Table 4. Comparison results between the proposed algorithm and Dijkstra's algorithm with different granularities of the meshed cross-sections of the tube.

		$N_\rho = 5$				$N_\rho = 20$			
		$N_\theta = 4$	$N_\theta = 16$	$N_\theta = 32$	$N_\theta = 64$	$N_\theta = 4$	$N_\theta = 16$	$N_\theta = 32$	$N_\theta = 256$
Tube 6	L.P (cm)	22.28	22.35	22.29	22.27	22.32	22.37	22.44	22.26
	L.D (cm)	25.83	25.83	25.83	23.74	25.18	25.18	25.18	22.05
	RMSE^{P} (%)	1.05	1.34	0.77	0.71	1.13	1.70	1.72	0.89
	RMSE^{D} (%)	13.58	13.58	13.58	2.90	12.00	12.00	12.00	0.30
	T.P (s)	0.45	0.78	1.18	1.82	0.86	1.70	2.73	17.01
	T.D (s)	0.16	1.54	5.70	22.00	1.66	21.38	82.90	5582.00
Tube 7	L.P (cm)	20.06	20.04	20.04	20.08	20.05	20.05	20.04	20.08
	L.D (cm)	20.82	20.82	20.82	20.82	20.32	20.32	20.32	20.09
	RMSE^{P} (%)	0.62	0.14	0.30	0.73	0.55	0.17	0.20	0.59
	RMSE^{D} (%)	2.92	2.92	2.92	2.92	1.61	1.61	1.61	0.06
	T.P (s)	0.31	0.60	0.85	1.34	0.66	1.15	1.67	7.67
	T.D (s)	0.15	1.52	5.58	21.82	1.63	21.89	84.34	5585.00

6. Discussion

In this section, the extended application scope of the proposed algorithm and the ability to apply it as a reactive method for the navigation problem in unknown environments will be discussed.

6.1. Extended Applications

We can extend the application scope of the proposed method for general tunnels with convex and variable cross-sections (see Figure 11). Indeed, with a minor modification on Remark 1 for points in **P3**, the correct direction is towards the (only) visible point of the farthest visible cross-section instead of considering the longest length of sight, one can confirm that the correctness of Remark 1 will still be preserved (see the Appendix A.2).

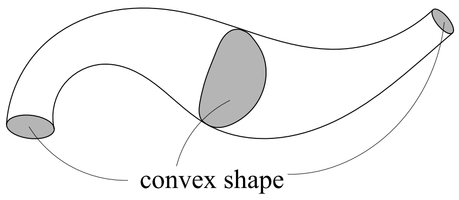


Figure 11. Canal space with convex and variable cross-sections.

6.2. A Reactive Method

In this work, we used the same directed graph for Dijkstra's algorithm and the proposed algorithm for the aim of simplifying the validation and the comparison results. It is important to note that the proposed method does not require knowledge of the entire volume Ω to obtain a weighted graph before searching. In fact, the correct direction of the particle can be determined based on the observation in front of it. While using Dijkstra's algorithm, we cannot determine which path is the ESP until visiting all nodes and arcs of the graph; since we need to store all possible paths during the operation, the proposed method enables us to directly generate the motion decision during the movement there by the ESP is gradually traced by the moving path of the VP. Thus, it can be applied as a reactive method for robots that need to explore unknown tubular spaces such as lava tubes on an astronomical object [39] or environments in the absence of GPS signals [40]. In practice, the proposed algorithm should be run together with a given safety boundary constraint to prevent collision with inspection robots.

7. Conclusions

In this paper, we presented a novel algorithm for solving the ESP problem inside tubular spaces based on its geometric properties. Computational results were obtained for various types of tubular spaces. We demonstrated that the achieved efficiency of the proposed algorithm is better than that of Dijkstra's algorithm. Concretely, the proposed method provided smoother and more precise results with a faster calculation speed than one obtained by Dijkstra's algorithm with the same grid. The strength of the proposed method is also reflected in the fact that it can work without knowing the environment in advance, which allows it to process as a reactive method. Even though the algorithm was described for the tubular space, it is also strongly promising for more complex tunnel spaces to which it can directly be applied with the aforementioned minor modification. A limitation of this method is that it is only applicable to unbranched tubular spaces. In order to apply this method for a branched tubular space, additional information will be required to make decisions at the junctions of branches.

As the ESP may lie on the tubular surface, the requirement of using a collision-free method together with the proposed algorithm has been left for future research. Our plans for future work concern some applications such as the online trajectory generation of navigation robots in unknown tunnels or determine the deformation of a tendon-driven tube-like robot in medical applications, which are also included in our domain of interest.

Author Contributions: Conceptualization, D.-V.-A.N. and K.R.; methodology, D.-V.-A.N. and K.R.; software, D.-V.-A.N.; validation, D.-V.-A.N., J.S. and K.R.; mathematical proofs, D.-V.-A.N., J.S. and K.R.; investigation, D.-V.-A.N., J.S. and K.R.; resources, D.-V.-A.N., J.S. and K.R.; data analysis, D.-V.-A.N., J.S. and K.R.; writing—original draft preparation, D.-V.-A.N., J.S. and K.R.; writing—review and editing, D.-V.-A.N., J.S. and K.R.; funding acquisition, J.S. and K.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the regional BFC project CoErCIVe, the EIPHI Graduate School (contract "ANR-17-EURE-0002") and ANR μRoCS (contract "ANR-17-CE19-0005").

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A

Appendix A.1. Proof of Lemma 2

By S_X, we denote the cross-section of Ω that contains $\boldsymbol{X}$. Let Ω_m and L_m be the sub-space of Ω limited between S_X and S and its length along the centerline curve, respectively. Under a discrete point of view, Ω_m can be considered as a series of $(K+1)$ cross-sections perpendicular to the centreline curve: $S_X = S_0^m, \dots, S_K^m = S$ $(K \in \mathbb{N}^+)$ with the discrete step $\Delta h = \frac{L}{K}$. Let $\sigma_X(S)$ be the visible area of the cross-section S by $\boldsymbol{X}$, we then have:

$$\forall \boldsymbol{Y} \in \sigma_X(S), \forall i \in \{0, \dots, K\} \Rightarrow \exists \boldsymbol{a}_i = \left(\overline{\boldsymbol{XY}} \cap S_i^m \right) \neq \emptyset$$

Therefore, $\boldsymbol{Y}$ is the perspective projection of $\boldsymbol{a}_i$ $(\forall i \in \{0, \ldots, K\})$ from the view point $\boldsymbol{X}$ to the view plane S, hence:

$$\sigma_X(S) \subset \bigcap_{i=0}^{K} P_X^S(S_i^m)$$

where $P_X^S(S_i^m)$ is the perspective projection of S_i^m from the view point $\boldsymbol{X}$ to the view plane S. If $K \to \infty$, or $\Delta h \to 0$, then the problem becomes continuous:

$$\sigma_X(S) \subset \bigcap_{i=0}^{\infty} P_X^S(S_i^m) \tag{A1}$$

Inversely:

$$\forall \boldsymbol{W} \in \bigcap_{i=0}^{\infty} P_X^S(S_i^m) \Rightarrow \left(\overline{\boldsymbol{XW}} \cap S_i^m\right) = \boldsymbol{b}_i \neq \emptyset, \forall i \in \mathbb{N} \tag{A2}$$

When $\Delta h \to 0$, then we obtain:

$$\overline{\boldsymbol{XW}} = \bigcup_{i=0}^{\infty} \boldsymbol{b}_i \subset \Omega$$

Indeed, if $\overline{\boldsymbol{XW}} \not\subset \Omega$, we can always find a value $\Delta h > 0$ in order to have a cross-section S_i^m so that $S_i^m \cap \overline{\boldsymbol{XW}} = \emptyset$ (conflict with (A2)). Consequently, $\boldsymbol{W}$ can be seen by $\boldsymbol{X}$, then we have:

$$\boldsymbol{W} \in \sigma_X(S) \Rightarrow \bigcap_{i=0}^{\infty} P_X^S(S_i^m) \subset \sigma_X(S) \tag{A3}$$

From (A1) and (A3), then:

$$\sigma_X(S) = \bigcap_{i=0}^{\infty} P_X^S(S_i^o) \tag{A4}$$

As the cross-section of Ω is convex and the convexity is preserved under perspective projection and intersection [41], then $\sigma_X(S)$ is a convex region. (Q.E.D.).

Appendix A.2. Proof of Remark 1

We will prove the correctness of the proposed direction of the VP at each partition.

i. Case 1: $\boldsymbol{X} \in$ **P1** *(**X** can see* $\boldsymbol{\Omega}$*)*

As the line segment joining $\boldsymbol{X}$ and $\boldsymbol{\Omega}$ is the shortest path between them, the direction of the ESP $\dot{\boldsymbol{p}}(s)$ at $\boldsymbol{X}$ must be towards $\boldsymbol{\Omega}$:

ii. Case 2: $\boldsymbol{X} \in$ **P2** *(**X** can see S_{end}, but* $\boldsymbol{\Omega}$*)*

Let $\boldsymbol{Y} \in \sigma_X(S_{end})$ be the set of visible points on the ending cross-section such that the angle between $\boldsymbol{XY}$ and $\boldsymbol{X\Omega}$ is the smallest one. We define a cone surface (C_0) with the apex $\boldsymbol{X}$ and the generatrix makes an angle $\widehat{\boldsymbol{YX\Omega}}$ to the axis $\boldsymbol{X\Omega}$, then $\boldsymbol{Y} \in (\sigma_X(S_{end}) \cap C_0)$ (see Figure A1). As $\sigma_X(S_{end})$ is convex, we can easily prove that the existence of $\boldsymbol{Y}$ is unique, moreover $\boldsymbol{Y} \in (\partial\sigma_X(S_{end}) \setminus \partial\Omega)$. Thus, $\boldsymbol{XY}$ must be tangent to $\partial\Omega$ at $\boldsymbol{T}$. Let (α) be the corresponding tangent plane, then we obtain that (α) is also the tangent plane of $\sigma_X(S_{end})$ (see the proof of Corollary 2 for a similar case).

As $\boldsymbol{Y}$ is the tangent point between $\sigma_X(S_{end})$ and $(C_0 \cap S_{end})$ (these two convex sets have only one common point $\boldsymbol{Y}$), (α) is also the tangent plane of (C_0). Let $\boldsymbol{I}$ be the center of the cross-section at $\boldsymbol{T}$. As $\boldsymbol{IT} \perp (\alpha)$, $\boldsymbol{IT}$ must intersect the axis $\boldsymbol{X\Omega}$ of (C_0). Thus, $\boldsymbol{X}, \boldsymbol{T}, \boldsymbol{I}, \boldsymbol{Y}$, and $\boldsymbol{\Omega}$ are coplanar. We denote this coplanar plane by (P_c).

Let $\boldsymbol{W}$ be the intersection between the ending cross-section plane $\beta(S_{end})$ and $\dot{\boldsymbol{p}}(s_X)$. Now, we have to prove that $\boldsymbol{W} \equiv \boldsymbol{Y}$. Using Corollary 2, we obtain: $\boldsymbol{W} \in \sigma_X(S_{end})$. Let

(C_1) be the closed surface enclosed by $\sigma_X(S_{end})$ and the set of line segments from $\boldsymbol{X}$ to every point of $\partial\sigma_X(S_{end})$. Thus, $\boldsymbol{X}$ can see every point in (C_1). If $\boldsymbol{W} \notin \partial\sigma_X(S_{end})$ (that is, $\boldsymbol{W}$ belongs to the inner zone of $\sigma_X(S_{end})$), then the ESP goes into the inner space of (C_1) with the direction $\dot{\boldsymbol{p}}(s_X)$. As $\boldsymbol{\Omega}$ is outside (C_1), the ESP must pass the boundary of (C_1). We denote $\boldsymbol{H}$ as the passing point. Since $\boldsymbol{X}$ can see $\boldsymbol{H}$, the part of the ESP connecting $\boldsymbol{X}$ and $\boldsymbol{H}$ is not the shortest path (as it is longer than $\overline{\boldsymbol{XH}}$). This leads to a contradiction with Lemma 1. Hence:

$$\boldsymbol{W} \in \partial\sigma_X(S_{end}) \tag{A5}$$

In addition, if $\boldsymbol{W} \not\equiv \boldsymbol{Y}$, then $\boldsymbol{W} \notin (P_c)$. By using Corollary 1, we can confirm that the particle will move far away from (P_c) so it cannot reach $\boldsymbol{\Omega}$ on (P_c). Thus, $\boldsymbol{W} \equiv \boldsymbol{Y}$. (Q.E.D.)

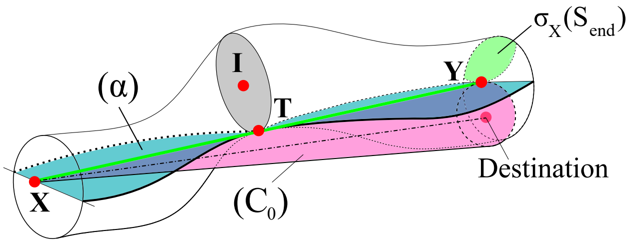


Figure A1. $\boldsymbol{X}$ can see the ending cross-section. By defining the cone surface (C_0), we can prove that $\boldsymbol{\Omega}$ is coplanar with $\boldsymbol{X}, \boldsymbol{I}, \boldsymbol{Y}$.

iii. Case 3: $\boldsymbol{X} \in$ **P3** (*$\boldsymbol{X}$ cannot see S_{end}*)

As $\boldsymbol{X}$ cannot see S_{end}, the farthest cross-section of the tube is S_f which can be seen by $\boldsymbol{X}$. We will prove that $\boldsymbol{X}$ can see only one point in this cross-section. In S_f, if there exist two different visible points $\boldsymbol{Y_1}$ and $\boldsymbol{Y_2}$ by $\boldsymbol{X}$, then $\boldsymbol{X}$ can see the midpoint $\boldsymbol{Y_m}$ of $\overline{\boldsymbol{Y_1Y_2}}$ (using Lemma 2). As $\boldsymbol{Y_m} \notin \partial\Omega$, we infer that S_f is not the farthest visible cross-section by $\boldsymbol{X}$ ($\boldsymbol{X}$ can see farther with the line of sight through $\boldsymbol{Y_m}$). Thus, there is only one visible point $\boldsymbol{Y}$ in S_f that can be seen by $\boldsymbol{X}$, and $\boldsymbol{XY}$ is the correct direction of the tendon according to Corollary 2.

Moreover, we can demonstrate that $\boldsymbol{XY}$ is also the longest length of sight from $\boldsymbol{X}$. One can easily confirm that $\boldsymbol{XY}$ must be tangent to $\partial\Omega$ at a point $\boldsymbol{T}$ of the cross-section S_T. Let (α) be the corresponding tangent plane. Let Ω_v be the space enclosed by $\partial\Omega$, (α), and the cross-section containing $\boldsymbol{X}$ as illustrated in Figure A2. Then, Ω_v contains all the visible points by $\boldsymbol{X}$ of Ω located behind the cross-section S_T. The problem now is to prove that $\overline{XY}$ is the longest length of sight in Ω_v. As the tube does not overlap itself, we obtain: $\boldsymbol{XY} \geq \boldsymbol{TY} \geq 2R$. Thus, one can confirm that Ω_v is totally contained by the sphere (χ) center $\boldsymbol{X}$ and the radius $\overline{\boldsymbol{XY}}$. We then have $\overline{XY}$ as the longest length of sight from $\boldsymbol{X}$.

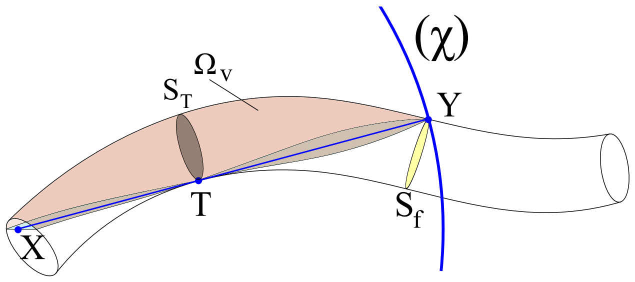


Figure A2. $\boldsymbol{X}$ cannot see the ending cross-section. It can only see one point $\boldsymbol{Y}$ on the farthest visible cross-section S_f.

It is evident that every point on the ESP must belong to one of the three partitions (**P1**: see $\boldsymbol{\Omega}$; **P2**: see S_{end}, but not see $\boldsymbol{\Omega}$; and **P3**: not see S_{end}) and as the correct direction is unique for each position, if the VP follows the proposed correct direction throughout its journey, its moving path will describe the ESP. (Q.E.D).

References

1. Li, F.; Klette, R. Euclidean shortest paths. In *Euclidean Shortest Paths*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 3–29.
2. Özaslan, T.; Shen, S.; Mulgaonkar, Y.; Michael, N.; Kumar, V. Inspection of penstocks and featureless tunnel-like environments using micro UAVs. In *Field and Service Robotics*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 123–136.
3. Özaslan, T.; Mohta, K.; Keller, J.; Mulgaonkar, Y.; Taylor, C.J.; Kumar, V.; Wozencraft, J.M.; Hood, T. Towards fully autonomous visual inspection of dark featureless dam penstocks using MAVs. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4998–5005.
4. Özaslan, T.; Loianno, G.; Keller, J.; Taylor, C.J.; Kumar, V.; Wozencraft, J.M.; Hood, T. Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1740–1747. [CrossRef]
5. Quenzel, J.; Nieuwenhuisen, M.; Droeschel, D.; Beul, M.; Houben, S.; Behnke, S. Autonomous MAV-based indoor chimney inspection with 3D laser localization and textured surface reconstruction. *J. Intell. Robot. Syst.* **2019**, *93*, 317–335. [CrossRef]
6. Petrlík, M.; Báča, T.; Heřt, D.; Vrba, M.; Krajník, T.; Saska, M. A Robust UAV System for Operations in a Constrained Environment. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2169–2176. [CrossRef]
7. Shukla, A.; Karki, H. Application of robotics in onshore oil and gas industry—A review Part I. *Robot. Auton. Syst.* **2016**, *75*, 490–507. [CrossRef]
8. Chataigner, F.; Cavestany, P.; Soler, M.; Rizzo, C.; Gonzalez, J.P.; Bosch, C.; Gibert, J.; Torrente, A.; Gomez, R.; Serrano, D. Arsi: An aerial robot for sewer inspection. In *Advances in Robotics Research: From Lab to Market*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 249–274.
9. Tan, C.H.; Ng, M.; Shaiful, D.S.B.; Win, S.K.H.; Ang, W.; Yeung, S.K.; Lim, H.; Do, M.N.; Foong, S. A smart unmanned aerial vehicle (UAV) based imaging system for inspection of deep hazardous tunnels. *Water Pract. Technol.* **2018**, *13*, 991–1000. [CrossRef]
10. Tan, C.H.; bin Shaiful, D.S.; Ang, W.J.; Win, S.K.H.; Foong, S. Design optimization of sparse sensing array for extended aerial robot navigation in deep hazardous tunnels. *IEEE Robot. Autom. Lett.* **2019**, *4*, 862–869. [CrossRef]
11. Mallios, A.; Ridao, P.; Ribas, D.; Carreras, M.; Camilli, R. Toward autonomous exploration in confined underwater environments. *J. Field Robot.* **2016**, *33*, 994–1012. [CrossRef]
12. Fairfield, N.; Kantor, G.; Wettergreen, D. Real-time SLAM with octree evidence grids for exploration in underwater tunnels. *J. Field Robot.* **2007**, *24*, 03–21. [CrossRef]
13. Gary, M.; Fairfield, N.; Stone, W.C.; Wettergreen, D.; Kantor, G.; Sharp, J.M., Jr. 3D Mapping and Characterization of Sistema Zacatón from DEPTHX (DE ep P hreatic TH ermal e X plorer). In Proceedings of the ASCE 11th Sinkhole Conference (KARST '08), Tallahassee, Florida, 22–26 September 2008; pp. 202–212.
14. Pidic, A.; Aasbøe, E.; Almankaas, J.; Wulvik, A.; Steinert, M. Low-Cost Autonomous Underwater Vehicle (AUV) for Inspection of Water-Filled Tunnels During Operation. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Quebec City, QC, Canada, 26–29 August 2018.
15. Alvarez, A.; Caiti, A.; Onken, R. Evolutionary path planning for autonomous underwater vehicles in a variable ocean. *IEEE J. Ocean. Eng.* **2004**, *29*, 418–429. [CrossRef]
16. Gao, X.Z.; Hou, Z.X.; Zhu, X.F.; Zhang, J.T.; Chen, X.Q. The shortest path planning for manoeuvres of UAV. *Acta Polytech. Hung.* **2013**, *10*, 221–239.
17. Dang, T.; Mascarich, F.; Khattak, S.; Nguyen, H.; Nguyen, H.; Hirsh, S.; Reinhart, R.; Papachristos, C.; Alexis, K. Autonomous search for underground mine rescue using aerial robots. In Proceedings of the 2020 IEEE Aerospace Conference, Big Sky, Montana, 7–14 March 2020; pp. 1–8.
18. Swaney, P.J.; York, P.A.; Gilbert, H.B.; Burgner-Kahrs, J.; Webster, R.J. Design, fabrication, and testing of a needle-sized wrist for surgical instruments. *J. Med. Devices* **2017**, *11*, 014501. [CrossRef] [PubMed]
19. Nguyen, D.-V.-A.; Girerd, C.; Boyer, Q.; Rougeot, P.; Lehmann, O.; Tavernier, L.; Szewczyk, J.; Rabenorosoa, K. A Hybrid Concentric Tube Robot for Cholesteatoma Laser Surgery. *IEEE Robot. Autom. Lett.* **2022**, *7*, 462–469. [CrossRef]
20. Canny, J.; Reif, J. New lower bound techniques for robot motion planning problems. In Proceedings of the 28th Annual Symposium on Foundations of Computer Science (sfcs 1987), Los Angeles, CA, USA, 12–14 October1987; pp. 49–60.
21. Sharir, A.; Baltsan, A. On shortest paths amidst convex polyhedra. In Proceedings of the second annual symposium on Computational Geometry, Yorktown Heights, NY, USA, 2–4 June 1986; pp. 193–206.
22. Gewali, L.P.; Ntafos, S.; Tollis, I.G. Path planning in the presence of vertical obstacles. *IEEE Trans. Robot. Autom.* **1990**, *6*, 331–341. [CrossRef]
23. Agarwal, P.K.; Sharathkumar, R.; Yu, H. Approximate Euclidean shortest paths amid convex obstacles. In Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, New York, NY, USA, 4–6 January 2009; pp. 283–292.
24. Mitchell, J.S. Geometric Shortest Paths and Network Optimization. *Handb. Comput. Geom.* **2000**, *334*, 633–702.
25. Deschamps, T.; Cohen, L.D. Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Med. Image Anal.* **2001**, *5*, 281–299. [CrossRef]
26. Bulow, T.; Klette, R. Digital curves in 3D space and a linear-time length estimation algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 962–970. [CrossRef]
27. Li, F.; Klette, R. Rubberband algorithms for solving various 2D or 3D shortest path problems. In Proceedings of the 2007 International Conference on Computing: Theory and Applications (ICCTA'07), Kolkata, India, 5–7 March 2007; pp. 9–19.

28. Dogan, F.; Yayli, Y. On the curvatures of tubular surface with Bishop frame. *Commun. Fac. Sci. Univ. Ank. Ser. A1 Math. Stat.* **2011**, *60*, 59–69.
29. Chow, S.N.; Lu, J.; Zhou, H.M. Finding the shortest path by evolving junctions on obstacle boundaries (E-JOB): An initial value ODEś approach. *Appl. Comput. Harmon. Anal.* **2013**, *35*, 165–176. [CrossRef]
30. Elmokadem, T.; Savkin, A.V. A method for autonomous collision-free navigation of a quadrotor UAV in unknown tunnel-like environments. *Robotica* **2021**, 1–27. [CrossRef]
31. Moore, E.F. The shortest path through a maze. *Proc. Int. Symp. Switching Theory* **1959**, *1959*, 285–292.
32. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
33. Savkin, A.V.; Hoy, M. Reactive and the shortest path navigation of a wheeled mobile robot in cluttered environments. *Robotica* **2013**, *31*, 323–330. [CrossRef]
34. Hachour, O. The use of the 3D Smoothed parametric curve Path planning for Autonomous mobile robots. *Int. J. Syst. Appl. Eng. Dev.* **2009**, *3*, 105–116.
35. Blaga, P.A. On tubular surfaces in computer graphics. *Stud. Univ. Babes-Bolyai Inform.* **2005**, *50*, 81–90.
36. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
37. Bose, P.; Maheshwari, A.; Shu, C.; Wuhrer, S. A survey of geodesic paths on 3D surfaces. *Comput. Geom.* **2011**, *44*, 486–498. [CrossRef]
38. Rucker, D.C.; Webster, R.J., III. Statics and dynamics of continuum robots with general tendon routing and external loading. *IEEE Trans. Robot.* **2011**, *27*, 1033–1044. [CrossRef]
39. Thangavelautham, J.; Robinson, M.S.; Taits, A.; McKinney, T.; Amidan, S.; Polak, A. Flying, hopping Pit-Bots for cave and lava tube exploration on the Moon and Mars. *arXiv* **2017**, arXiv:1701.07799.
40. am Ende, B. 3D mapping of underwater caves. *IEEE Comput. Graph. Appl.* **2001**, *21*, 14–20. [CrossRef]
41. Boyd, S.; Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.