applied sciences

MDPI

Article

Collision Avoidance for a Selective Compliance Assembly Robot Arm Manipulator Using Topological Path Planning

Josias G. Batista [1,*,†], Geraldo L. B. Ramalho [1,†], Marcelo A. Torres [1], Anderson L. Oliveira [1] and Daniel S. Ferreira [2]

[1] Automation and Industrial Networks Laboratory—Department of Industry, Campus Fortaleza, Federal Institute of Education, Science and Technology of Ceará—IFCE, Fortaleza 60.040-215, CE, Brazil; gramalho@ifce.edu.br (G.L.B.R.); marcelotorres@aluno.ifce.edu.br (M.A.T.); anderson.lima.oliveira05@aluno.ifce.edu.br (A.L.O.)

[2] Department of Computing Science, Campus Maracanaú, Federal Institute of Education, Science and Technology of Ceará—IFCE, Maracanaúe 61.939-140, CE, Brazil; daniels@ifce.edu.br

* Correspondence: josiasbatista@ifce.edu.br

† These authors contributed equally to this work.

Abstract: Industrial applications with robotic manipulators have grown and made production systems increasingly efficient. However, there are still some limitations that can delay production, causing losses. Several factors, such as accidents and collisions of manipulator robots with operators and other machines, can cause unforeseen stops. Thus, this work aims to develop a trajectory planning method to avoid collisions applied to a selective compliance assembly robot arm (SCARA) robotic manipulator in the context of collaborative robotics. The main contribution of this paper is a path planning method based on mathematical morphology, named topological path planning (TPP). Through some evaluation metrics such as the number of path points, computing time, distance, standard deviation of the joint acceleration, and maximum acceleration rate along the path, we show that TPP is a collision-free, deterministic, and predictable route planning. In our experiments, our proposal presented better results for applications in industrial robotic manipulators when compared to the probabilistic roadmap method (PRM) and TPP*, a particular case of TPP that is similar to the generalized Voronoi diagram (GVD).

Keywords: path planning; collision avoidance; topological path; collaborative robotics

check for updates

Citation: Batista, J.G.; Ramalho, G.L.B.; Torres, M.A.; Oliveira, A.L.; Ferreira, D.S. Collision Avoidance for a Selective Compliance Assembly Robot Arm Manipulator Using Topological Path Planning. *Appl. Sci.* **2023**, *13*, 11642. https://doi.org/10.3390/app132111642

Academic Editors: Renato Vidoni, Lorenzo Scalera and Andrea Giusti

Received: 29 September 2023
Revised: 20 October 2023
Accepted: 20 October 2023
Published: 24 October 2023

1. Introduction

Robotics was introduced in industrial environments with the aim of automating processes; that is, facilitating human work. With technological evolution came the use of robotics in the applied industrial environment and, therefore, several types of robots were created with the purpose of assisting or even replacing humans in certain tasks. In this perspective, it is assumed that robots can perform a task with sufficient intelligence to classify what action is necessary to be chosen; for example, to avoid a collision.

The large number of robots in an industrial environment has led, over the years, to the development of various methods with the aim of monitoring and controlling mobile robots or manipulator robots. With this, they acquired the ability to operate in environments that are dangerous for humans, such as, for example, in addition to the Earth's atmosphere, in aquatic explorations, and in the transport of materials, among others [1].

The problem of collision prevention that arises is not only to detect a "border crossing" and not only to make the decision to turn off the robot, but to allow the automation agent to be in the scene, coexisting in a harmonious way, avoiding the collision with other equipment or other robots. This requires the implementation of an algorithm that detects an obstacle in movement and, anticipating a collision avoids it. Classic approaches to collision avoidance path planning are the algorithms based on the artificial potential fields (APFs) [2], generalized Voronoi diagram (GVD) [3], and probabilistic roadmap (PRM) [4].

The APF method requires adjustments of several parameters (gain in the attraction force of the endpoint, repulsion of the starting point and obstacles) so that the generated path is satisfactory so that the target position and path are optimized, i.e., so that a minimum distance is traveled [5]. The PRM has been widely used for smart vehicles, unmanned aerial vehicle applications, and mobile robots. However, there exist some shortcomings, such as a low efficiency, low reuse rate of the roadmap, and a lack of guidance in the selection of sampling points [6].

The generalized Voronoi diagram (GVD) is a powerful method used to create a topological roadmap on a free space of a workspace populated with obstacles [7,8]. The GVD identifies points equidistant from multiple obstacles, including the boundary of space in which an object or agent moves [3,9]. The GVD roadmap is constituted of a network of linear and parabolic segments around obstacles that contain multiple collision-free paths. However, they are often too conservative, meaning that paths stray too far from obstacles, resulting in unnecessary detours.

To increasingly optimize the collision avoidance paths, different approaches were proposed to improve the APF, GVD, and PRM-based algorithms. Within this context, the following research studies can be highlighted [10–21], which used the modified APF and PRM. Some research also used other algorithms, such as computational intelligence, for example, to improve the PRM. One can cite the research from [22], which used the PRM with particle swarm optimization (PSO), and the research from [23], which used the PRM with reinforcement learning. The paper [7] presented a generalized Voronoi diagram (GVD)-based heuristic path planning algorithm to generate a heuristic path, guide the sampling process of rapidly exploring random trees (RRTs) and their variants, and further improve the motion planning efficiency of RRTs.

The research [24] presented an algorithm encountering dynamic obstacles, and an improved velocity potential field (IVPF). The algorithm considers direction factors, obstacle velocity factors, and tangential velocity. Therefore, the IVPF algorithm can avoid obstacles better to ensure the safety of both the human and robot manipulator. The paper [25] designed a pseudo-random sampling strategy with the main spatial axis as the reference axis. The method optimized the generation of sampling points, removed redundant sampling points, set the distance threshold between road points, and adopted a two-way incremental method for collision detections.

The research [26] proposed a novel approach that used the rapidly exploring random tree (RRT)-based scheme for optimizing the robot's motion planning and minimizing energy consumption. Sampling-based algorithms for path planning, such as RRT and many other variants, were widely used in robotic motion planning due to their efficiency in solving complex high-dimensional problems. The paper [27] presented an overview of the different approaches used in the optimal design of parallel manipulators (PMs), as well as the main difficulties encountered. The paper proposed technical solutions to solve the problem of the optimal dimensional design of PMs. In addition, a seven-stage optimal design methodology for PMs was proposed.

The informed RRT* (IRRT*) algorithm is one of the optimized versions of the rapidly exploring random trees (RRT) algorithm and finds near-optimal solutions faster than RRT and RRT* algorithms by restricting the search area to an ellipsoidal subset of the state space. Paper [28] reported a hybrid algorithm by combining the artificial potential field method (APF) with an IRRT* algorithm for mobile robot path planning. By introducing the virtual force field of the APF into the search tree expansion stage of the IRRT* algorithm, the guidance of the algorithm increased, which greatly improved the convergence rate and search efficiency of the IRRT* algorithm. Paper [29] proposed a novel motion planner named RRT*Smart-AD to ensure that the dual-arm robot satisfies obstacle avoidance constraints and dynamic characteristics in dynamic environments.

Contributions and Objectives

Several works have been developed to improve algorithms such as PRM, GVD, and APF. Improvements often use modifications or optimization algorithms to tune parameters for collision avoidance paths. Despite these efforts, the literature lacks approaches that explore homotopy classes of trajectories to estimate a topological path at the SCARA workspace. Furthermore, there is no evidence about the performance of mathematical morphology (MM) in constraining these trajectories. We hypothesize that an MM-based strategy used to thin the regions between the obstacles at the SCARA workspace reaches groups of homotopic trajectories to draw a deterministic topological path toward the robot's target.

This paper proposes an algorithm based on mathematical morphology operations such as dilation and erosion to create topological paths around obstacles, thus planning a collision-free path for an SCARA manipulator. We experimentally show that topological path planning (TPP) is better suited for collision avoidance and trajectory planning for an SCARA robot manipulator.

This paper presents the following contributions:

- A path planning with collision avoidance algorithm based on a mathematical morphology framework.
- The performance evaluation of path planning algorithms based on the dynamics of the acceleration of the manipulator joints.
- A trajectory planning suitable for collaborative manipulators.

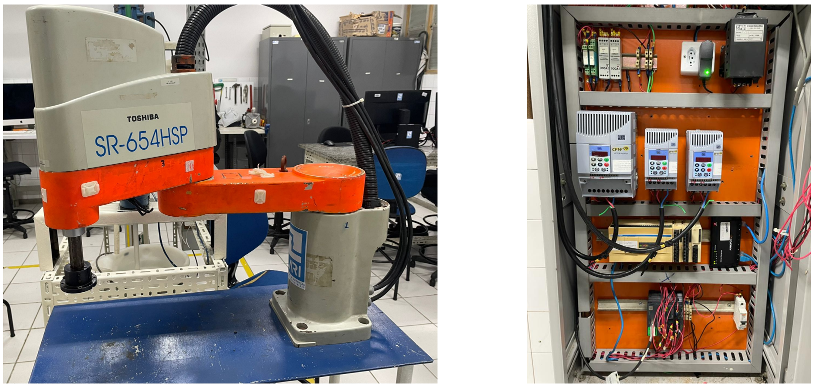
This paper is organized as follows. Section 2 presents the background on the SCARA manipulator and PRM. Section 3 presents the methodology of the proposed method, the scenarios used for the implementation of the algorithms, and the evaluation metrics. Section 4 presents the results of this research. In Section 5, a discussion is carried out about the results. Finally, conclusions and future work are mentioned in Section 6.

2. Theoretical Foundation

In this section, we present the theoretical foundations related to the subjects of this research, along with a brief description of the SCARA manipulator and descriptions of the PRM, a well-known algorithm for path planning with collision avoidance.

2.1. SCARA Manipulator

The SCARA manipulator is a 3-DOF robot shown in Figure 1a. For the sake of simplicity, in this work, our demonstrations use only 2 DOF. The first two joints revolve around the vertical axis (z_1 and z_2) performing together parallel to the horizontal plane X_E Y_E, thus behaving as a 2-DOF planar robot [30,31].



(**a**) (**b**)

Figure 1. (**a**) The SCARA manipulator used for the kinematic modeling and (**b**) its controller and drivers.

The SCARA manipulator used as an experimental study was adapted to operate more openly, as its controller and servomotor drives were damaged; that is, it was no longer working. Therefore, a programmable logic controller (PLC) was used to control the manipulator, and frequency inverter drivers were used to drive the servomotors. Figure 1b presents an idea of the experimental setup of the SCARA manipulator.

We highlight that our SCARA manipulator no longer uses the manufacturer's original drives and controllers, but rather a new configuration adapted for the experiments proposed in this paper.

2.1.1. Forward Kinematics of the SCARA Manipulator

We used the Denavit–Hartenberg (D-H) convention to develop the kinematic model and compute the parameters α, a, d and θ using the manipulator coordinate system [32]. D-H parameters are shown in Table 1.

Table 1. D-H parameters of the SCARA manipulator.

Link	$\boldsymbol{a_i}$	$\boldsymbol{\alpha_i}$	$\boldsymbol{d_i}$	$\boldsymbol{\theta_i}$
1	0.35	0	0.32	θ_1
2	0.30	π	0	θ_2
3	0	0	d_3	0

Applying the transformation matrices and the D-H conversion, the positions in the workspace space can be achieved from the joints space coordinates [33]

$$P_x = 0.35cos(\theta_1) + 0.30cos(\theta_1 + \theta_2) \tag{1}$$

and

$$P_y = 0.35sen(\theta_1) + 0.30sen(\theta_1 + \theta_2). \tag{2}$$

2.1.2. Inverse Kinematics of the SCARA Manipulator

From the equations of direct kinematics (1) and (2) and applying some trigonometric transformations, we find the inverse kinematics equations, given by [33]:

$$\theta_1 = tan^{-1}\left[\frac{P_y(L_1 + L_2cos(\theta_2)) - P_xL_2sen(\theta_2)}{P_x(L_1 + L_2cos(\theta_2)) - P_yL_2sen(\theta_2)}\right] \tag{3}$$

$$\theta_2 = cos^{-1}\left(\frac{P_x^2 + P_y^2 - L_1^2 - L_2^2}{2L_1L_2}\right) \tag{4}$$

where $L_1 = 0.35$ m and $L_2 = 0.30$ m are the length values of each manipulator joint.

Equations (3) and (4) represent the inverse kinematic model of the SCARA manipulator, for the first two DOFs, and will be used to calculate the displacements of the manipulator joints based on the paths generated by the path planning algorithms.

2.2. *Dynamics of an Industrial Manipulator*

Considering the manipulator's kinetic energy, the dynamic equation can be written as follows [31,33]:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau, \tag{5}$$

where $C(\theta, \dot{\theta}) \in \Re^n$ is the matrix that describes the centripetal and Coriolis forces, and $G(\theta) = \frac{\partial g}{\partial \theta} \in \Re^n$ is the gravity vector.

Applying the Lagrange equation and carrying out all calculations and substitutions of terms, we obtain the equations that represent the dynamic model of the manipulator.

The torques of joints 1 and 2 are defined as follows, substituting the values of $l_1 = 0.35$ m, $l_2 = 0.30$ m, $m_1 = 7.872$ kg, $m_2 = 4.277$ kg, and $g = 9.8$ m/s^2:

$$\begin{aligned} \tau_1 = [1.873 + 0.898C_2]\ddot{\theta}_1 + [0.384+ \\ 0.449C_2]\ddot{\theta}_2 - 0.769S_2\dot{\theta}_1\dot{\theta}_2- \\ 0.384S_2\dot{\theta}_2^2 + 12.574C_{12} + 41.671C_1 \end{aligned} \tag{6}$$

and

$$\begin{aligned} \tau_2 = [0.384 + 0.449C_2]\ddot{\theta}_1+ \\ 0.384\ddot{\theta}_2 + 0.449S_2\dot{\theta}_1^2 + 12.574C_{12}. \end{aligned} \tag{7}$$

where S_1 is $sin(\theta_1)$, S_2 is $sin(\theta_2)$, C_1 is $cos(\theta_1)$, C_2 is $cos(\theta_2)$, S_{12} is $sin(\theta_1 + \theta_2)$, and C_{12} is $cos(\theta_1 + \theta_2)$.

The mechanical energy is estimated based on the torques on Equations (6) and (7) for each motor m in joints 1 and 2. The torque is calculated with the expression

$$T_m = 9,55\frac{P_m}{N_m} \tag{8}$$

where T_m is the torque of motors in Nm, P_m is the nominal power of motors in W, and N_m is the value of the motors speed in rpm (min^{-1}). Thus, the mechanical energy E in kWh is computed using

$$E = \sum_{i=1}^{n}(P_{m_i}t_i) \tag{9}$$

where P is the power in W (watts) and t is the time in h (hours).

2.3. Probabilistic Roadmap

The probabilistic roadmap (PRM) algorithm provides a collision avoidance path planning successfully used for mobile robot applications in the presence of obstacles [34]. This method is extremely efficient due to the ability to design paths quickly and the prevalence of the shortest path. In this method, a random sample of the configuration space is initially generated using a uniform probability distribution. In sequence, the algorithm tests the samples for collision. The trajectory is generated from q_{start} to q_{goal} using the points where a collision is not detected [35,36].

The logic of the PRM is based on previously analyzing the robot's trajectory from a predetermined map. Therefore, with this knowledge of the location where the robot will transit, it is possible for it to calculate the route that it will follow. Therefore, the SCARA manipulator will calculate the route effectively, aiming to distance all obstacles along the path from the robot [35].

2.4. Mathematical Morphology

Mathematical morphology (MM) is a framework for analyzing spatial structures through the application of structural transformations [37].

Dilation and erosion are the basic morphological operators from which other operators are derived, namely opening, closing, and thinning. A structuring element B, whose size and shape are chosen a priori, determines the points to be added or removed in a shape A defined in the Euclidean space E. We provide a brief and simple definition of these operations since details of morphological mathematics are beyond the scope of this paper.

The erosion of a shape A is denoted by $A \ominus B = \{x \mid B_x \subseteq A\}$, where $B_x = \{b + x | b \in B\}$ is a translation $b + x$ of the structuring element B. All points where B_x is not completely contained in A are removed. Dilation is the complement of an erosion, and is defined by $A \oplus B = \{x \mid B_x \bigcap A \neq \emptyset\}$. In dilation, points will be added where B_x intersects with A.

The opening of A can be obtained by an erosion followed by a dilation. It is denoted by $A \circ B = (A \ominus B) \oplus B$. As a result, points belonging to the smaller structures of A are eliminated. The closing operation, denoted by $A \bullet B = (A \oplus B) \ominus B$, is the reverse, making the gaps in A decrease.

Skeletonization, shrinking, or thinning are operations guided by influence zones [38], using a combination of basic morphological operators. For the sake of simplicity, a skeleton can be viewed as the successive removal of the points on the edge of the shape, on the condition that they do not break the connectivity of the shape area, thus preserving its original topology.

For better understanding, the skeleton can be defined by the set D of the centers of maximal disks on the 2D grid A. Given a family of structuring elements $\{nB\}$ constructed by successive dilations of a disk B,

$$nB = \underbrace{B \oplus \cdots \oplus B}_{n \text{ times}}, n = 0, 1, \cdots ,$$

exists a disk $nB_c = \{x \in A \mid x - c \in nB\}$ centered in c and translated by $x - c$, $nB_c \in D$, under the condition that, for each point y and an integer m, $mB_y \not\subseteq D$ where $nB_c \subseteq mB_y$. This procedure is illustrated in Figure 2.

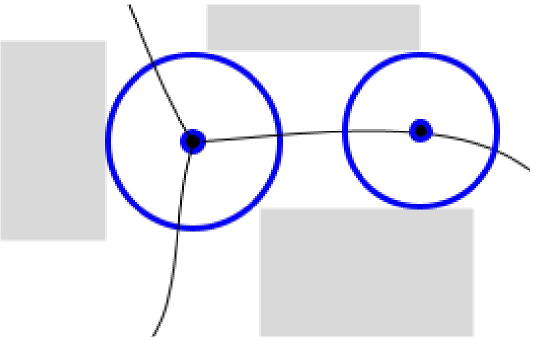


Figure 2. Skeleton (dark line) by maximum disks (blue).

2.5. Shortest Path Search

The Dijkstra algorithm is an algorithm that calculates the minimum cost between the vertices—in this paper, called "points"—the origin point, and the destination point in a graph (G). All points are connected by weighted edges where the weights represent the distances between the points. This algorithm is capable of calculating the minimum cost from the starting point to all other points in the graph.

Its complexity $(O(E \log P))$, where "E" represents the number of edges and "P" the number of points of the algorithm, can be influenced by the number of points and edges present in the graph, the weight of its edges, and the distance between neighbors.

The greater the number of neighbors that a node has, the greater its complexity will be as the algorithm will need to perform a greater number of tests with less weight, thus leading to an increase in execution time.

3. Topological Path Planning (TPP)

This section presents the proposed method, the scenarios, and the evaluation metrics used in performance assessment. We consider the context of collaborative robotics, where different scenarios containing static obstacles can occur.

SCARA manipulator path planning is obtained by connecting a set of points in a global map M of obstacles represented by an $N_1 \times N_2$ grid. The workspace of the SCARA manipulator must be respected and the result path must avoid collisions with the obstacles. It is desired that the robot perform the shortest path possible using smooth movements.

In industrial applications, a collaborative robotic manipulator must achieve three main objectives when moving from q_{start} to q_{goal}: (i) avoid collisions, (ii) minimize the distance traveled, and (iii) perform smooth movements. Thus, the main objective of our algorithm is to avoid obstacles in the manipulator's workspace and to find the shortest path from the initial point q_{start} to the final point q_{goal} that minimizes the rate of acceleration (jerk) during its trajectory.

Our proposal estimates a topological-constrained set of connected points on the free space, the area inside the manipulator workspace that contains no obstacles. These points, represented in a grid, provide many homotopic and non-homotopic topological paths [39]. A topological path roughly corresponds to the mid-distance between obstacles, thus providing a collision-free path for the manipulator. The TPP algorithm creates a topological-constrained set of connected points and finds the shortest topological path in this set. TPP is defined as follows.

3.1. Proposed Method

Consider the Euclidean space represented by an $N_1 \times N_2$ grid G of connected points. The manipulator workspace $W \in G$ contains only the points reachable by the robot arm and a known map of obstacles represented by a grid $M \in G$ containing only the points inside the obstacles. We define the free space $F = \{p \mid p \neq q, p \in W, q \in M\}$ as the set of points belonging to the workspace W and not belonging to the obstacles map M. The shapes numbered 1, 2, and 3 in Figure 3a represent the obstacles. The free space, illustrated in Figure 3b, represents our topological-constrained space of connected points. F can be seen as a finite set of paths between q_{start} and q_{goal}.

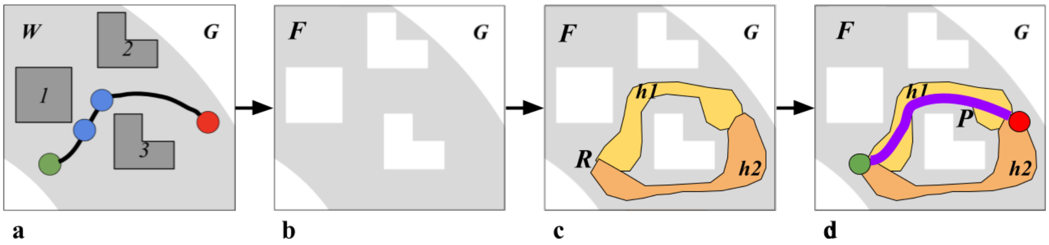


Figure 3. Topological path planning model. (**a**) The environment shows the obstacles (1, 2, and 3), start (green), goal (red), and the optimum path. Blue marks represent critical positions. (**b**) The free space F. (**c**) The subsets of homotopic roadmaps $R = h1 \cup h2$ are achieved by controlled thinning. (**d**) The topological path P is depicted in purple.

Using a controlled thinning, where relaxed influence zone criteria allow non-maximum circles to be present, the TPP algorithm shrinks the free space F until the resulting area is a roadmap $R \in F$ surrounding the obstacles, including larger regions, as shown in Figure 3c. In this paper, the thinning procedure is modulated by distance and a number of obstacles nearby. As a result, our proposal preserves more points between obstacles, as shown in Figure 4. This behavior allows the next step to find shorter paths between q_{start} and q_{goal}.

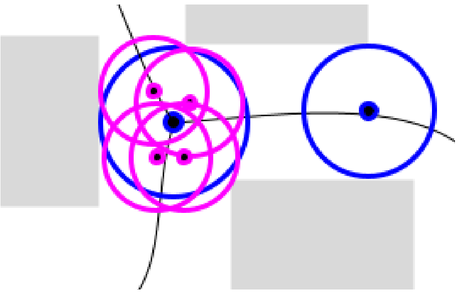


Figure 4. Disk size is modulated by zones of influence driven by distance and obstacle population. Both maximum (blue) and smaller (magenta) disks are allowed in specific regions.

The resulting roadmap $R = h1 \cup h2 \cup \ldots$ is composed by sets of collision-free homotopic paths h, shown in Figure 3c. Finally, a shortest path algorithm finds the connected points $P \subset R$ that represent the best possible topological path inside R. The final path P is smoother and minimizes the unnecessary detours when compared to TPP* as illustrated in Figure 3d. Notice that, by using different functions or transformations to compute R, one can find different paths that meet the same objectives. The topological path planning (TPP) method, shown in Figure 3, is presented in Algorithm 1.

Algorithm 1 Topological Path Planning (TPP)

```
Input: start point $q_{start}$, goal point $q_{goal}$, obstacles map $M$, manipulator workspace $W$.
1: Initialize a grid $G$ of $N_1 \times N_2$ points containing the workspace of the robot and obstacles.
2: Compute the free space $F$.
3: Compute the topological-constrained set of paths $R$ using a controlled thinning on $F$,
   modulated by the influence of obstacles.
4: Connect the $q_{start}$ and $q_{goal}$ points to the TPP.
5: Find the shortest path $P$ between $q_{start}$ and $q_{goal}$ on $R$.
Output: a topological-constrained path $P$ of between $q_{start}$, goal point $q_{goal}$.
```

3.2. Particular Case of TPP

A particular case of TPP is obtained when the influence zone rule is not relaxed. In this case, the homotopic regions are shrunk until they are only one point wide. The result is a skeleton, a structure that has similarities with the GVD-based path planning method, as shown in Figure 5c. Note that the skeleton surrounds all obstacles numbered 1, 2, and 3 in Figure 5a. As a particular case, we call it TPP*, a version of TPP that delivers a more conservative and safe roadmap, although much faster because it contains fewer nodes. The main disadvantage of the TPP* path is that it often contains unnecessary detours, as highlighted in Figure 5d.

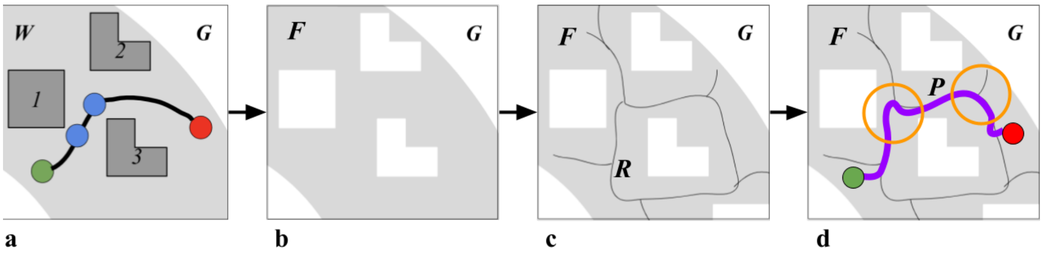


Figure 5. The conservative roadmap using TPP*. (**a**) The environment shows obstacles (1, 2, and 3), start (green), goal (red), and the optimum path. Blue marks represent critical positions. (**b**) The free space F. (**c**) The roadmap R is achieved by skeletonization. (**d**) The shortest path P depicted in purple, highlighting undesired detours.

3.3. Evaluation Metrics

Evaluation metrics based on both paths and trajectories are important to compare the method performances quantitatively and qualitatively. The metrics and the performance criteria are presented in Table 2.

Table 2. Performance metrics.

Metric	Description
Number of Path Points	Number of points contained in the path
Computing Time [s]	Algorithm computing time in seconds
Distance [cm]	Path length in cm measured from start $q_{initial}$ to goal q_{final}
$\sigma(\ddot{\theta})$	Standard deviation of the joint acceleration $\ddot{\theta}$ (std acc); estimates the intensity of the changes in velocity along the path
$\dddot{\theta}_{max}$	Maximum acceleration rate along the path (max jerk)

4. Results

In this section, we present the results of the performance comparison between the PRM and TPP algorithms for different scenarios. Both algorithms were fully implemented in Python using the method Dijkstra of the function `shortest_path` in the `networkx` library to compute the shortest path.

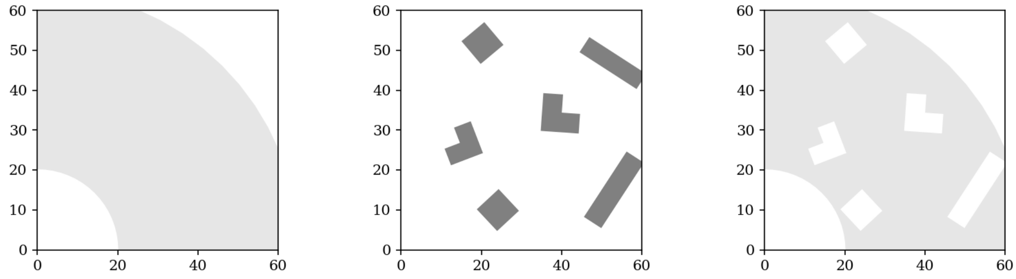
4.1. Scenarios

A total of 100 different scenarios were randomly generated. A safety margin corresponding to half the diameter of the robot tool was added to the obstacles to avoid unfeasible scenarios. Table 3 presents the following parameters used in generating the scenarios.

Table 3. Scenario parameters.

Parameter	Description Points
Scenarios	100 random scenarios with 6 obstacles of different shapes
Grid	Obstacles and workspace are simulated in a 60 cm × 60 cm grid with a 1 cm resolution
Start point	Fixed at location $x, y = 15$ cm, 15 cm
Goal point	Random location inside the SCARA workspace
Safety margin	Half the robot tool diameter is added to obstacle boundary (4 cm in this paper)

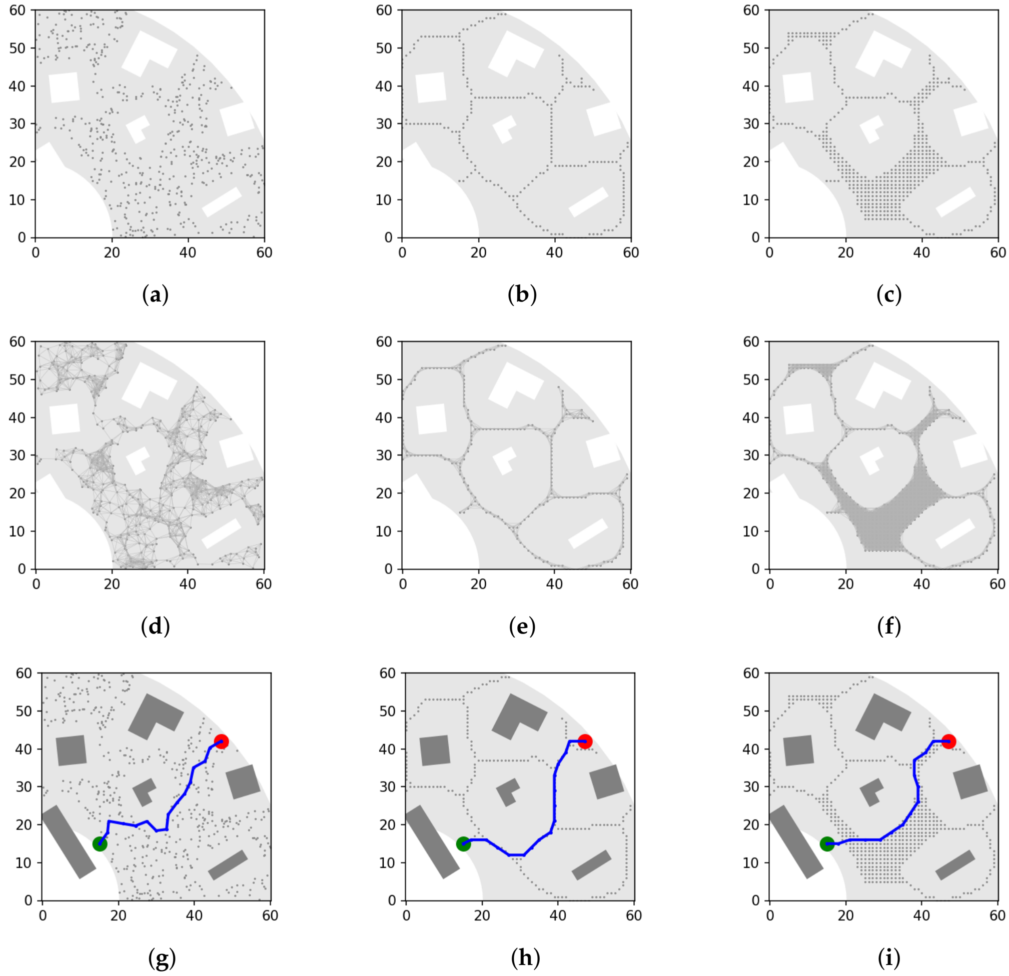
One of the inputs for TPP is a map M of the obstacles inside the robot workspace. This map can be generated using object detection from digital images acquired by a static camera or an image generated by light detection and ranging (LIDAR) sensors. Figure 6 shows an example of an SCARA workspace (Figure 6a), a map M (Figure 6b), and a free space F (Figure 6c).



(**a**) (**b**) (**c**)

Figure 6. Examples of (**a**) an SCARA workspace, (**b**) a map M, and (**c**) a free space F on a 60 cm × 60 cm grid.

The conditions set for the experiments are shown in Table 3. Random points are used only for the PRM algorithm. Fewer random points lead to faster PRM solutions, at the price of more susceptibility to collisions. Neighbor is used in the Dijkstra algorithm to find the shortest path. Closer neighbors lead to fewer collisions and help the Dijkstra algorithm achieve faster results on both PRM and TPP. Figure 7 shows the PRM, TPP*, and TPP roadmap construction and the final shortest path.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

Figure 7. Examples of PRM, TPP* and TPP, respectively: (**a**–**c**) the nodes, (**d**–**f**) the roadmps and (**g**–**i**) the final shortest path.

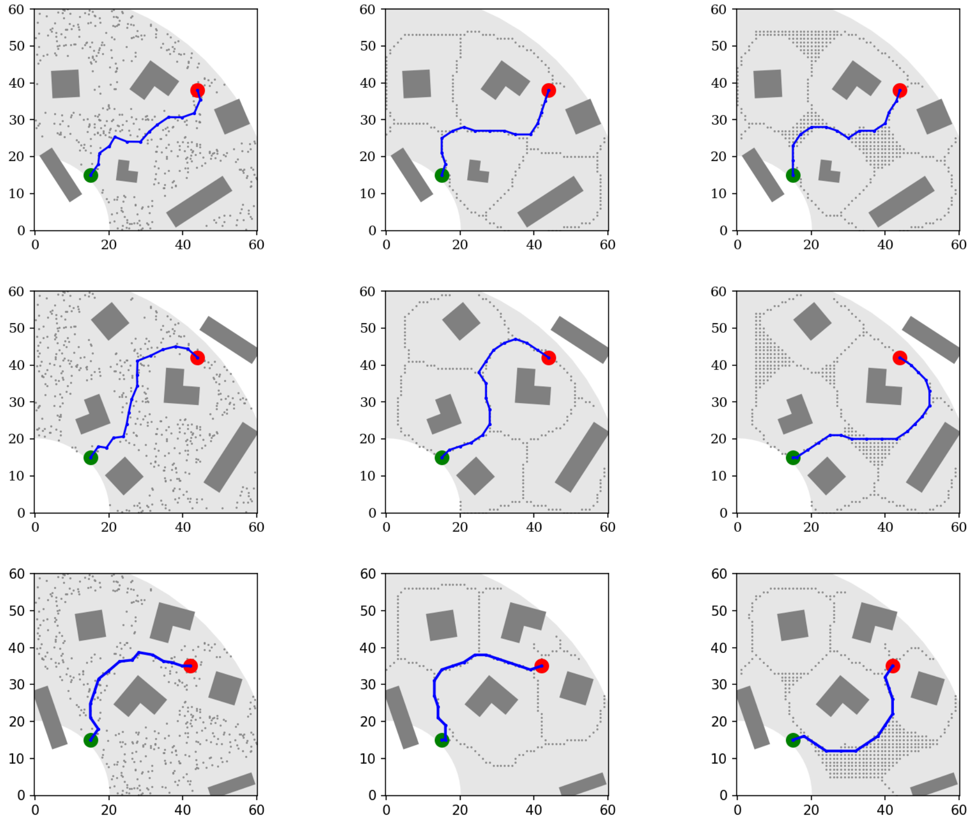
Figure 8 presents the paths collision avoidance PRM, TPP*, and TPP results for three scenarios, different on each line.

4.2. Performance Evaluation

Table 4 shows a comparison of metrics' average performance: distance, number of points on the path, maximum jerk, acceleration deviation, and computing time for 100 scenarios, shown in Figure 8.

Table 4. Comparisons of metrics of methods for path planning shown in Figure 8. The average for 100 scenarios.

Method	**Distance [cm]**	**Points on the Path**	**Maximum Jerk [Degree/s^3]**	**Accel. Deviation**	**Computing Time [s]**
PRM	43.03 ± 6.87	13.80 ± 2.05	10.91 ± 2.81	3.35 ± 0.57	**2.26** ± **0.82**
TPP*	45.90 ± 6.48	14.52 ± 1.91	9.05 ± 4.01	2.87 ± 0.55	2.75 ± 0.52
TPP	**42.86** ± **6.29**	**13.42** ± **1.82**	**7.91** ± **3.22**	**2.69** ± **0.57**	5.37 ± 1.82



(**a**) (**b**) (**c**)

Figure 8. Results of (**a**) PRM, (**b**) TPP*, and (**c**) TPP for three different scenarios, same scenario per row.

As shown in Table 4, we highlight that the TPP presented better results in relation to the PRM and TPP* in the distance criteria, which were the smallest in relation to the PRM and TPP*. Regarding the number of points on the path, the TPP also presented lower results when compared to the PRM and TPP*. The paths shown in Figure 7 illustrate that the TPP path tends to be shorter, thus containing fewer points. The maximum jerk for TPP was the lowest among the others. Regarding the standard deviation of acceleration, the TPP also presented a lower result compared to the PRM and TPP*. Figure 8 shows how smooth the TPP path tends to be, considering the method's intrinsic topological restriction. The TPP has an average processing time that is twice that required by the PRM. However, TPP* has no statistically significant difference to PRM in this criterion. This means that, depending on the application's needs, TPP* can be used instead of TPP, ensuring all the advantages of TPP over PRM.

Figure 9 shows details of the performance metrics. It can be highlighted here, according to Figure 9 and Table 4, that the number of points on the path is similar for the three methods. However, for TPP, the acceleration deviation value is smaller than the other methods. This is important for the application requirements and is explained because the trajectories of the manipulator generated by TPP are smoother.

Although the PRM is clearly faster, due to the randomness of PRM points initialization, some distributions may result in areas of scarcity, thus impeding the shortest path algorithm to find a solution, since it depends on the neighbor distance parameter. Using greater values for the neighbor distance can help to solve this problem.

Table 5 presents the average energy consumption of joint motors for 100 scenarios. The energy consumption values were calculated using Equations (8) and (9) and with the torque values calculated from the dynamic model (Equations (6) and (7)). Notice that TPP's trajectory presents a similar average energy consumption for the application presented in this research when compared to the other methods.

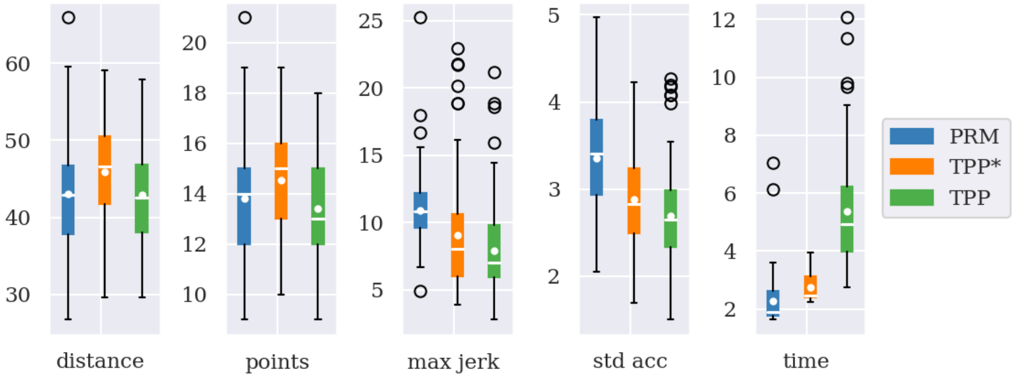


Figure 9. Evaluation metrics for 100 random scenarios.

Table 5. Average energy consumption of joint motors for 100 scenarios for PRM, TPP*, and TPP methods.

Method	Energy Consumption [kWh]
PRM	0.028 ± 0.0084
TPP*	0.030 ± 0.0095
TPP	0.029 ± 0.0092

5. Discussion

In this section, we present a quantitative and qualitative analysis of each algorithm regarding the path planning with collision avoidance and the manipulator dynamics.

Figure 10 shows an overall comparison between the TPP path, features PRM, and TPP*. PRM is very fast for a small number of random nodes. However, this reduces its chance of finding a solution and the randomness provides different solutions for the same scenario, as shown in Figure 11. Indeed, in the worst case, for the same scenario, PRM can find a valid solution for one execution, but not find a valid solution for the next execution. This unpredictability and inconsistency are not desired in collaborative robotics.

Because the TPP roadmap is built to preserve the topology of the free space, it constrains the resulting path at an intermediate distance between obstacles. The path is short, minimizes undesired detours, and ultimately prevents collisions with the obstacles, as shown in Figure 10. TPP finds solutions for any number of nodes, and every solution is feasible. Obviously, the greater the number of nodes, the greater its execution time. Despite TPP always finding the same solution for the same scenario, an alternative solution can be chosen if the prior must be discarded for any reason. Figure 10d illustrates an alternative solution obtained using TPP. Thus, TPP results highlight two main general advantages of its collision-free path: it is predictable and smooth. Although it provides a better single solution per scenario, it does not always deliver the shortest path and the greater number of nodes of TPP compared to TPP* makes the prior slower.

Another relevant property is the fact that the TPP has only one parameter to adjust. This aspect makes the algorithm less dependent on external factors, thus forcing context independence and contributing to simplicity. Finally, TPP always generates collision-free paths, which makes it safer and more efficient. It is also fundamental in the context of real-time environments that require a predictable solution.

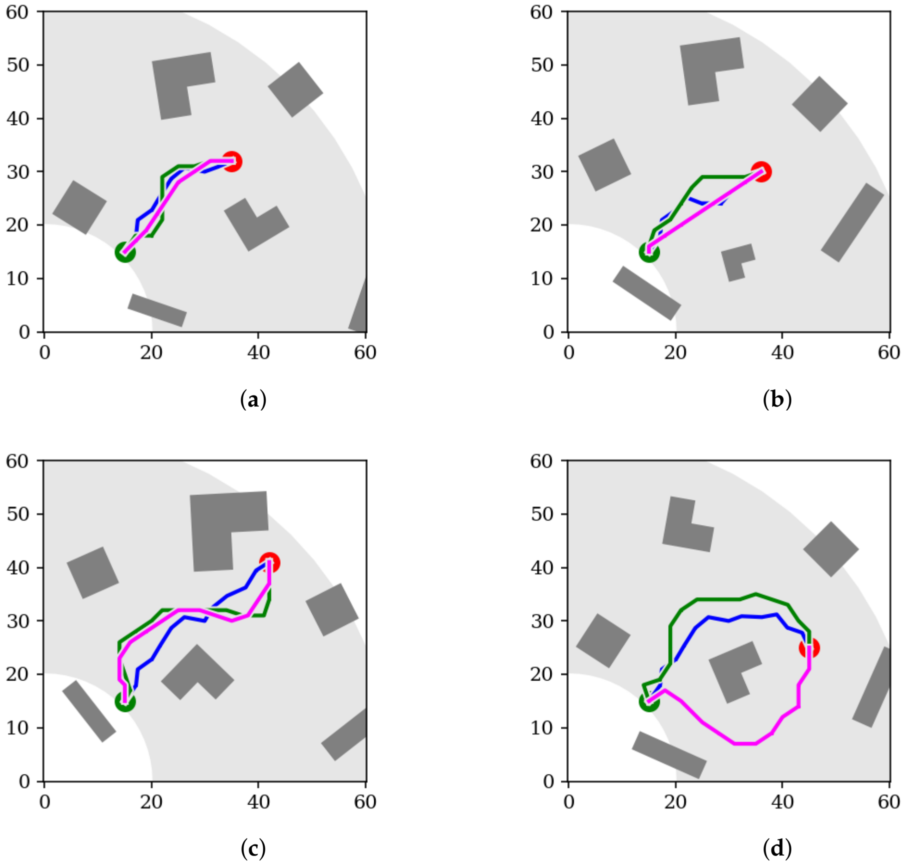


Figure 10. Best features of TPP path (magenta) in comparison to PRM (blue) and TPP* (green). TPP delivers (**a**) shorter, (**b**) less conservative (fewer detours), (**c**) smother, and (**d**) alternative paths.

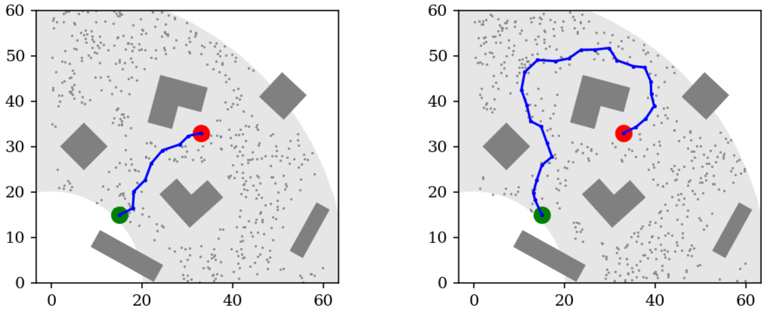
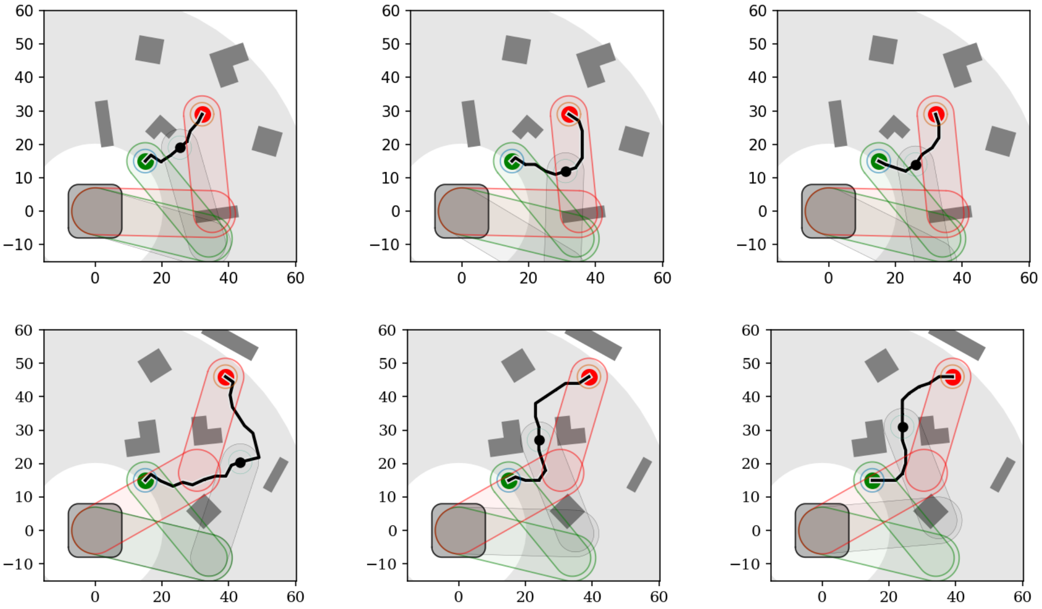


Figure 11. Example of PRM, showing two different solutions for the same scenario.

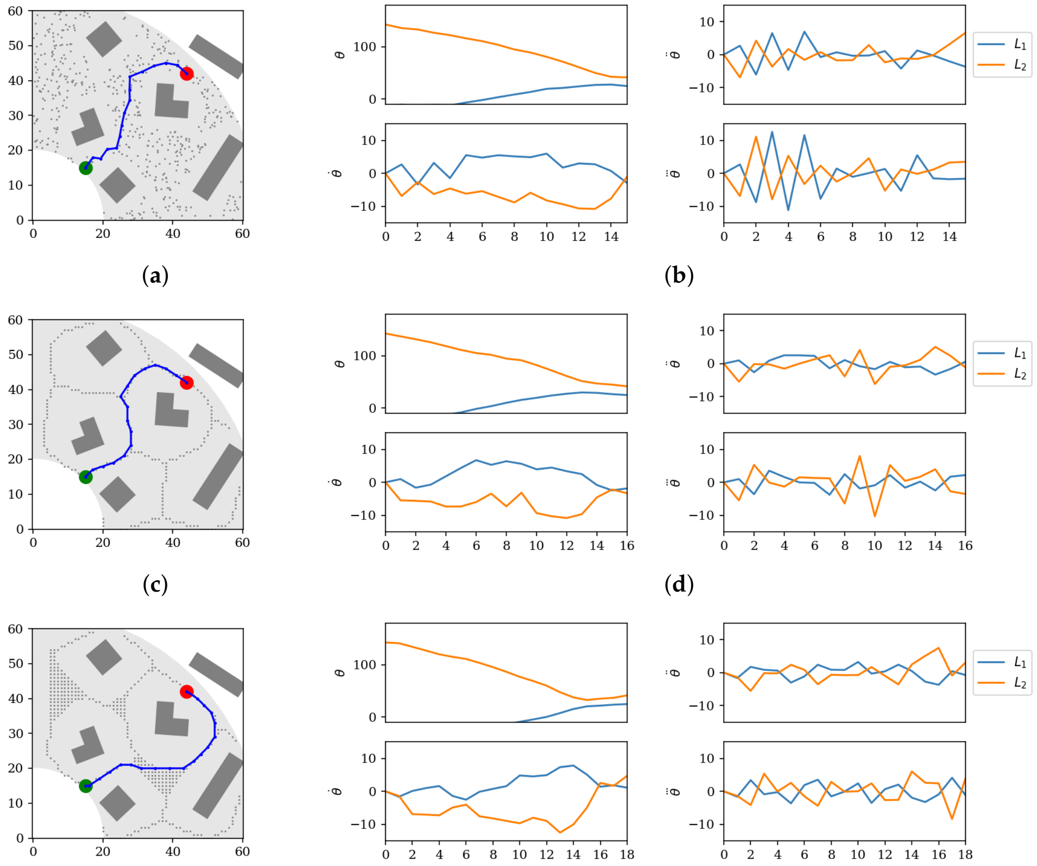
Figure 12 shows an SCARA manipulator trajectory where the manipulator is positioned at the start, middle, and goal points for two scenarios. Figure 12 also shows that all paths are feasible. We can state that the paths generated by the methods are executable by the manipulator, without having problems with singularities. Another important issue that we can highlight is that the manipulator's movements are carried out within its workspace.

The values of θ for the PRM, TPP*, and TPP, presented in Figure 13 were found from the inverse kinematic model presented in Equations (3) and (4). From the values of θ, the velocities, accelerations, and jerks can be found. It should also be noted here that the values of θ will be sent to the joint controllers to carry out their displacements; that is, they are the controllers' set points.



(**a**) (**b**) (**c**)

Figure 12. SCARA manipulator trajectories for (**a**) PRM, (**b**) TPP*, and (**c**) TPP. Manipulator positioned at start (green), middle (black), and goal (red) points for two different scenarios (top and bottom rows).



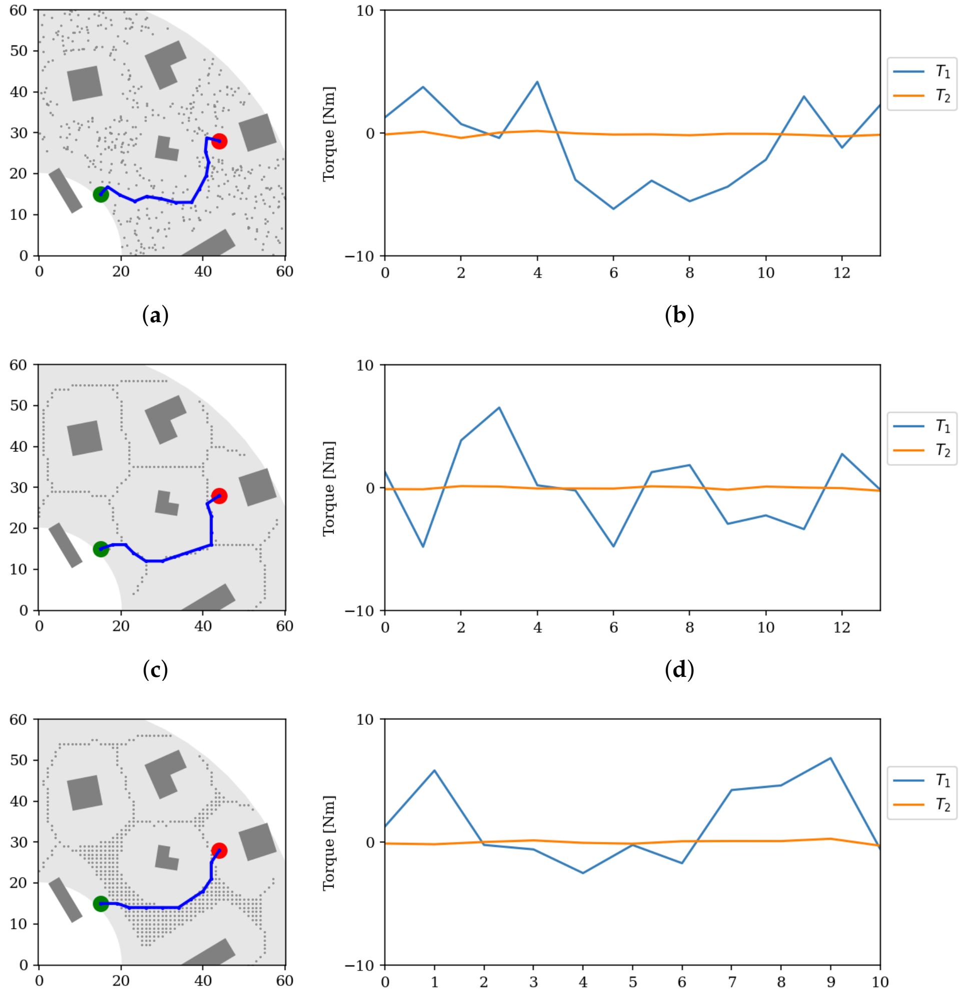
(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 13. Results comparison for same scenario. (**a**,**b**) PRM path and thetas. (**c**,**d**) TPP* path and thetas. (**e**,**f**) TPP path and thetas.

The joint's torques were obtained from the dynamic model and Equations (6) and (7) of the manipulator, and are shown in Figure 14. A scenario is presented for each method: PRM, TPP*, and TPP. Torques were calculated by taking the trajectories, calculated by inverse kinematics (Equations (3) and (4)), in addition to the values of speeds and accelerations of the joints.

An important contribution to this case study can be highlighted by the analysis carried out using the standard deviation of acceleration (see Table 4 and Figure 9), which is a measure that indicates how uniform the acceleration values are, which is why, in this research, this value was calculated to be analyzed. The idea of this check was to analyze how smooth the trajectories of the manipulator are; that is, how the joints behave.

It is also important to highlight that the algorithms used are generating paths for a manipulator and not for a mobile robot. In view of this, some observations must be taken into consideration, as a manipulator will be transporting some material trapped in its claw.



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 14. SCARA manipulator joint's torques calculated from trajectories, velocities, and accelerations for PRM, TPP*, and TPP results for same scenario. (**a**,**b**) PRM path and torques. (**c**,**d**) TPP* path and torques. (**e**,**f**) TPP path and torques.

Another point that must be analyzed is the joint jerks, which represent how quickly a joint will move from one point to another. This must also be analyzed very carefully, as the manipulator may suffer damage that could damage them and cause problems in couplings and mechanical and electrical systems.

We highlight that TPP presents smoother trajectories than the PRM, and overall shorter paths than TPP*, as can be seen in Figure 10. Using TPP, the manipulator will have a better performance for the proposed application, though TPP* is faster.

We also highlight here that TPP has some advantages over classical methods, such as the PRM and APF. TPP will always generate consistent paths; that is, it does not have probabilistic routes like the PRM, nor is it very sensitive to parameter adjustment as is the case with APF. APF adjustments of the three parameters are necessary, such as the gain in the attraction force of the endpoint, the gains in repulsion from the obstacles, and the initial position repulsion. In the case of TPP, as stated here, it will always generate the same path for the same adjustment and its single parameter is very easy to tune. The TPP also generated smooth trajectories for the displacements of the manipulator's joints, as shown in Figure 13.

TPP is based on the framework of mathematical morphology that was developed for application in robotic manipulators and mobile robotics for path planning and collision avoidance applications. TPP produces a roadmap in only one step without using additional optimization to prune nodes and has only one parameter to adjust $\in [0, \mathrm{inf}]$, which can be used to tune the compromise between processing time and trajectory quality (considering the application requirements) without changing the success rate to achieve the goal.

Table 6 presents a summary comparison of the characteristics and each method mentioned in the references of this work and in comparison with the TPP, the proposed method.

Table 6. Comparison of the relevant characteristics of TPP with literature methods.

Method	Characteristics	References
PRM	Based on random graphs, this global method presents a high probability of random paths, not always guaranteeing that the path found is the optimal path. The number of random nodes plays an important role in the result.	[4,22]
RRT	Based on random trees, this global method presents a high probability of random paths and does not always guarantee that the path found is the optimal path. The path is smooth.	[28,29]
APF	This potential-based global model present a low probability of random paths, but the solution is highly dependent on the attractive and repulsive forces adjustments. The path is short and smooth.	[2]
GVD	This is a topology-based global model featuring a low probability of random walks. The path found is the safest but too conservative, leading to unnecessary detours. No parameters to adjust.	[3]
TPP	This is a topology-based global model that presents a low probability of random paths (high predictability) and often guarantees that the path found is close to the optimal path. Its single parameter modulates the result and influences the computing time.	-
TPP*	This is a particular case of TPP presenting the same characteristics of GVD. No adjustments are necessary.	-

The method proposed in this paper will be used for pick-and-place tasks. Thus, the manipulator must perform slow movements, as it is moving with a load held in the gripper. According to our applications, for manipulation tasks, the speeds that we want to apply are in the range of 0.1 to 1 m/s (linear speeds) and 5 to 50 degrees/s (angular speeds) [35]. For inspection tasks, the speed of the camera (mounted on the end-effector) will be in the range of 0.10 to 0.30 m/s [40].

6. Conclusions

This paper presented an approach based on an image processing algorithm to perform path planning with collision avoidance. Our method, named topological path planning (TPP), uses morphological operations on a map of obstacles to create a topological path that is collision-free. This path is used to guide an SCARA manipulator and make it move around the obstacles from start to goal points. Because of the topological constraint, TPP provides smoother paths than the PRM, a desired characteristic to create the manipulator trajectories. However, because the number of points in TPP is higher than for TPP*, the latter is faster.

We highlight that the TPP presented better results in relation to the PRM and TPP* in the distance criteria, which were the smallest, 42.86 cm, in relation to the PRM and TPP*'s 43.03 cm and 45.90 cm, respectively. Regarding the number of points on the path, the TPP also presented lower results, 13.42, while they were 13.80 for the PRM and 14.52 for the TPP*. The maximum jerk for TPP was the lowest among the others, which was 7.91 degree/s^3; for PRM, it was 10.91 degree/s^3, and for TPP*, it was 9.05 degree/s^3. Regarding the standard deviation of acceleration, the TPP also presented a lower result, 2.69, while that of the PRM was 3.35 and that of the TPP* was 2.87. In the computation time criterion, PRM presented the best result, 2.26 s, TPP* spent 2.75 s, and TPP spent 5.37 s, which was the longest time. From the five criteria compared, TPP presented the best result in four criteria, being below the others only in relation to computing time. With this, it can also be concluded that TPP can be used as a collision-free trajectory planning method for industrial manipulators.

The performance of TPP was compared to the PRM, a fast method that is also based on the previous knowledge of the static obstacle map. The results show overall advantages to using TPP since it provides better manipulator control, considering the smaller variations in the joint velocities. Furthermore, TPP is a non-stochastic method that provides a collision avoidance path planning solution with repeatability and predictability, since, for a given scenario, the solution is always the same. Evaluating TPP only for 2D trajectory planning and the use of only two SCARA axes are the main shortcomings of this study. Furthermore, TPP is often slower than the PRM because the script generated by TPP contains more points than necessary, making it more difficult to find the shortest path. In future works we intend to optimize TPP and use the full SCARA model to evaluate it in pick-and-place tasks, considering 3D obstacles. Implementing TPP on our SCARA platform is not straightforward, requiring PLC programming to interface a computer system that generates the maps and the paths. These implementations will be carried out in future work.

Author Contributions: Conceptualization, J.G.B. and G.L.B.R.; methodology, J.G.B. and G.L.B.R.; software, J.G.B., G.L.B.R. and D.S.F.; validation, J.G.B., G.L.B.R. and D.S.F.; formal analysis, G.L.B.R.; investigation, J.G.B. and G.L.B.R.; resources, J.G.B., G.L.B.R. and D.S.F.; data curation, M.A.T. and A.L.O.; writing—original draft preparation, J.G.B., G.L.B.R., M.A.T. and A.L.O.; writing—review and editing, J.G.B., G.L.B.R. and D.S.F.; visualization, J.G.B., D.S.F. and G.L.B.R.; supervision, G.L.B.R.; project administration, J.G.B., D.S.F. and G.L.B.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by PROINFRA/PRPI/IFCE 13/2022 grant number 13208.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank FUNCAP for the financial support of this work.

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

Abbreviations

The following abbreviations are used in this manuscript:

APF	Artificial Potential Field
DOF	Degree Of Freedom
D-H	Denavit–Hartenberg
GVD	Generalized Voronoi Diagram
IRRT	Informed Rapidly exploring Random Tree
IVPF	Improved Velocity Potential Field
LIDAR	Light Detection and Ranging
MM	Mathematical Morphology
PLC	Programmable Logic Controller
PMs	Parallel Manipulators
PRM	Probabilistic Roadmap Method
PSO	Particle Swarm Optimization
RRT	Rapidly exploring Random Tree
SCARA	Selective Compliance Assembly Robot Arm
TPP	Topological Path Planning

References

1. Pinto, M.F.; Mendonça, T.R.; Olivi, L.R.; Costa, E.B.; Marcato, A.L. Modified approach using variable charges to solve inherent limitations of potential fields method. In Proceedings of the 2014 11th IEEE/IAS International Conference on Industry Applications, Juiz de Fora, Brazil, 7–10 December 2014; pp. 1–6.
2. Khatib, O. The potential field approach and operational space formulation in robot control. In *Adaptive and Learning Systems*; Springer: Berlin/Heidelberg, Germany, 1986; pp. 367–377.
3. Takahashi, O.; Schilling, R. Motion planning in a plane using generalized Voronoi diagrams. *IEEE Trans. Robot. Autom.* **1989**, *5*, 143–150. [CrossRef]
4. Kavraki, L.E.; Svestka, P.; Latombe, J.C.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
5. Kazim, I.J.; Tan, Y.; Qaseer, L. Integration of DE Algorithm with PDC-APF for Enhancement of Contour Path Planning of a Universal Robot. *Appl. Sci.* **2021**, *11*, 6532. [CrossRef]
6. Dias, E.V.; Batista, J.G.; Silva, C.G.; Ramalho, G.L.; Souza, D.A.; Silva, J.L.N.; Moreira, A.P. Path Planning Collision Avoidance of a SCARA Manipulator using PRM with Fuzzy. In Proceedings of the 2022 XXIV Congresso Brasileiro de Automática (CBA-2022), Fortaleza, Brazil, 16–19 October 2022. Available online: https://www.sba.org.br/cba2022/wp-content/uploads/artigos_cba2022/paper_7227.pdf (accessed on 28 September 2023).
7. Chi, W.; Ding, Z.; Wang, J.; Chen, G.; Sun, L. A generalized Voronoi diagram-based efficient heuristic path planning method for RRTs in mobile robots. *IEEE Trans. Ind. Electron.* **2021**, *69*, 4926–4937. [CrossRef]
8. Lee, J.Y.; Choset, H. Sensor-based exploration for convex bodies: A new roadmap for a convex-shaped robot. *IEEE Trans. Robot.* **2005**, *21*, 240–247.
9. Lee, D.T.; Drysdale, R.L., III. Generalization of Voronoi Diagrams in the Plane. *SIAM J. Comput.* **1981**, *10*, 73–87. [CrossRef]
10. Zhang, Y.; Zhang, L.; Lei, L.; Xu, F. An Improved Potential Field-Based Probabilistic Roadmap Algorithm for Path Planning. In Proceedings of the 2022 6th International Conference on Automation, Control and Robots (ICACR), Shanghai, China, 23–25 September 2022; pp. 195–199.
11. Chen, J.; Zhou, Y.; Gong, J.; Deng, Y. An improved probabilistic roadmap algorithm with potential field function for path planning of quadrotor. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 3248–3253.
12. Jin, Q.; Hu, Q.; Zhao, P.; Wang, S.; Ai, M. An Improved Probabilistic Roadmap Planning Method for Safe Indoor Flights of Unmanned Aerial Vehicles. *Drones* **2023**, *7*, 92. [CrossRef]
13. Li, W.; Wang, L.; Zou, A.; Cai, J.; He, H.; Tan, T. Path Planning for UAV Based on Improved PRM. *Energies* **2022**, *15*, 7267. [CrossRef]
14. Wu, Y.; Jiang, B.; Xu, H. Formation control strategy of multi-agent system with improved probabilistic roadmap method applied in restricted environment. In Proceedings of the 2021 6th International Conference on Computational Intelligence and Applications (ICCIA), Xiamen, China, 11–13 June 2021; pp. 233–237.
15. Ravankar, A.A.; Ravankar, A.; Emaru, T.; Kobayashi, Y. HPPRM: Hybrid potential based probabilistic roadmap algorithm for improved dynamic path planning of mobile robots. *IEEE Access* **2020**, *8*, 221743–221766. [CrossRef]
16. Han, L. Hybrid probabilistic roadmap-Monte Carlo motion planning for closed chain systems with spherical joints. In Proceedings of the IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, New Orleans, LA, USA, 26 April–1 May 2004; Volume 1, pp. 920–926.

17. Altinoz, O.T.; Yanar, T.A.; Ozguven, C.; Demirdogen, G.; Yilmaz, A.E. Improved non-Probabilistic Roadmap method for determination of shortest nautical navigation path. In Proceedings of the 2017 4th International Conference on Electrical and Electronic Engineering (ICEEE), Ankara, Turkey, 8–10 April 2017; pp. 261–266.
18. Cao, K.; Cheng, Q.; Gao, S.; Chen, Y.; Chen, C. Improved PRM for path planning in narrow passages. In Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2019; pp. 45–50.
19. Jang, K.; Baek, J.; Park, S.; Park, J. Motion planning for closed-chain constraints based on probabilistic roadmap with improved connectivity. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 2035–2043. [CrossRef]
20. Liu, C.; Chang, J.; Liu, C. Path planning for mobile robot based on an improved probabilistic roadmap method. *Chin. J. Electron.* **2009**, *18*, 395–399.
21. Chi, Z.; Yu, Z.; Wei, Q.; He, Q.; Li, G.; Ding, S. High-Efficiency Navigation of Nonholonomic Mobile Robots Based on Improved Hybrid A* Algorithm. *Appl. Sci.* **2023**, *13*, 6141. [CrossRef]
22. Chai, Q.; Wang, Y.; He, Y.; Xu, C.; Hong, Z. Improved PRM path planning in narrow passages based on PSO. In Proceedings of the 2022 IEEE International Conference on Mechatronics and Automation (ICMA), Guilin, China, 7–10 August 2022; pp. 41–46.
23. Baek, D.; Hwang, M.; Kim, H.; Kwon, D.S. Path planning for automation of surgery robot based on probabilistic roadmap and reinforcement learning. In Proceedings of the 2018 15th International Conference on Ubiquitous Robots (UR), Honolulu, HI, USA, 26–30 June 2018; pp. 342–347.
24. Xia, X.; Li, T.; Sang, S.; Cheng, Y.; Ma, H.; Zhang, Q.; Yang, K. Path Planning for Obstacle Avoidance of Robot Arm Based on Improved Potential Field Method. *Sensors* **2023**, *23*, 3754. [CrossRef] [PubMed]
25. Li, Q.; Xu, Y.; Bu, S.; Yang, J. Smart vehicle path planning based on modified PRM algorithm. *Sensors* **2022**, *22*, 6581. [CrossRef] [PubMed]
26. Alam, M.M.; Nishi, T.; Liu, Z.; Fujiwara, T. A Novel Sampling-Based Optimal Motion Planning Algorithm for Energy-Efficient Robotic Pick and Place. *Energies* **2023**, *16*, 6910. [CrossRef]
27. Kelaiaia, R.; Chemori, A.; Brahmia, A.; Kerboua, A.; Zaatri, A.; Company, O. Optimal dimensional design of parallel manipulators with an illustrative case study: A review. *Mech. Mach. Theory* **2023**, *188*, 105390. [CrossRef]
28. Wu, D.; Wei, L.; Wang, G.; Tian, L.; Dai, G. APF-IRRT*: An Improved Informed Rapidly-Exploring Random Trees-Star Algorithm by Introducing Artificial Potential Field Method for Mobile Robot Path Planning. *Appl. Sci.* **2022**, *12*, 10905. [CrossRef]
29. Long, H.; Li, G.; Zhou, F.; Chen, T. Cooperative Dynamic Motion Planning for Dual Manipulator Arms Based on RRT*Smart-AD Algorithm. *Sensors* **2023**, *23*, 7759. [CrossRef]
30. Batista, J.G. Planejamento de Trajetória Usando Campos Potenciais Artificiais com Metaheurísticas. *IEEE Lat. Am. Trans.* **2020**, *18*, 914–922. [CrossRef]
31. Batista, J.G.; da Silva, J.L.; Thé, G.A. Can Artificial Potentials Suit for Collision Avoidance in Factory Floor? In Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018), Porto, Portugal, 29–31 July 2018; Volume 2, pp. 547–556.
32. Hartenberg, R.; Danavit, J. *Kinematic Synthesis of Linkages*; McGraw-Hill: New York, NY, USA, 1964.
33. Batista, J.; Souza, D.; Dos Reis, L.; Barbosa, A.; Araújo, R. Dynamic model and inverse kinematic identification of a 3-DOF manipulator using RLSPSO. *Sensors* **2020**, *20*, 416. [CrossRef]
34. Mohanta, J.C.; Keshari, A. A knowledge based fuzzy-probabilistic roadmap method for mobile robot navigation. *Appl. Soft Comput.* **2019**, *79*, 391–409. [CrossRef]
35. Sciavicco, L.; Siciliano, B.; Villani, L.; Oriolo, G. *Robotics: Modelling, Planning and Control, Ser. Advanced Textbooks in Control and Signal Processing*; Springer: London, UK, 2011.
36. Spong, M.W.; Vidyasagar, M. *Robot Dynamics and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2020.
37. Soille, P. *Morphological Image Analysis*; Springer: Berlin/Heidelberg, Germany, 2004. [CrossRef]
38. Preteux, F. Watershed and Skeleton by Influence Zones: A Distance-Based Approach. *J. Math. Imagind Vis.* **1992**, *1*, 239–255. [CrossRef]
39. Bhattacharya, M.L.S.; Kumar, V. Topological constraints in search-based robot path planning. *Auton. Robot. Vol.* **2012**, *33*, 273–290. [CrossRef]
40. Mineo, C.; Pierce, S.G.; Nicholson, P.I.; Cooper, I. Robotic path planning for non-destructive testing–A custom MATLAB toolbox approach. *Robot. Comput.-Integr. Manuf.* **2016**, *37*, 1–12. [CrossRef]

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