sensors

MDPI

Article

Water Sink Model for Robot Motion Planning

Gi-Yoon Jeon [1,2] and Jin-Woo Jung [1,*]

[1] Department of Computer Science and Engineering, Dongguk University, 30, Pildong-Ro 1-Gil, Jung-Gu, Seoul 04620, Korea; gyjeon@add.re.kr

[2] Agency for Defense Development, Songpa P.O. Box 132, Seoul 05771, Korea

[*] Correspondence: jwjung@dongguk.edu; Tel.: +82-2-2260-3812

Received: 22 January 2019; Accepted: 7 March 2019; Published: 13 March 2019

Abstract: There are various motion planning techniques for robots or agents, such as bug algorithm, visibility graph, Voronoi diagram, cell decomposition, potential field, and other probabilistic algorithms. Each technique has its own advantages and drawbacks, depending on the number and shape of obstacles and performance criteria. Especially, a potential field has vector values for movement guidance to the goal, and the method can be used to make an instantaneous and smooth robot movement path without an additional controller. However, there may be some positions with zero force value, called local minima, where the robot or agent stops and cannot move any further. There are some solutions for local minima, such as random walk or backtracking, but these are not yet good enough to solve the local minima problem. In this paper, we propose a novel movement guidance method that is based on the water sink model to overcome the previous local minima problem of potential field methods. The concept of the water sink model is to mimic the water flow, where there is a sink or bathtub with a plughole and floating piece on the water. The plughole represents the goal position and the floating piece represents robot. In this model, when the plug is removed, water starts to drain out via the plughole and the robot can always reach the goal by the water flow. The water sink model simulator is implemented and a comparison of experimental results is done between the water sink model and potential field.

Keywords: robot motion planning; water sink model; artificial potential field; local minima

1. Introduction

There are a number of motion planning techniques for robots or agents. Bug algorithms are very intuitive and simple, but inefficient in many cases [1], while visibility graphs can guarantee the shortest path on a two-dimensional (2D) map, but make low clearance from obstacles and are only good for polygonal obstacle environments [2]. Moreover, Voronoi diagrams can guarantee high clearance from obstacles, but they also have high computational complexity for path creation [3], and cell decomposition is simple but also carrying high computational complexity depending on the number of created cells [4]. The PRM (Probabilistic Roadmap Method) and RRT (Rapidly exploring Random Tree) are probabilistic algorithms. PRM makes a graph with node that is randomly generated and find the shortest path on the graph [5,6]. The RRT makes a tree that is randomly expanded from the starting point to the goal. The RRT provides a successful result in many cases. However, the RRT could not guarantee the optimality and completeness of path. The randomized kinodynamic planning considers velocity, acceleration, force, and torque of mobile robot with original RRT [7]. The RRT* is a mixture of PRM and RRT and it generates a random graph to find a path [8] and the informed RRT* is an improvement method for the optimality and completeness of RRT* [9]. In recent years, the heuristic-based algorithm, including neural network, fuzzy logic technique, and nature inspired algorithms has been proposed [10].

Most of these methods require an additional controller in the physical application to follow the created path. Besides, the potential field method can make an instantaneous and smooth robot movement path without an additional controller, but it has a local minima problem that is caused by positions with zero force value where the robot or agent stops and cannot move any further [11,12]. There are some techniques to overcome the local minima problem, such as random walk [13,14] or backtracking. In the case of the random walk technique, it is impossible to predict the path or the processing time because of randomness. The potential guided directional-RRT* [15] is a hybrid method of RRT* and artificial potential field, where artificial potential field directs the random samples toward the goal. This leads to an increase in the speed of RRT*. However, this is still one of the RRT methods. Therefore, this method does not guarantee completeness. In addition, there is an adaptive motion planning method for dynamic environment with artificial potential field using an RRT-based prior path between the starting position and the goal position [16]. However, this is still one of potential field methods. Therefore, this method is not completely free from the local minima problem and inefficiency because this method requires two kinds of path planning processes. To reduce the processing time of the original potential field method, there is a technique by replacing complex obstacles to bounding circles for high-speed vehicles [17]. Similarly, the processing time of potential field method depends on the number of sampling points or the resolution of configuration space. Therefore, a method for reducing the processing time of the potential field method using non-uniform cell decomposition has also been proposed [18]. Even though many variants of the potential field method have been proposed, the local minima problem was not completely resolved yet.

In this paper, we propose a new robot motion planning method, namely water sink model-based motion planning, to resolve the local minima problem using the potential field method by mimicking the flow of water draining out of a sink through a plughole. In the experimental results, the water sink model is compared with the potential field method and the results show a sufficiently practical path on the grid map.

2. Related Works

2.1. Potential Field

A robot in a potential field can be treated as a point, i.e., the robot is represented in the configuration space as a particle under the influence of an artificial potential field U, whose variations reflect the structure of the free space $U(q)$ for given configuration q.

The potential function can be defined over the free space as the sum of an attraction potential $U_{att}(q)$ pulling the robot towards the goal configuration and a repulsion potential $U_{rep}(q)$ pushing the robot away from the obstacles [10], and it can be expressed as

$$U(q) = U_{att}(q) + U_{rep}(q) \tag{1}$$

which is visually exhibited in Figure 1 [12]. The $U_{att}(q)$ is attractive forces (see Figure 1a), the $U_{rep}(q)$ is repulsive forces (see Figure 1b), and $U(q)$ is the sum of attractive forces and repulsive forces (see Figure 1c). The attraction and repulsion potential functions can be formalized, as follows:

$$U(q)_{att} = \frac{1}{2}\xi \parallel q - q_{goal} \ \parallel^2 \tag{2}$$

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)^2, & \rho(q) \le \rho_0 \\ 0, & \rho(q) \ge \rho_0 \end{cases} \tag{3}$$

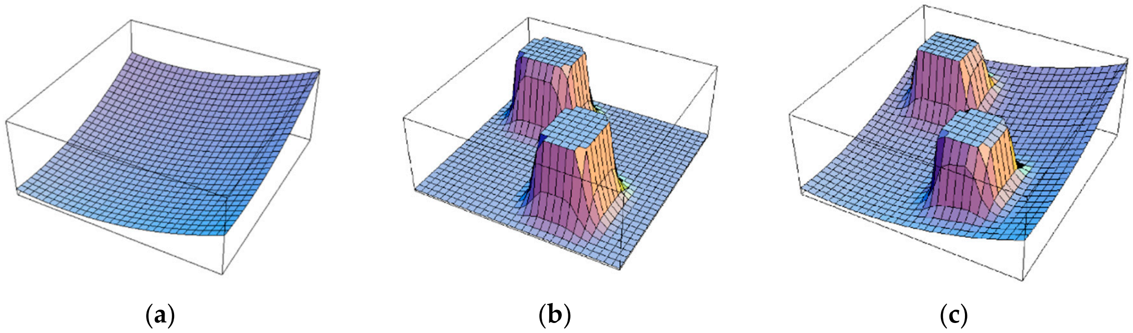


Figure 1. The potential field method: (**a**) attractive forces; (**b**) repulsive forces; and, (**c**) the sum of attractive forces and repulsive forces for potential filed.

In Equation (2), ξ is a scaling factor of attraction force and q_{goal} is the position of goal. In Equation (3), η is a scaling factor of repulsive force between robot and obstacles. $\rho(q)$ is the minimum value of distance from the obstacles and ρ_0 is the influent distance.

2.2. Analysis of Potential Field Method

The potential field method is a kind of combination model for robot path planning and control. The control input for the next robot movement is calculated while using the force field that is the negative gradient of the sum of the potential fields, thus a robot can move smoothly and instantly on the field. However, the sum of attraction and repulsion potentials may have zero vector values at some positions (local minima—see Figure 2), where the robot cannot move to any direction.

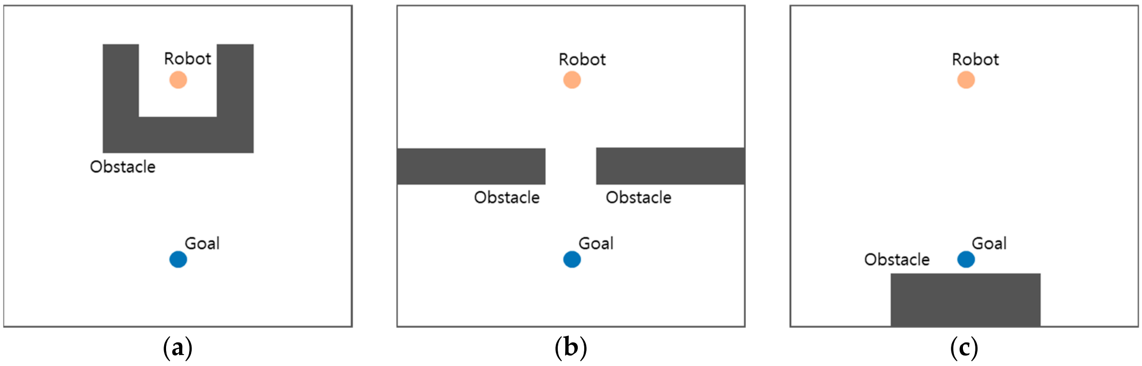


Figure 2. Representative local minima in a potential field: (**a**) the robot is surrounded by obstacles and the exit is opposite the goal; (**b**) narrow distance between obstacles; and, (**c**) the goal is very close to the obstacles.

There have been some suggested improvements to resolve the local minima problem, with one method being random walk. If a robot is in a local minimum position, the robot tries to walk in a random direction to escape the local minimum [15]. However, this method has no guarantee of arrival at the goal on every occasion.

3. Water Sink Model-Based Motion Planning

3.1. The Main Idea behind the Water Sink Model

The concept of the water sink model is to consider the field as a sink or bathtub with a plughole, in which the sink is assumed to be filled with enough water at the initial stage, the robot is assumed to be a floating piece, and the goal is assumed to be the plughole. In this model, when the plug is removed, water starts to drain out via the plughole, and ideally the water supply cannot be exhausted.

The main idea is that there are no local minima that are caused by the flow mechanism of the water and the robot is always able to reach the goal (see Figure 3).

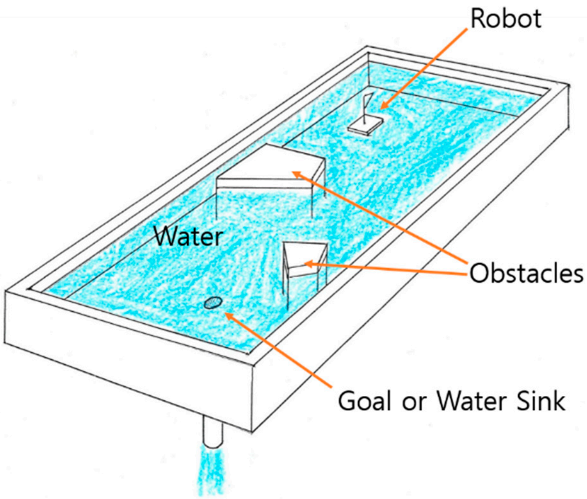


Figure 3. Water sink model concept.

3.2. Water Sink Model Generation

For the implementation of the water sink model, a grid is made of the field in four directions (up, down, left, right) out of cells, the number of plugholes is assumed to be one and it is always included, the volume of draining water via the plughole over time is assumed to be constant, and the number of cells that are involved in water flow expands as time passes. Water flow is assumed to occur from a cell with a higher water level to a cell with a lower water level (due to gravity) at each discrete time instance.

First, only one grid cell is involved in draining. During the next time instance, the first cell and its four connected ones are used together for water drainage, then the number of cells for water drainage continuously expands and the water volume per cell for drainage decreases as time passes. In Figure 4, the assumed draining water volume per one time instance is assumed as 100. Algorithm 1 reflects the WaterMapGeneration algorithm and Figure 5 is an execution example for the water sink model. The example consists of 5 × 5 cells. The cell number 7 is plughole and the cell number 5, 9, 10, 14, and 15 are obstacles. The color of the cell means the water level. The color of the cell becomes brighter when the water level is lower. Contrarily, the color of the cell becomes darker when the water level is higher. The propagation velocity of water drainage, v_{prop} should be defined for the water sink model. In the case of Figure 4, the v_{prop} is assumed as 1.

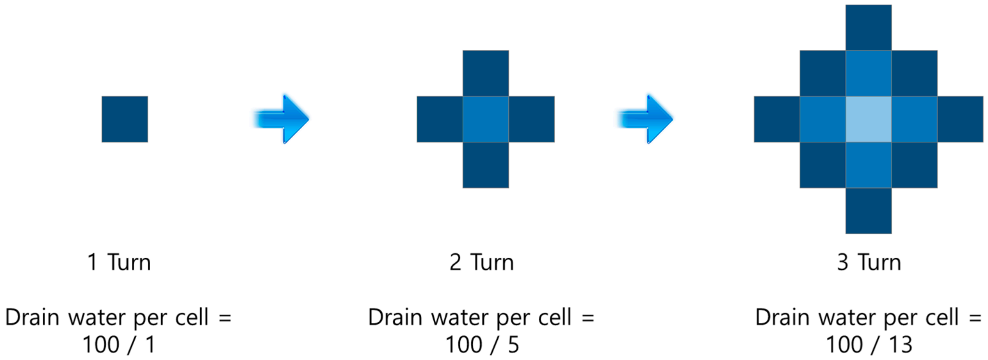

1 Turn
Drain water per cell = 100 / 1
2 Turn
Drain water per cell = 100 / 5
3 Turn
Drain water per cell = 100 / 13

Figure 4. The water sink grid model.

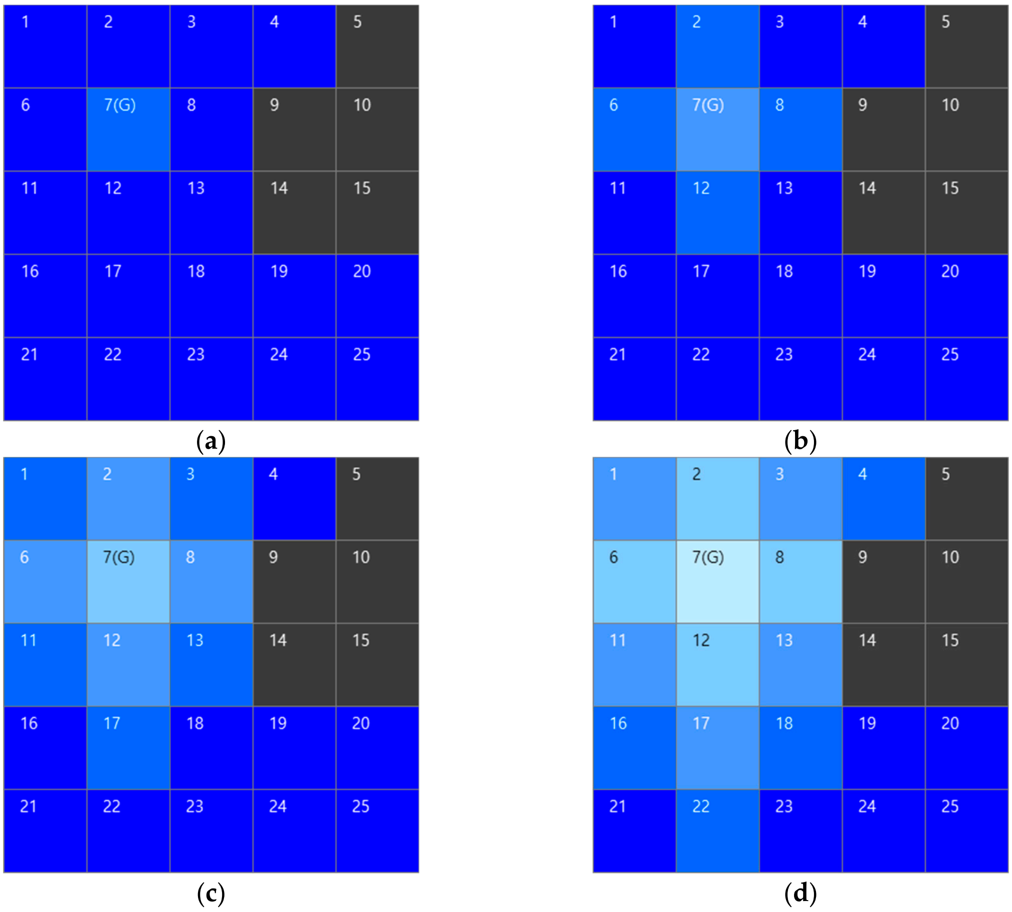


Figure 5. The sequences of water flow in water sink model. The cell number 7(G) is goal or plug hole and cell number 5, 9, 10, 14, and 15 are obstacles. These examples show the progress of water drainage as: (**a**) first turn, only plug hole cell is concerned about drainage; (**b**) second turn, plug hole, and the adjacent cells are involved in drainage; (**c**) third turn; and, (**d**) fourth turn, the cells that are involved in the drainage gradually expand and the each cell color becomes gradually brighter.

Algorithm 1. WaterMapGeneration

```
procedure generateWaterMap(plughole, robot, wall, drainList, waterMap)
  if drainList is empty or robot is in drainList
    return waterMap
  cellNumber ← the size of drainList.
  drain ← drainage water per 1 turn/cellNumber.
  index ← 0.
  repeat
    remove water from waterMap[drainList[index]].
    insertDrain(four neighborhood cells of waterMap[drainList[index]], wall, drainList).
  until the end of drainList
  return waterMap
procedure insertDrain(drainCell, wall, drainList)
  if drainList is empty or drainCell is in drainList
    return drainList
  end ← last cell index of drainList.
  drainList[end] ← drainCell.
  return drainList
```

The complexity of WaterMapGeneration algorithm is $\mathrm{O}(\mathrm{n}^2)$. Here, n is the number of cells in the grid. Water drainage occurs at every turn by propagating to the adjacent cells. The searching continues until the drainList includes the robot cell where the robot exists. In the worst case, the robot cell will be inserted to the drainList at the *n*th turn.

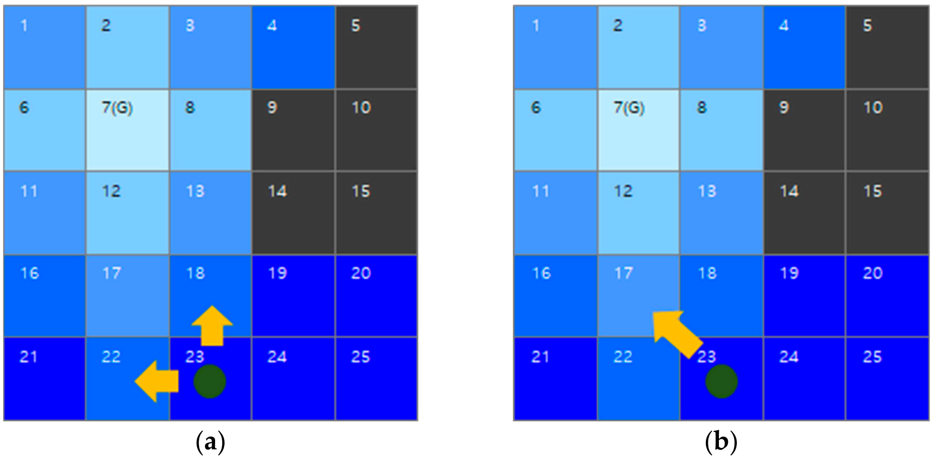
3.3. Water Sink Model-Based Path Generation

Water drainage extends to the adjacent four cells (up, down, left, and right), but the robot can move in any of the adjacent eight directions. If there are two or more cells with the same level of water, then the robot can move to any of these, as cells with the same level of water means an equal chance for the robot to move to any one of them. The following Algorithm 2 represents a pseudocode for path-finding in water sink model:

Algorithm 2. RobotPathGeneration

```
procedure generateRobotPath(plughole, robot, waterMap, robotPath)
  if plughole position is same robot
    return robotPath
  index ← 0.
  repeat
    tempCell[0 . . . 7] ← eight neighborhood cells of robot.
    robot ← minimum water value cell of tempCell[0 . . . 7].
    robotPath[index] ← robot.
    index ← index + 1.
  until robot meets plughole
  return robotPath
```

Figure 6 shows two cases of path generation when the robot is positioned at the cell number 23 in Figure 5d. Figure 6a shows that the water sink model considers four-connectivity for path generation. The robot is able to select the cell number 18 or 22. Regardless of the selection, the lengths of the generated paths are equivalent. However, the four-connectivity model may generate a zigzag motion. To avoid the zigzag motion and enhance the efficiency, the eight-connectivity model is applied in the water sink model. In Figure 6b, the generated sequence of path is cell number 23, 17, 12, and 7.



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

Figure 6. An example of water sink model-based path generation when using four-connectivity and eight-connectivity: (**a**) the case of four-connectivity for robot movement; (**b**) the case of eight-connectivity for robot movement.

The concept of the water sink model may be looked at similarly with Wavefront path planning methods. However, the Wavefront method [19,20], Modified Wavefront method [19], and Focused

Wavefront method [19] use grid cells with a natural number and that number is only utilized for path finding. However, the water sink model uses the real number for the water level in the grid cell. The robot controller or actuator uses this real value to determine the speed as well as the direction of robot movement as the potential field method. In addition, in various kinds of Wavefront methods, all grid cells' natural numbers are fixed, but in the water sink model, the water levels are changed by turn by turn and the amount of water that is drained at a time can be adjusted for various situations.

The complexity of the RobotPathGeneration algorithm is O(n). Here, the n is the number of cells in drainList. The last data of drainList is the index number of the robot cell and the first data of drainList is the index number of the plughole cell. This algorithm generates robotPath by traversing cell numbers from the last data to the first data of drainList. Since the water level could be searched with the cell number at a constant time, the traversing is also completed in O(n).

3.4. Mathematical Verification

In this chapter, we attempt to mathematically verify our method. We explain the simple water model without obstacles and then consider the case with them.

Theorem 1. *In the waterMap generated from WaterMapGeneration algorithm, the water level of a cell farther from the plughole is always higher than the one closer to the plughole if there are no obstacles.*

Proof of Theorem 1. First, we calculate the population of cells that are involved in drainage for each turn (drainage is performed as in Figure 4) and seek for the recurrent equation.

The population of cells that are related with first turn drainage is 1, the population of cells related with second turn drainage are 1 + 4 = 5, the population of cells related with third turn drainage are 1 + 4 + 8 = 13, and the population of cells related with fourth turn drainage are 1 + 4 + 8 + 12 = 25, and so on.

The population of boundary cells for drainage is increased to 0, 4, 8, 12, . . . and the population of boundary cells for *n*th drainage are

$$4n - 4, n = 1, 2, 3, \ \ldots \tag{4}$$

and the population of cells for total water drainage until *n*th turn are

$$\begin{aligned} a_n &= a_{n-1} + 4n - 4, \qquad\qquad a_1 = 1 \\ &= \sum_{k=1}^{n} (4k-4) + a_1 = 4 \times \tfrac{n(n-1)}{2} + 1 = 2n(n-1) + 1 \\ &= 2n^2 - 2n + 1 \end{aligned} \tag{5}$$

Let d be a distance to the cell from the plughole and the distance is measured by city-block or taxicab geometry. Let $V_d(t)$ be the water level in the cell at d distance from the plughole for t turn $\left(0 \le d \le v_{prop}t\right)$. Let V_0 be the initial water level in each cell and C be the amount of drainage water per turn. When the cell with distance d_k is concerned with the drainage for the first time, $\frac{C}{2d_k^2 - 2d_k + 1}$ of water will fall out of the cell. At the next turn, the cells with distance d_k and d_{k+1} are concerned in the drainage and $\frac{C}{2d_k^2 - 2d_k + 1} + \frac{C}{2d_{k+1}^2 - 2d_{k+1} + 1}$ of water will fall out of the cells. Therefore, the remained water level of cell until t turn with distance d_1 and d_2 are expressed by $V_{d_1}(t), V_{d_2}(t)$, and $\left(0 \le d_1 < d_2 \le v_{prop}t\right)$, as

$$V_{d_1}(t) = V_0 - \left(\frac{C}{a_{d_1}} + \frac{C}{a_{d_1+1}} + \cdots + \frac{C}{a_{d_2}} + \frac{C}{a_{d_2+1}} + \cdots + \frac{C}{a_t}\right),\ v_{prop} = 1 \tag{6}$$

$$V_{d_2}(t) = V_0 - \left(\frac{C}{a_{d_2}} + \frac{C}{a_{d_2+1}} + \cdots + \frac{C}{a_t}\right),\ v_{prop} = 1 \tag{7}$$

Subsequently, the water level of $V_{d_2}(t)$ is always smaller than $V_{d_1}(t)$, and so

$$V_{d_1}(t) < V_{d_2}(t) \tag{8}$$

□

Theorem 2. *There are no local minima on the robot path from RobotPathGeneration algorithm in water sink model.*

Proof of Theorem 2. Assume that the robot is located at a local minimum position, then without the loss of generality, this position can be regarded as the new robot starting position. Before finding the robot path, the water sink model has already been created with a plughole while using the above WaterMapGeneration algorithm. Therefore, there is a path that connects the goal to the robot starting position.

If there is a local minimum in the water sink model, then the robot cannot move in any direction from this position. Therefore, there is no difference in the water level between neighboring cells and there is no water drainage thereafter. When a grid cell has no water drainage, the neighboring grid cells also cannot have water drainage. Consequently, the number of grid cells with no water drainage is expanded until some of them meet with the plughole cell. As result, water cannot drain out of the plughole, although some water is able to drain through the plughole during every discrete time instance, which is contradictory. Therefore, the first assumption that a robot located at a local minimum position is false, and so there is no local minimum on the robot path. □

Theorem 3. *RobotPathGeneration algorithm always generate a path from the robot position to the goal if the robot or plughole is not surrounded by obstacles.*

Proof of Theorem 3. If we consider backtracking from the goal position (plughole) to the start position (robot), a cell that is adjacent to the robot position always exists and the cell is closer to the goal by a single cell length. After the robot moves to an adjacent cell in the same manner, a closer cell to the goal exists by a single cell length once again. This process is continued until the robot reaches the goal. Therefore, if the goal or robot is not surrounded by obstacles, there exists a path from the robot to the plughole on every occasion. □

Theorem 4. *The robot always arrives at the goal by using RobotPathGeneration algorithm as long as the robot is not completely surrounded by obstacles.*

Proof Theorem 4. In the first case (see Figure 7), the lower water level cell is certainly being presented to the four cells adjacent to the robot's position. Thus, there is a water level difference, and so the robot should move to the lower water level cell. This is contradictory and so the assumption is false. In the case of a circular path (see Figure 8), we can arbitrarily select a cell and set it to be the start cell for the robot's motion loop, and its movement along the circular path causes the water level to become lower. However, on some occasions, the robot might jump to the start cell, but it is restricted from moving to a higher water level cell in the water sink model. This is contradictory and so the assumption is false. □

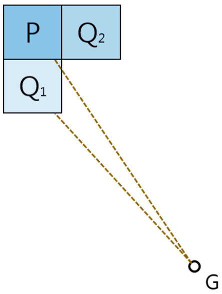


Figure 7. The case that robot is stopped in another position P, not same with position G.

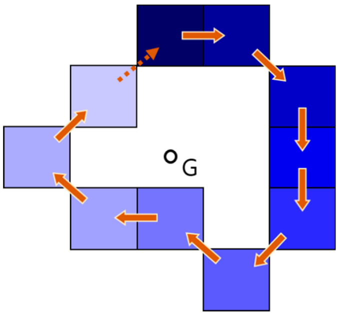


Figure 8. The case of circular path.

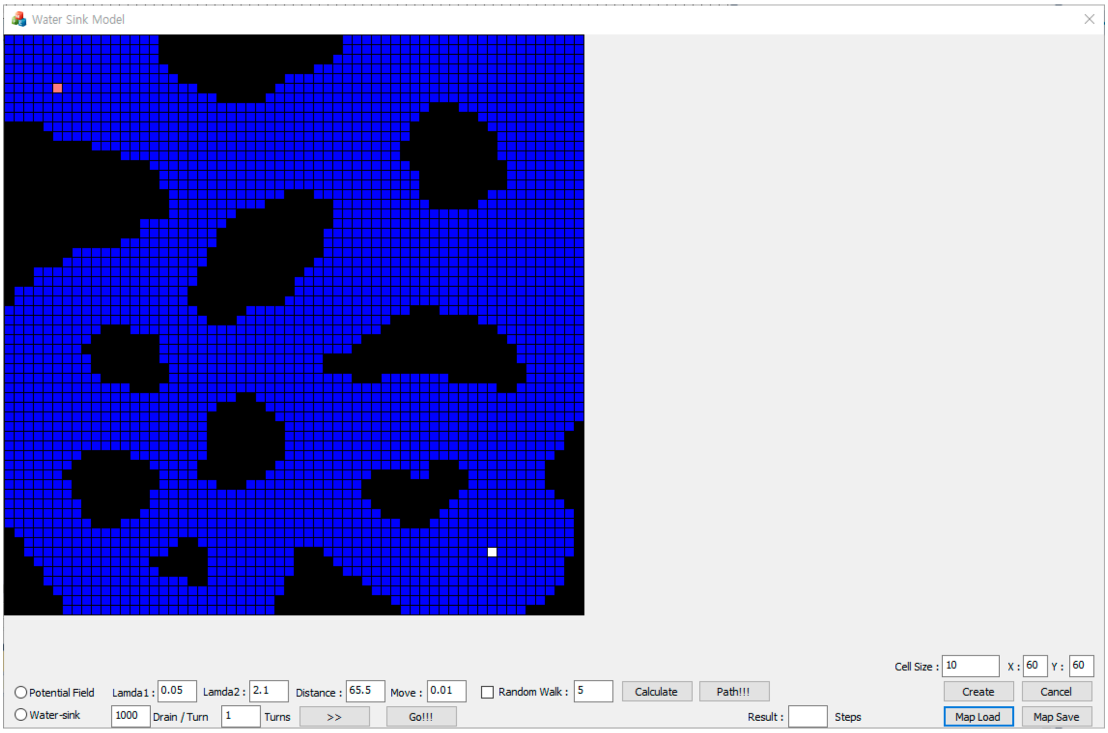
In a world with obstacles, there is a path from the robot's position to the plughole (goal) by Theorem 2 and the robot can reach the goal by Theorems 3 and 4.

4. Experimental Results

4.1. Implementation of Water Sink Model

4.1.1. Experimental Simulator Setup

An experimental simulator was implemented with Visual C++ (Visual Studio Community 2015) on Windows OS (see Figure 9). The simulator is composed of three parts. The first is map generation part (see Figure 10a). At this part, operator can create new map with any size and load or save a map. The second part is a potential field part (see Figure 10b). This part is needed to compare with the water sink model and the operator determines the potential field parameters. The last part is the water sink model part (see Figure 10c). The operator can set turn number for drainage, but this is not the main function of this part and the water sink model has one input parameter, which is the amount of water of drainage per one turn.

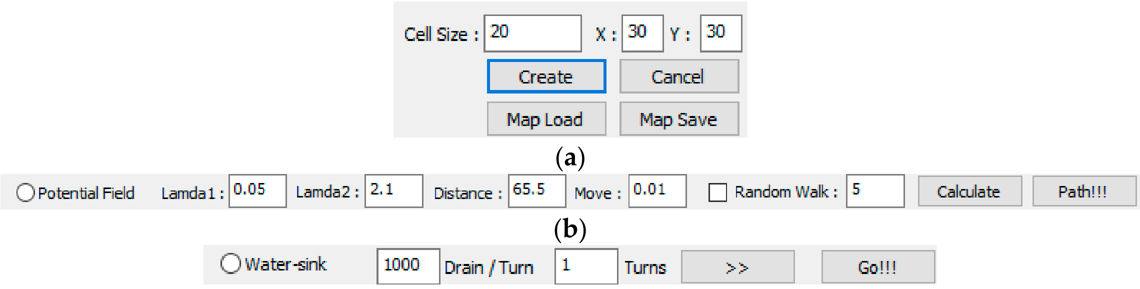


Cell Size : 10 X : 60 Y : 60

○ Potential Field Lamda1 : 0.05 Lamda2 : 2.1 Distance : 65.5 Move : 0.01 □ Random Walk : 5

○ Water-sink 1000 Drain / Turn 1 Turns Result : Steps

Figure 9. Water sink model simulator.



Cell Size : 20 X : 30 Y : 30

(**a**)

○ Potential Field Lamda1 : 0.05 Lamda2 : 2.1 Distance : 65.5 Move : 0.01 □ Random Walk : 5

(**b**)

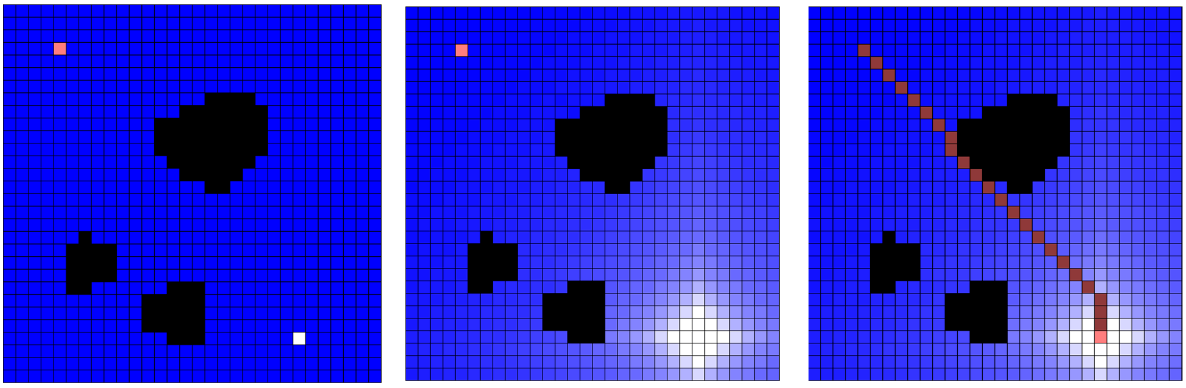
○ Water-sink 1000 Drain / Turn 1 Turns

(**c**)

Figure 10. Parameter adjustment of experimental simulator: (**a**) map creation part; (**b**) potential field part; and, (**c**) water sink model part.

4.1.2. The Process of Water Sink Model Simulation

The implemented simulator can make a water map with the cell size and the number of cells. After that, the user can determine obstacle (black-colored cell), robot position (pink-colored cell) and plug hole (white-colored cell). When the cell in which the robot is located is involved in the drainage, the draining is completed, and the robot starts to move. When the robot reaches the plughole, the experiment is terminated. The robot steps by water sink are colored dark-pink cell (see Figure 11).



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

Figure 11. Water sink model simulator example. Example is 20 (pixels) cell size, 30 × 30 cells and the following is a series of map creation, water drainage, and robot path. The drainage water is 1000 per turn: (**a**) there are three obstacles, the robot is located at the upper-left corner, and the plughole is located at the lower-right corner; (**b**) water drainage progressed up to 50 turns; and, (**c**) final result of robot path.

In the process of simulation, the robot is assumed as a point and to be always positioned at the center of a cell. A cell is considered to be an obstacle cell if the cell contains even a part of obstacle. Since the water sink model generates a path on the grid metric, there are some stairway paths or sharp turns. Even though the robot is assumed as a point in this implemented simulator, these stairway paths or sharp turns may present another problem deriving from the kinodynamic characteristics of the robot. In this case, some additional path smoothing techniques can resolve it [21,22].

4.1.3. Several Other Results

To confirm the accuracy and practicality of the water sink model, some sample maps are generated with 10 (pixels) size and 60 × 60 cells. The drainage water is 1000 per turn. The experimental results of the water sink model are shown in Figure 12.

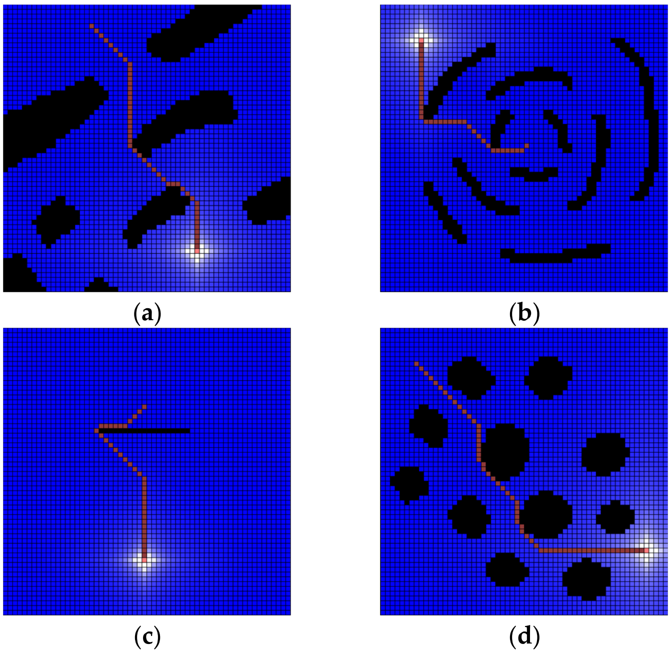


Figure 12. Several results of water sink model. The cell size is 10 (pixels) and 60 × 60 cells map. The drainage water is 1000 per turn. All experiments show that the robot (pink-colored cell) reached the plughole (white-colored cell) with the shortest path on the grid metric. As a result, the plughole is obscured by the robot: (**a**) test map-1 for water sink model; (**b**) test map-2 for water sink model; (**c**) test map-3 for water sink model; (**d**) test map-4 for water sink model.

The several other results of the potential field are shown in Figure 13. In this experiment, the source maps are the same as the water sink model. However, there are no results of reaching to the goal in potential field method. The parameter ξ has variation from 0.01 to 0.02, η has variation from 1.0 to 2.5, and ρ_0 has variation from 10.0 to 155.5.

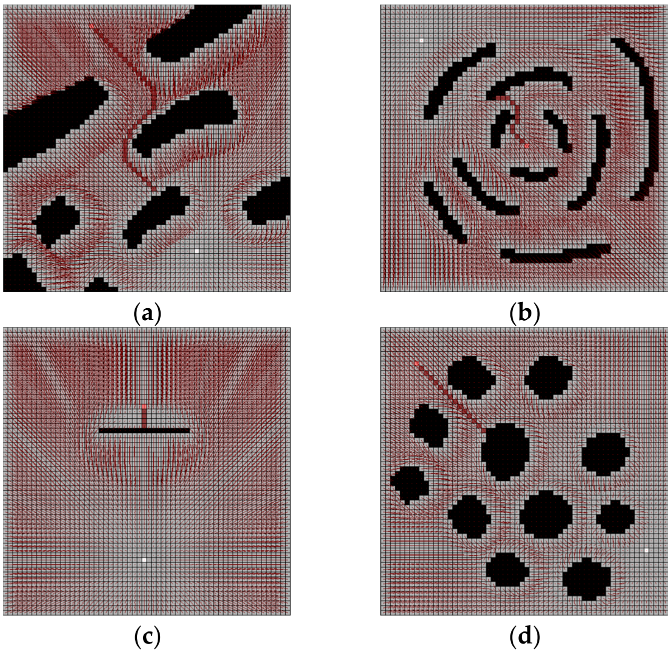


Figure 13. Several results of potential field method with the same map of water sink model in Figure 12: (**a**) test map-1 for potential field method; (**b**) test map-2 for potential field method; (**c**) test map-3 for potential field method; (**d**) test map-4 for potential field method.

The potential field method has three parameters; ξ, η and ρ_0. Figure 14 shows the results of the variation of parameters. The grid map is the same as Figures 12a and 13a. Here, the parameter value that is considered to be sufficiently small and large is respectively set. The parameters should be empirically found with the combination of each parameter characteristic (see Table 1).

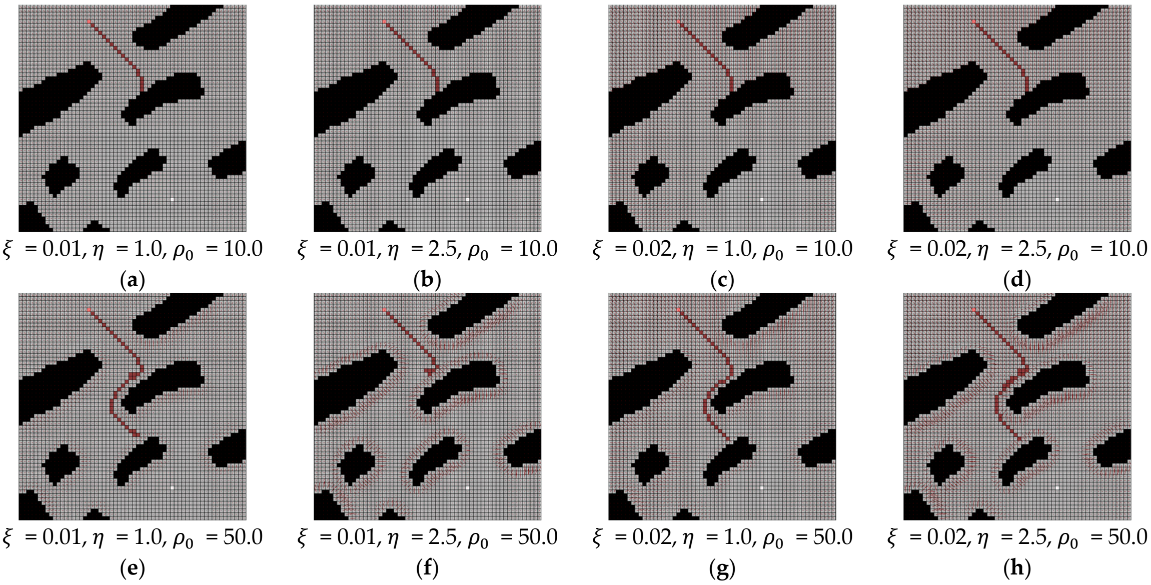

ξ = 0.01, η = 1.0, ρ_0 = 10.0
(**a**)
ξ = 0.01, η = 2.5, ρ_0 = 10.0
(**b**)
ξ = 0.02, η = 1.0, ρ_0 = 10.0
(**c**)
ξ = 0.02, η = 2.5, ρ_0 = 10.0
(**d**)
ξ = 0.01, η = 1.0, ρ_0 = 50.0
(**e**)
ξ = 0.01, η = 2.5, ρ_0 = 50.0
(**f**)
ξ = 0.02, η = 1.0, ρ_0 = 50.0
(**g**)
ξ = 0.02, η = 2.5, ρ_0 = 50.0
(**h**)

Figure 14. The results of potential field method with parameter variations.

Table 1. The characteristics of parameters for potential field method.

Parameters	ξ	η	ρ_0
small value	The robot could not move from the beginning	The robot could move around obstacles	Fail to reflect overall obstacle situation
large value	The robot could stop when encountering an obstacle	The robot could move to opposite to the goal	The robot could be affected by obstacles in distant

4.1.4. Water Sink Model for Dynamic Environment

For the application of the water sink model in a dynamic environment, two additional assumptions are required. The first assumption is that the robot knows the goal position. The second assumption is that the robot can store the obstacle location whenever the robot encounters or find out some obstacles. It is also assumed that the entire outside of the sensing area of the robot is already filled with water. The robot could move using the water sink model after drainage. The drainage will occur again and the obstacle location will be memorized in the robot when the robot encountered some obstacles after movement. Finally, the robot could repeatedly reach the goal with the above process. Figure 15 shows a simple example of simulation for the application to dynamic environment. There is a 7 × 4 sized obstacle and the sensing area of the robot is assumed as 7 × 7 cells around the robot (see Figure 15a). The water drainage occurred once at the first time as the determination of the robot path. However, the sensor of robot could not detect any obstacle the first time (see Figure 15b). After a few movements of robot, the robot finds out some obstacles (see Figure 15c). However, the obstacles do not yet interfere with the robot path. Therefore, the robot continues to move. In Figure 15d, the robot encounters the obstacle that disturbs the determined robot path. At this time, the water drainage occurred again based on the current information of the robot with obstacles. In Figure 15e, the obstacle disturbs the robot path and the water drainage occurred again at this time. Finally, the robot modifies the path and it could reach the plughole (see Figure 15f).

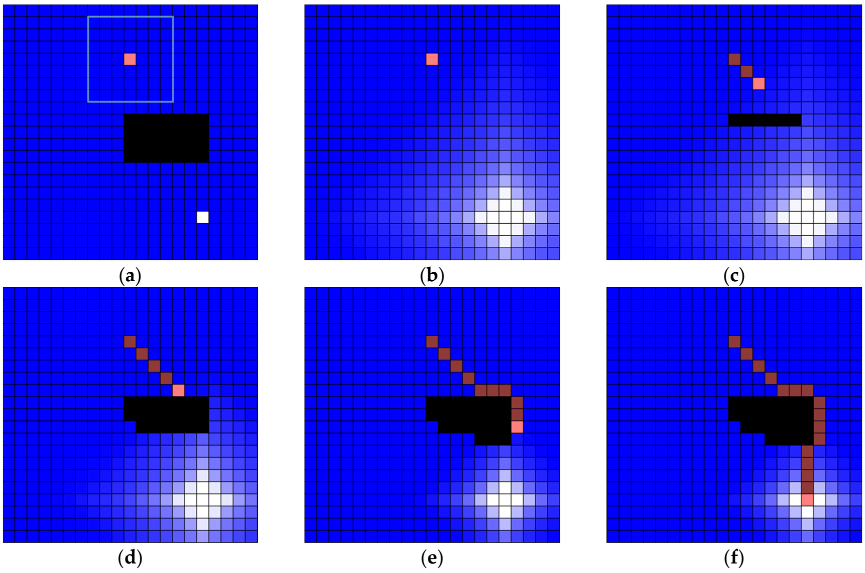


Figure 15. An example of simulation for dynamic environment. Workspace includes 21 × 21 cells whose size is 20 pixels: (**a**) there is a 7 × 4 sized obstacle and the sensing area of robot is 7 × 7 cells; (**b**) the sensor of robot could not detect any obstacle and water drainage is occurred at first; (**c**) the obstacle could not interfere with the robot path; (**d**) the robot path is disturbed by an obstacle and water drainage is occurred again; (**e**) the robot path is disturbed by obstacle again and water drainage is occurred again; and, (**f**) the final result of water sink model.

Figure 16 shows another example for the application to the dynamic environment. The map is more complex than Figure 15 and the same as Figure 12a. The sensing area of the robot is assumed as 11 × 11 cells around the robot (see Figure 16a). The water drainage occurred once at the first time (see Figure 16b) and the robot moves to the plughole without recognition of any obstacle at first. However, whenever the robot encounters an obstacle at the next robot position, local water drainage occurred again, from the plughole, not to the whole area, but to the current robot position (see Figure 16c–f). Figure 16g,h shows that the robot could finally reach the plughole in the dynamic environment. Here, the water drainage totally occurred five times and the path length is 62.

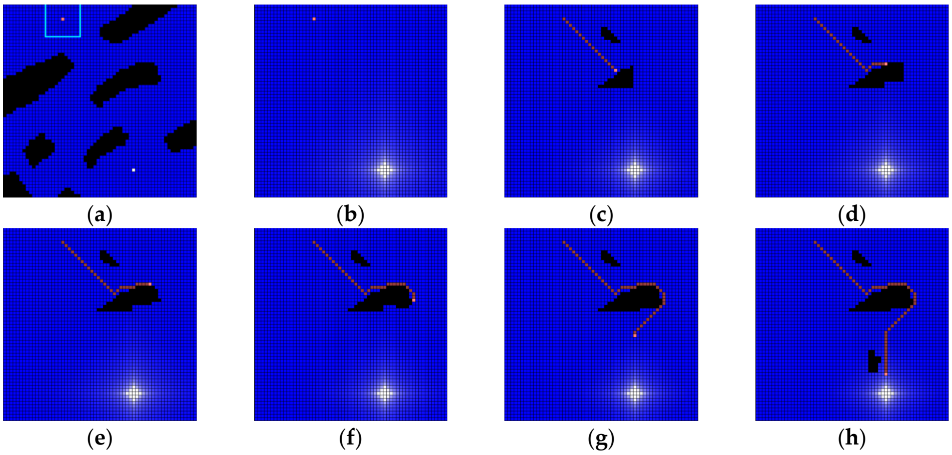


Figure 16. Another example of simulation for dynamic environment. Workspace includes 60 × 60 cells whose size is 10 pixels: (**a**) there are some obstacles and the sensing area of robot is 11 × 11 cells; (**b**) the sensor of robot could not detect any obstacle and water drainage is occurred at first; (**c**) the robot encounters some obstacles at first; (**d**–**f**) the robot path is disturbed by an obstacle and water drainage is occurred again; (**g**) the robot escapes the obstacle area; and, (**h**) some obstacles are found, but there is no interference with the robot path. Finally, the robot could reach the plughole.

4.2. Comparision with Potential Field

4.2.1. Experimental Setup for Potential Field Method

For the comparison of the water sink model and the potential field method, the test maps, including various shapes and numbers of obstacles, are needed. However, the local minima exist in almost cases of potential field map. Therefore, some simple maps (see Figure 17) with various parameters set were used. Table 2 shows the parameter set and 12 combinations of parameters are used in each map.

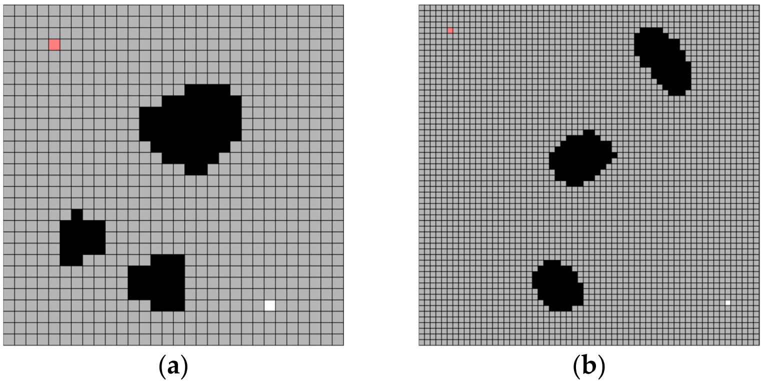


Figure 17. The potential field maps for comparison with water sink model: (**a**) map-1: 30 × 30 size map; and, (**b**) map-2: 60 × 60 size map.

Table 2. Parameter set for map-1 and map-2.

Map	ξ	η	ρ_0
map-1	0.05, 0.1	2, 5	50, 100, 150
map-2	0.05, 0.1	2, 3	50, 70, 90

4.2.2. The Results of Water Sink Model

The water sink model has no parameter and the path is deterministic at any case. Figure 18 shows the water sink model results at each map. The robot path length is 23 on map-1 and 57 on map-2. The path lengths of sideways, upside down, and diagonal are all equal to 1.

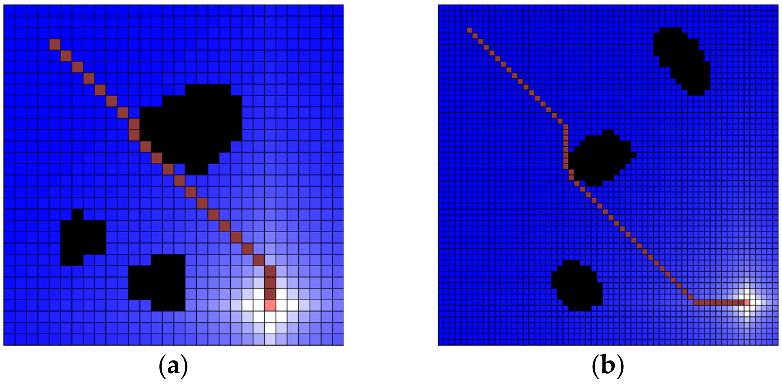


Figure 18. The results of water sink model. The path length is (**a**) map-1: 23, the same as Figure 10c; and, (**b**) map-2: 57.

4.2.3. The Results of Potential Field Method and Comparison

The goal(plughole) reaching of a potential field method is dependent on the parameters (see Figure 19). The simulation is performed on a parameter set. Tables 3 and 4 show the success or failure and the path length of robot on the map-1 and map-2. Table 5 shows the average path length of the potential field when the robot reached the goal (plughole) successfully, the path length of water sink model and the path length ratio.

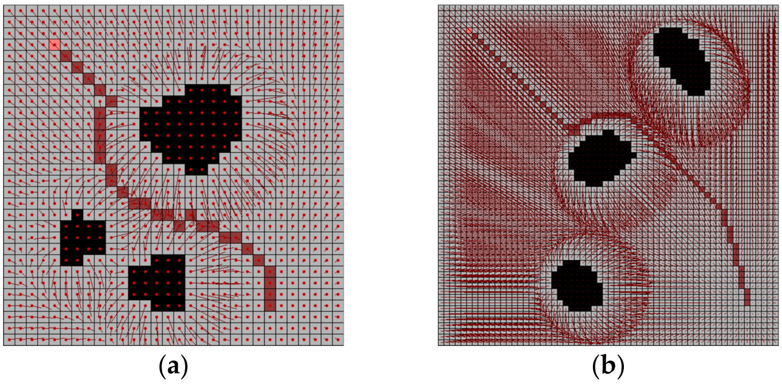


Figure 19. The results of water sink model. The path length is: (**a**) map-1: 23; and, (**b**) map-2: 57.

Table 3. Path length on map-1 for potential field method.

ξ	η	ρ_0	Result/Path Length
0.05	2	50	NR [1]
0.05	2	100	23
0.05	2	150	26
0.05	5	50	23
0.05	5	100	25
0.05	5	150	29
0.1	2	50	NR
0.1	2	100	NR
0.1	2	150	23
0.1	5	50	NR
0.1	5	100	23
0.1	5	150	27

[1] Not Reached.

Table 4. Path length on map-2 for potential field method.

ξ	η	ρ_0	Result/Path Length
0.05	2	50	NR [1]
0.05	2	70	NR
0.05	2	90	NR
0.05	3	50	59
0.05	3	70	NR
0.05	3	90	NR
0.1	2	50	NR
0.1	2	70	NR
0.1	2	90	60
0.1	3	50	NR
0.1	3	70	59
0.1	3	90	63

[1] Not Reached.

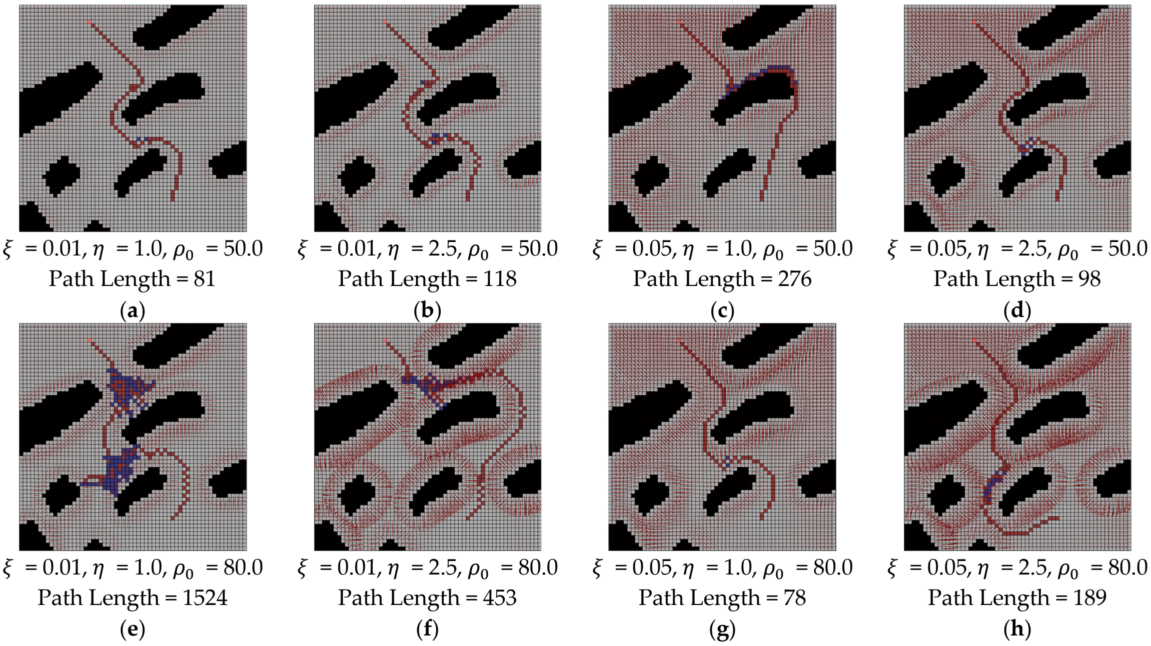
Table 5. Comparison of water sink model with potential field method. Averaged path length of potential field, if successful, deterministic path length of water sink model and path length ratio. The unit is cell.

Map	Average Path Length of Potential Field (PF)	Path Length of Water Sink Model (WSM)	Path Length Ratio (WSM/PF)
map-1	24.875	23	0.9246
map-2	60.25	57	0.9461

4.3. Comparision with Potential Field Using Random Walk

It is easily found that one or more local minima always occur in the potential field method if the complexity of the map is more than a certain level. To overcome this problem, the random walk method could be applied in addition to the potential field method. The random walk method is one of the probability-based techniques. Although the random walk technique does not guarantee processing time and path length of the robot, but the robot reached the goal in most times of our experiments. When the robot encounters some local minima, the robot randomly determines a direction and it moves to escape the local minima as pre-defined length. In our experiment, the pre-defined length is assumed as five cells and the direction is randomly determined without consideration of obstacles and goal location.

Figure 20 shows the results of potential field with random walk as various parameter changes. Each experiment was performed 10 times and the path length is selected as the shortest one. Here, the purple colored cell means the trace of random walk. Unfortunately, it is very hard to find some tendency of parameter change effects, as in Figure 20. For example, the path length was increased in Figure 20c when compared with Figure 20a, as ξ increases when all other parameters are the same, but the path length was decreased in Figure 20d as compared with Figure 20b as ξ increases. Similarly, the effects of parameter η and ρ_0 also show no tendency. In addition, the variation of path length as the parameter changes was high.



ξ = 0.01, η = 1.0, ρ_0 = 50.0
Path Length = 81
(**a**)

ξ = 0.01, η = 2.5, ρ_0 = 50.0
Path Length = 118
(**b**)

ξ = 0.05, η = 1.0, ρ_0 = 50.0
Path Length = 276
(**c**)

ξ = 0.05, η = 2.5, ρ_0 = 50.0
Path Length = 98
(**d**)

ξ = 0.01, η = 1.0, ρ_0 = 80.0
Path Length = 1524
(**e**)

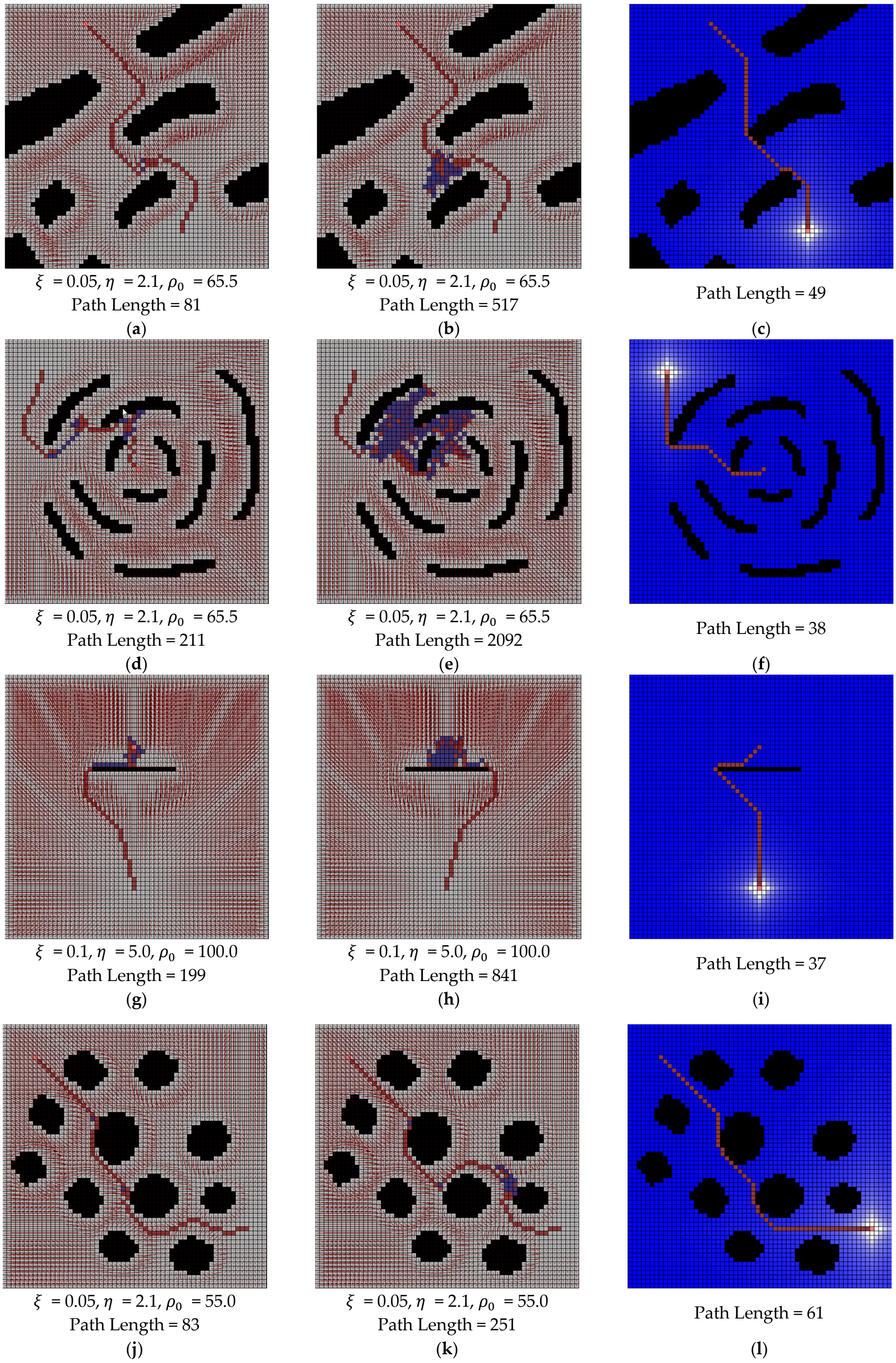
ξ = 0.01, η = 2.5, ρ_0 = 80.0
Path Length = 453
(**f**)

ξ = 0.05, η = 1.0, ρ_0 = 80.0
Path Length = 78
(**g**)

ξ = 0.05, η = 2.5, ρ_0 = 80.0
Path Length = 189
(**h**)

Figure 20. The results of potential field method using random walk with parameter variations.

For the comparison with potential field using the random walk and water sink model, some experiments were performed in the same maps that were used in Figures 14 and 15. The parameters of the potential field method with random walk were $\xi = 0.05$, $\eta = 2.1$, $\rho_0 = 65.5$ for Figure 21a,b,d,e, $\xi = 0.1$, $\eta = 5.0$, $\rho_0 = 100.0$ for Figure 21g,h and $\xi = 0.05$, $\eta = 2.1$, $\rho_0 = 55.0$ for Figure 21j,k. These parameters were empirically selected by repetitive experiments that consider some parameter range. The used ranges of parameters were [0.01, 0.1] for ξ, [1.0, 5.0] for η and [40.0, 150.0] for ρ_0. The results of the first column in Figure 21 are the case of shortest path length and the results of the second column in Figure 21 are the case of the longest path length in our experiments on each map. The results of the last column in Figure 21 are the result of the water sink model method on each map. Each path length is described in the caption of Figure 21.



ξ = 0.05, η = 2.1, ρ_0 = 65.5
Path Length = 81
(**a**)

ξ = 0.05, η = 2.1, ρ_0 = 65.5
Path Length = 517
(**b**)

Path Length = 49
(**c**)

ξ = 0.05, η = 2.1, ρ_0 = 65.5
Path Length = 211
(**d**)

ξ = 0.05, η = 2.1, ρ_0 = 65.5
Path Length = 2092
(**e**)

Path Length = 38
(**f**)

ξ = 0.1, η = 5.0, ρ_0 = 100.0
Path Length = 199
(**g**)

ξ = 0.1, η = 5.0, ρ_0 = 100.0
Path Length = 841
(**h**)

Path Length = 37
(**i**)

ξ = 0.05, η = 2.1, ρ_0 = 55.0
Path Length = 83
(**j**)

ξ = 0.05, η = 2.1, ρ_0 = 55.0
Path Length = 251
(**k**)

Path Length = 61
(**l**)

Figure 21. The results of comparison between water sink model and potential field with random walk: (**a**,**b**,**d**,**e**) parameters are $\xi = 0.05$, $\eta = 2.1$ and $\rho_0 = 65.5$; (**g**,**h**) parameters are $\xi = 0.1$, $\eta = 5.0$ and $\rho_0 = 100.0$; (**j**,**k**) parameters are $\xi = 0.05$, $\eta = 2.1$ and $\rho_0 = 55.0$; (**c**,**f**,**i**,**l**) are the results of water sink model.

In the results of the potential field method, it could be easily found that the path length could be too much large, even though the complexity of map is simple. In addition, there is a tendency to renew the maximum path length whenever more experimental results are updated. On the other hand, the path length of water sink model is always smaller than the path length of the potential field method and its value remains constant.

5. Conclusions

In this paper, we propose a water sink model to resolve the local minima problem found in previous potential field methods for path planning. The water in the field drains from a higher location to a lower one that is affected by gravity. Basically, the proposed method mimics the draining of water from a sink with a plughole. As a result, when compared to the potential field, the proposed method achieved a shorter path with low clearance from obstacles and without local minima. In addition, it is beneficial for robot path planning, since it does not require an additional controller similar to potential field methods.

The water sink model is provided to resolve the local minima problem of potential field method, but there are some considerations in the application of the water sink model to real world problems. The first one is that the robot is assumed as a point in water sink model. Commonly, the robot has own some mechanical and kinodynamic characteristics. Therefore, the post-processing for path smoothing is additionally required when considering the kinodynamic characteristics of the robot after path planning. The second one is that the obstacle is approximated as a rectangle with sharp edge, because the water sink model is based on the grid cell. In the water sink model, the obstacle region is recognized and processed, as it is greater than or equal to the actual obstacles. Since the size of the cell may not be sufficiently small, the path may be clogged due to an over-approximation of the obstacle, even though there is actually empty space for the robot to pass through. On the other hand, if the size of the cell is small enough, the resolution of representation for the problem space may be improved, but the time for processing will be increased enormously. For the same reason that the obstacle is approximated as a grid cell, the robot is moving along the obstacles when the robot encounters the obstacle. Because of this, the clearance issue should be considered. The third one is that, since the water sink model is based on grid cell space, grid cells also present the obstacles and the sequence of path. To more accurately improve the representation of the real world, the hexagonal cell or triangular mesh could be considered for the representation of the real problem space. Especially, applying the irregular triangle mesh that can more accurately represent the real world and reduce the processing time like the non-uniform cell decomposition method.

Nonetheless, the water sink model is very simple and intuitive in finding the path. Additionally, the water sink model is a kind of global path planning method and it could always find the shortest path in the workspace.

Author Contributions: Conceptualization, G.-Y.J.; methodology, G.-Y.J.; software, G.-Y.J.; formal analysis, G.-Y.J. and J.-W.J.; writing—original draft preparation, G.-Y.J.; writing—review and editing, G.-Y.J. and J.-W.J.; visualization, G.-Y.J.; project administration, J.-W.J.

Funding: This research was partially supported by the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korean government (MOTIE: Ministry of Trade, Industry & Energy, HRD Program for Embedded Software R&D) (No. N001884) and partially supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (2015R1D1A1A09061368).

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

References

1. Lumelsky, V.; Stepanov, A. Path planning strategies for point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica* **1987**, *2*, 403–430. [CrossRef]
2. Asano, T.; Asano, T.; Guibas, L.; Hershberger, J.; Imai, H. Visibility-polygon search and euclidean shortest paths. In Proceedings of the 26th Annual Symposium on Foundations of Computer Science, Portland, OR, USA, 21–23 October 1985.

3. Osamu, T.; Schilling, R.J. Motion Planning in a Plane Using Generalized Voronoi Diagram. *IEEE Trans. Robot. Autom.* **1989**, *5*, 143–150.
4. Keil, J.M.; Sack, J.R. Minimum decomposition of polygonal objects. *Comput. Geom.* **1985**, *2*, 197–216.
5. Kavraki, L.E.; Kolountzakis, M.N.; Latombe, J.C. Analysis of Probabilistic Roadmap for Path Planning. In Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, 22–28 April 1998; Volume 14, pp. 166–171.
6. Bohlin, R.; Kavraki, L.E. Path Planning Using Lazy PRM. In Proceedings of the 2000 IEEE International Conference on Robotics & Automation, San Francisco, CA, USA, 24–28 April 2000; pp. 521–528.
7. LaValle, S.M.; Kuffner, J.J. Randomized Kinodynamic Planning. *Int. J. Robot. Res.* **2001**, *20*, 378–400. [CrossRef]
8. Karaman, S.; Frazzoli, E. Incremental Sampling-based Algorithms for Optimal Motion Planning. In Proceedings of the Robotics: Science and Systems 2010, Zaragoza, Spain, 27–30 June 2010; pp. 267–274.
9. Gammell, J.D.; Srinivasa, S.S.; Barfoot, T.D. Informed RRT*: Optimal Sampling-based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 2997–3004.
10. Mac, T.T.; Copot, C.; Tran, D.T.; Keys0er, R.D. Heuristic approaches in robot path planning: A survey. *Robot. Auton. Syst.* **2016**, *86*, 13–28. [CrossRef]
11. Khatib, O. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE J. Robot. Autom.* **1987**, *3*, 43–53. [CrossRef]
12. Warren, C.W. Global path planning using artificial potential fields. In Proceedings of the 1989 International Conference on Robotics and Automation, Scottsdale, AZ, USA, 14–19 May 1989; Volume 1, pp. 316–321.
13. Carpin, S.; Pillonetto, G. Motion planning using adaptive random walks. *IEEE Trans. Robot.* **2005**, *21*, 129–136. [CrossRef]
14. Palmieri, L.; Rudenko, A.; Arras, K.O. A Fast Random Walk Approach to Find Diverse Paths for Robot Navigation. *IEEE Robot. Autom. Lett.* **2017**, *2*, 269–276. [CrossRef]
15. Qureshi, A.H.; Iqbal, K.F.; Qamar, S.M.; Islam, F.; Ayaz, Y.; Muhammad, N. Potential Guided Directional-RRT* for Accelerated Motion Planning in Cluttered Environments. In Proceedings of the 2013 IEEE International Conference on Mechatronics and Automation, Takamatsu, Japan, 4–7 August 2013; pp. 519–524.
16. Amiryan, J.; Jamzed, M. Adaptive Motion Planning with Artificial Potential Fields Using a Prior Path. In Proceedings of the 2015 3rd RSI International Conference on Robotics and Mechatronics, Tehran, Iran, 7–9 Octomber 2015; pp. 731–736.
17. Daily, R.; Bevly, D.M. Harmonic Potential Field Path Planning for High Speed Vehicles. In Proceedings of the 2008 American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 4609–4614.
18. Yang, Y.; Zhang, Y.; Li, X. An improved artificial potential field algorithm based on nonuniform cell decomposition. In Proceedings of the 2017 6th International Conference on Measurement, Instrumentation and Automation, Zhuhai, China, 29–30 June 2017; pp. 735–739.
19. Pal, A.; Tiwari, R.; Shukla, A. A focused wavefront algorithm for mobile robot path planning. In Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, Wroclaw, Poland, 23–25 May 2011; pp. 190–197.
20. Nooraliei, A.; Nooraliei, H. Path planning using wave front's improvement methods. *Comput. Technol. Dev.* **2009**, *1*, 259–264.
21. Gonzalez, D.; Perez, J.; Milanes, V.; Nashashibi, F. A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1135–1145. [CrossRef]
22. Ravankar, A.; Ravankar, A.; Kobauashi, Y.; Hishino, Y.; Peng, C.-C. Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges. *Sensors* **2018**, *18*, 3170. [CrossRef] [PubMed]