robotics

MDPI

Article

Investigating the Impact of Triangle and Quadrangle Mesh Representations on AGV Path Planning for Various Indoor Environments: With or Without Inflation

Ahmadreza Meysami *,†, Jean-Christophe Cuillière †, Vincent François † and Sousso Kelouwani †

Mechanical Engineering Department, Université du Québec à Trois-Rivières, Trois-Rivieres, QC G8Z 4M3, Canada; jean-christophe.cuilliere@uqtr.ca (J.-C.C.); vincent.francois@uqtr.ca (V.F.); sousso.kelouwani@uqtr.ca (S.K.)

* Correspondence: ahmadreza.meysami@uqtr.ca

† These authors contributed equally to this work.

Abstract: In a factory with different kinds of spatial atmosphere (warehouses, corridors, small or large workshops with varying sizes of obstacles and distribution patterns), the robot's generated paths for navigation tasks mainly depend on the representation of that environment. Hence, finding the best representation for each particular environment is necessary to forge a compromise between length, safety, and complexity of path planning. This paper aims to scrutinize the impact of environment model representation on the performance of an automated guided vehicle (AGV). To do so, a multi-objective cost function, considering the length of the path, its complexity, and minimum distance to obstacles, is defined for a perfect circular robot. Unlike other similar studies, three types of representation, namely quadrangle, irregular triangle, and varying-size irregular triangle, are then utilized to model the environment while applying an inflation layer to the discretized view. Finally, a navigation scenario is tested for different cell decomposition methods and an inflation layer size. The obtained results indicate that a nearly constant coarse size triangular mesh is a good candidate for a fixed-size robot in a non-changing environment. Moreover, the varying size of the triangular mesh and grid cell representations are better choices for factories with changing plans and multi-robot sizes due to the effect of the inflation layer. Based on the definition of a metric, which is a criterion for quantifying the performance of path planning on a representation type, constant or variable size triangle shapes are the only and best candidate for discretization in about 59% of industrial environments. In other cases, both cell types, the square and the triangle, can together be the best representation.

Keywords: AGV; robotics mapping; mesh representation; path planning

check for updates

Citation: Meysami, A.; Cuillière, J.-C.; François, V.; Kelouwani, S. Investigating the Impact of Triangle and Quadrangle Mesh Representations on AGV Path Planning for Various Indoor Environments: With or without Inflation. *Robotics* **2022**, *11*, 50. https://doi.org/10.3390/robotics11020050

Academic Editors: Giovanni Boschetti and João Miguel da Costa Sousa

Received: 18 February 2022
Accepted: 10 April 2022
Published: 13 April 2022

1. Introduction

Automated guided vehicles (AGVs) have been used for several decades [1,2]. They mostly rely on predefined routes to move from one point to another. However, the flexibility and agility introduced by industry 4.0 [3] are pushing these technologies towards a complete autonomous navigation capability [4]. The required autonomy forces the AGVs to know the environment prior to moving intelligently and safely. Therefore, one of the most important tasks for these vehicles is to model the navigation environment as accurately as possible [5]. In indoor navigation on a planar floor of a factory, a 2D model of the environment, including the specifications of the obstacle-free area, is required to perform path planning. Abstracting, encoding, and saving this type of information is called 'mapping' in mobile robots [6]. A mobile robot uses this map repeatedly to find a path between the current position and the goal position. The essential information for constructing the environment model is usually obtained from raw sensor data with the help of data structures. There are different types of data structures for saving and reading necessary spatial information in navigation. After saving the information, path planning uses this information via queries (by means of the data structure) to perform a task. Therefore, the data structure types, spatial models, and the resulting map all play decisive roles in the performance of the global planners.

One of the most well-known mapping methods, which has gained special attention due to its simplicity and robustness, is the occupancy grid technique [7]. This method is composed mainly of uniform grids with simple square shapes, which are made cell by cell, independently. Other shapes such as triangles (with application in polygonal mesh [8], reconstruction [9], etc.) can be used for the discretization of the environment. Squares and triangles are the most famous cell types in a planar application. The analogy between the occupancy grid map and a triangular mesh is the creation of a discretized environment model using the cell concept. Their difference, though, lies on the shape and number of cells, as well as their connection rules (topology) in forming a complete and accurate free space model. By utilizing such a model, a global path planner, e.g., cell decomposition method [10], can be used to find an efficient path between two points.

The efficiency of the path planning algorithm is highly dependent on the environment model, which in turn is impacted by the representation type (cell's shape, number, and topology). For instance, according to the number of cells, it may take a shorter or a longer time to find a path. Moreover, the realized path may have different lengths and distances to the obstacles considering the cell shape and size in the environment. An efficient path should consider all of these essential parameters (such as length, distance, and computational time) [11]. Another aspect worth noting is that, when the conditions of an environment change (new distribution of obstacles), a new representation (discretized model) may lead to a more efficient path. However, investigating the influence of environment model representation on generating a suitable path has escaped the attention of many researchers. Works such as [12,13] have tried to define the environment's difficulty based on parameters such as obstacle shape and distribution or distance to the nearest obstacles. Nevertheless, they did not consider the effect of a navigation scenario (position of start and goal points) and map representation, and they did not propose any solution to consider this difficulty. Recently, valuable work such as that of [14] tried to relate this difficulty to the robot's traversal time in navigation benchmarking. However, this work focused on the overall performance of the navigation platform and does not consider the global planner effect. In a similar task, [14] only considers the effect of cell shape and size on computational efficiency and does not consider their effect on the path length and safety. In addition, ref. [13] defines some metrics and compares different path planning methods only on grid-type cells. Other works such as [15], which uses an adaptive mesh for risk-aware path planning, do not mention important parameters such as distance to the nearest obstacle and the effect of robot size. When the environment is cluttered, to our knowledge, there is no straightforward method to select the most appropriate navigation environment approach. For consideration of robot size in path planning, there are various methods. The first strategy applies offset to the boundary of the obstacles, then uses this scaled view as an input for discretization [16]. This strategy is unsuitable when it is necessary to modify the map according to changes in the environment or when the real model of the obstacles is necessary for manipulation. The second strategy is used in the game industry (famous as navigation mesh [17]) and checks each triangle for the input and output edge's length as a possibility of robot passage [18,19]. The third strategy is map generation based on the actual boundary, then applying the inflation layer on the discretized view (the offset or inflation layer is the robot's radius plus any optional value for safety, such as the method implemented in the ROS (Robot operating System) [20]). In addition, it is usual to build an inflated map on top of the metric map in robotic applications [21]. Nevertheless, no specific study has been carried out to select the best environment representation based on the effect of this inflation. Therefore, the main contributions of this paper are:

- Showing that the type of environment representation has a direct impact on path planning, and quantifying this by a metric.

- Presenting the results showing the pros and cons of each representation when considering the inflation layer

By considering the points discussed, this paper explores the effect of the environment representation on the performance of an AGV. In this regard, a multi-objective cost function is developed, and three kinds of cell decomposition (quadrangle, irregular triangle, and varying-size irregular triangle) are used for environment modeling. Furthermore, the robot size is considered through an inflation layer imposed on the model. To the best of the authors' knowledge, this is one of the first attempts to explore the impact of different cell decomposition techniques and robot size on the performance of an AGV. A specific platform is used to generate the three representations with different applicable sizes [22]. Then, a performance comparison is performed for each representation using a fixed scenario obtained by a global planner [13,14].

The rest of the paper is organized as follows. Section 2 depicts the problem formulation and assumptions. Section 3 introduces some common concepts concerning cells and neighbors regarding all the representations. In Section 4, the methodology for all representations is explained by a mathematical formulation. Section 5 discusses all the results, corresponding to the simulation tests on various environments and representations. Section 6 summarizes the outcomes and puts forward future directions.

2. Problem Formulation and Assumption

An AGV (Automated guided vehicle), denoted by A, navigates in a factory as workspace W_s. The workspace contains obstacles, B_i, with $i \in \{1, 2, 3, \ldots, N_o\}$ and N_o, the number of obstacles. The obstacle boundaries ($\Gamma(B_i)$) are generated by a LIDAR (Light Detection and Ranging) placed on the AGV [23]. These can be polygonal or curvilinear curves. The W_s is divided into W_B, the union of all obstacles, and W_{free}, the free space. S and G are the start point and the goal point of A, respectively, in the W_{free}. According to the distribution of B_is and the size of A, various paths (τ) can be traversed to get to G from S. From an economic point of view, τ must be realized quickly, be as short as possible, and keep a safe distance to B_is. If τ does not satisfy one of these properties, AGV cannot be used reliably for a long time. In this regard, an optimal path (τ_{opt}) is the one that has the best of the properties mentioned. It is typically determined by a global planner with the help of encoding W_{free} to an environment map (M_{reps}). The subscript "reps" denotes the encoding types or representations, which have a significant impact on finding τ_{opt}.

This study assumes a perfect circular robot without any sensor noise and environment modeling uncertainty, in order to be able to map spatial information into various representation formats. The index for comparing different representations is a multi-objective cost function (f) which takes the length of τ_{opt}, the complexity of finding the τ_{opt} (correlated with running time), and minimum distance to obstacles into account [24]. Furthermore, no post-processing on the generated path, such as path smoothing [25], has been performed, since this work studies the behavior of various representations on a global planner (as in the deliberative approach concept, in which information is complete before starting the navigation) [26]). The next section discusses common concepts regarding the environment representation types used in mapping and navigation.

3. Environment Modelling for Navigation

The simplest required information to perform navigation in an environment is the W_{free}. In this regard, one of the best-known methods is cell decomposition, in which W_{free} is converted into digitized cells or a graph of cells. Various shapes can be used for the discretization of the environment. In the planar application, squares and triangles are the most famous cell types. Another practical cell type is the irregular triangular mesh (ITM), used extensively in computer graphics and gaming [17,18] as well as outdoor robotics applications [27]. Using these cells for navigation needs the definition of the neighbor types as either common-edge and common-node neighbors. In a quadrangle (QUAD) cell in a general condition (far from the obstacle), the number of common-edge neighbors and common-node neighbors is four and four. However, in an ITM, these numbers are generally three for common-edge neighbors or a variable number, depending on the degree of nodes.

As is clear, a QUAD cell with the edges parallel to the global frame of an environment cannot be flexible enough to show the boundary of obstacles in the environment except for shapes parallel with the global frame. This limitation causes some inaccuracy in the definition of the area around obstacles and presents limitations to accessing this area. Solving this problem requires a fine square, which, on the other hand, increases the computational burden. An ITM cell, with the flexibility of an edge's direction, can accurately show the boundaries of obstacles. Nevertheless, their irregular shape requires the saving of topological and geometrical information, which makes the map larger. Therefore, each cell shape has its own pros and cons. These features can affect the performance of the algorithm that uses the encoded information of these cells.

The cell decomposition method takes a point as the representative of a cell in a navigation task in order to model the W_{free}. The cell's center can be a simple selection for square cells, in order to calculate properties such as the nearest distance to an obstacle, or as a reference for the position of cells and for moving between cells. After obtaining a representative for each cell, a path can be defined as the lines connecting the start and goal points through the center of cells, if they exist. For a fair comparison between QUAD and ITM, the center of the triangle is selected as the representative of each cell in this study (other methods like that of [28] consider the middle of the edges, with lower computational time but with a longer path). Optimal searching in this graph is performed according to criteria such as the cost and the heuristic of each cell with respect to start and goal points. Since the building block of this graph is a cell, the selection of its size is an effective parameter for the construction and searching of the graph. Small cells give more information than large cells after digitizing, but increase the computational effort of the agent. In addition to cell size, the rules of neighboring, which are different for each cell shape, can change the graph. Hence, finding a good path presents the challenge of finding contradictory parameters. The selection of a suitable cell shape and size can help find a reasonable solution. In the next section, the methodology for studying the effects of these various representations will be described.

4. Methodology

The performed analysis in this study is divided into two parts. First, the effect of representation is considered regardless of the robot size. Second, the robot size is considered using the concept of inflation layer. This concept generates the map based on the real boundary. Then, it applies an inflation layer to the discretized view. In fact, the offset or inflation layer is the radius of the robot plus any optional value for safety [20]. This strategy has already been implemented in the ROS (Robot Operating System) [20]. In what follows, the decomposition methods are described.

4.1. Pure Decomposition

The geometry and the physical size of the robot are not considered in this section. Furthermore, the discretized cell is general and does not depend on the specific type of representation. For navigation on a planar surface by cell decomposition method, it is necessary to encode and abstract the environment information into a map representation, M_{reps}.

$$M_{reps} = \{\text{All } c_i \ (i = 1, 2, \ldots N)\ ; \ \sum c_i \approx W_{free}\} \tag{1}$$

where c_i denotes the i^{th} cell and implies that the union of all discretized cells (finite number of N) can approximately estimate the free space W_{free}. Each path between S and G points in the map is defined as τ, which is a set of cells:

$$\tau = \{c_s, c_1, c_2, \ldots, c_g\}, g \leq N \tag{2}$$

A set of cells as a path is divided into several segments S_{ij} between successive cells c_i and c_j in τ. This leads to:

$$\tau = \{S_{s1}, S_{12}, S_{23}, \ldots, S_{ij}, \ldots, S_{ig}\}, g \leq N, i = g - 1 \tag{3}$$

where S_{ij} is generally a curve connecting two cells (c_i and c_j) together with all the possible points inside the cells. This segment will be a building unit of movement for AGV according to its controller. So, each movement between c_i and c_j is a line segment between their centers. However, linear movement between the centers of two neighbors does not always guarantee safety near the obstacles. Therefore, it is necessary to check the feasibility of movement for some cells near obstacles. To do so, $\mathrm{Mov}(c_i, c_j)$ is defined as:

$$\mathrm{Mov}(c_i, c_j) = \begin{cases} \text{True if } \forall p \in \mathrm{Line}(\mathrm{cent}(c_i), \mathrm{cent}(c_j)) \implies p \in W_{free} \\ \text{False if } \exists p \in \mathrm{Line}(\mathrm{cent}(c_i), \mathrm{cent}(c_j)) \implies p \notin W_{free} \end{cases} \tag{4}$$

If $\mathrm{Mov}(c_i, c_j)$ is true, it shows that the movement between the centers of c_i and c_j is feasible. Otherwise, there is the possibility of collision between agents and obstacles. In the geometrical view, if the combination of neighbor's vertices is convex, then the linear segment will be inside the union of cells, and the movement is safe. Nevertheless, if the combination of c_i and c_j's vertices is concave, then the movement is not permitted. This is because it increases the possibility of collision with an obstacle. To include this extra condition into τ:

$$\tau = \{\{c_s, c_1, c_2, \ldots, c_g\}, \forall\ c_i \text{ and } c_j \implies \mathrm{Mov}(c_i, c_j) = \text{True}\} \tag{5}$$

Checking the movement feasibility is simple for QUAD cells as the common-edge neighbor produces a convex shape. However, in ITM, it is possible to have a concave shape even for common-edge neighbors near obstacles. Hence, the movement feasibility needs to be checked in this mesh. This extra checking enhances the computational effort of this mesh type [29].

Among all possible τ, finding τ_{opt} is of great interest. According to the discretized nature of the path, it is possible to convert cells and movement between them into vertices and edges of a graph. Then a graph search algorithm can be used to find τ_{opt} between the start and goal cells. A*, as a popular graph search method [30], is employed in this study. This method uses Euclidean distance for calculating the cost and heuristic criterion between start and goal [31]. The cost of cell $g(c_i)$ is given by:

$$g(c_i) = \sum \| \mathrm{cent}(c_i), \mathrm{cent}(\mathrm{par}(c_i)) \| \tag{6}$$

where $\mathrm{cent}(c_i)$ stands for the center of the cell, Symbol $\| a, b \|$ is the Euclidean distance between point a and point b, which are the center of two successive cells, and $\mathrm{par}(c_i)$ is the parent of c_i. This cost starts from S and, in each sequence, accumulates the Euclidean distance between the current cell and its parent in a graph. The heuristic criterion of the cell ($h(c_i)$) is calculated by:

$$h(c_i) = \| \mathrm{cent}(c_i), \mathrm{goal} \| \tag{7}$$

The result of A* is a τ with the minimum cost between two cells containing start and goal points. This τ is shown by τ_{opt}:

$$\tau_{opt} = \{\{c_s, c_1, c_2, \ldots, c_g\}, \forall\ c_i \text{ and } c_j,$$
$$c_i = \mathrm{par}(c_j) \implies L_{\tau_{opt}} \approx \min(L_\tau) \tag{8}$$

where L_τ denotes the length of a path.

As stated earlier, in this study, the utilized multi-objective cost function is composed of three metrics. These are the complexity of A* algorithm for finding τ_{opt}, length of τ_{opt}, and minimum distance between AGV and obstacles when it follows τ_{opt}.

Herein, the complexity (C_{reps}) is defined by counting the number of visited (In A*, visited cells go to a list named open list, so C_{reps} = size (Open list)) nodes during A*.

$$C_{reps} = \text{Number of visited cells in } A^* \tag{9}$$

The length of the optimal path ($L_{\tau_{opt,\ reps}}$) is calculated as follows:

$$L_{\tau_{opt,\ reps}} = \sum_{i=S}^{i=G} \| \operatorname{cent}(c_i), \operatorname{cent}(\operatorname{par}(c_i)) \| \tag{10}$$

The minimum distance to the nearest obstacle is calculated with the help of the Euclidean distance grid, as explained in [32,33].

$$DT(p) = d_k,\ p \in\ c_{RG,k} \tag{11}$$

where $DT(p)$ is a function for mapping each point of free space p into a cell $c_{RG,k}$ in the reference grid (RG), which is a Euclidean distance transform on a 200 × 100 grid. $c_{RG,k}$ contains the value of distance to the nearest obstacle as d_k.

Therefore, $D_{min,reps}$, as the minimum distance between all agent's distances to the nearest obstacles, is given by:

$$\begin{gathered} D_{min,reps} = \min(DT(\tau)) \\ \text{which } \forall c_i \in \tau \bigwedge \operatorname{cent}(c_i) \in c_{RG,k} \end{gathered} \tag{12}$$

In triangular representation, as shown in Figure 1, there are some free points near obstacles that are not placed inside any cells in the RG. In this situation, the Euclidean distance between the point and other nodes and edges, which are tagged as the obstacle, is calculated. Then, the minimum value is returned as the minimum distance. This value may be smaller and larger than the previous distance value of the path and must be compared to the current minimum distance.

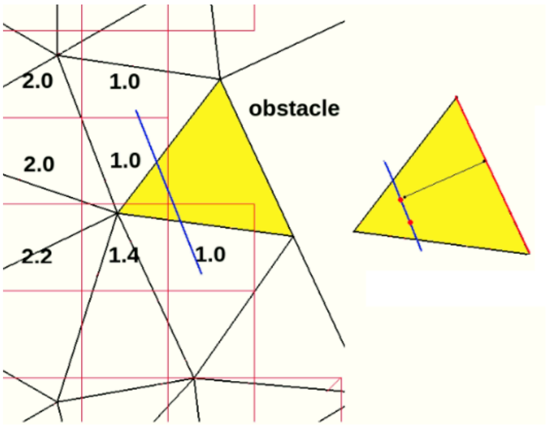

Figure 1. Methodology for calculation of distance when the checking point is not placed inside any cell. 1.0 shows the minimum number in a grid, but some values can be calculated based on the edge of nodes shared with obstacles boundaries.

After calculating the three normalized metrics, a linear weighted sum approach is utilized to define the cost function for each representation.

$$\begin{cases} LN_{reps} = \frac{L_{\tau_{opt,\ reps}}}{L_{max}} \\ CN_{reps} = \frac{C_{reps}}{C_{max}} \\ DN_{reps} = \frac{D_{min}}{D_{min,reps}} \end{cases} \tag{13}$$

where LN_{reps} is the normalized optimal length, CN_{reps} is the normalized complexity for finding optimal length, DN_{reps} is the minimum distance of agent to the nearest obstacle when following the optimal path, $L_{\tau_{opt,reps}}$ is the optimal length obtained by (10), C_{reps} is the calculated complexity from (9), $D_{min,reps}$ is the minimum distance of the optimal path from (12), and L_{max}, C_{max}, and D_{min} are the maximum lengths between start and goal, maximum complexity to find this maximum length path, and the minimum possible distance between agent and obstacle, respectively. Finally, the multi-objective cost function (f_{reps}) is formulated as:

$$f_{reps} = \omega_1 \times \ LN_{reps} + \omega_2 \times \ CN_{reps} + \ \omega_3 \times \ DN_{reps} \tag{14}$$

where $\omega_1, \ \omega_2$ and ω_3 are the weights. For simplicity, equal importance is considered for all weights: ω_1 = ω_2 = ω_3 = 0.333.

4.2. Decomposition of Inflation Layer

In this sub-section, the robot size is integrated into the inflation layer. The inflation layer is defined as follows:

$$\text{Inflation layer } = \{\forall \ p \in W_{free}, \ DT(p) < (r_{AGV} + SD)\} \tag{15}$$

where r_{AGV} is the radius of AGV, and SD is an optional value for a safe distance between the AGV and an obstacle. According to (15), the inflation layer is formed by all the points in W_{free} whose distance to the nearest obstacle is lower than a specific value. With this definition, the resulting map considering this inflation will be:

$$M_{reps} = \{\forall c_i(i = 1, 2, \ldots N); \ \sum c_i \approx W_{free}; \text{cent}(c_i) \not\subset \text{inflation layer}\} \tag{16}$$

5. Results and Discussion

This section presents the results regarding the AGV performance in different conditions. After this, the considered case studies and the utilized platform are first explained. Subsequently, the results concerning environment representation and the inflation layer are discussed.

5.1. Platform for Experiment

The intended comparative study is expected to clarify the best representation (M_{reps}) for each obstacle's distribution (W_s) according to the above-explained multi-objective cost function (f). In this regard, a fixed scenario (a path planning between S and G) is utilized for various M_{reps} and W_s. The following types of representation (M_{reps}) are considered in this study:

- QUAD mesh (GRID Nx × Ny): four cell sizes (GRID 100 × 50, GRID 120 × 60, GRID 160 × 80, and GRID 200 × 100) are considered for this mesh where Nx and Ny are the number of cells in horizontal and vertical directions.
- ITM: four cell sizes (ITM-700, ITM-500, ITM-300, ITM-100) are also considered for this mesh. ITM-100 refers to a triangle with an average of 100 mm edges. For some environments where this was not applicable, larger triangles and smaller ones were used.
- Varying-size ITM (VITM): Concerning this mesh, ITM-100-700 is utilized. This size refers to a triangular mesh with an average edge size of 700 mm, which has been refined to size 100 mm in some places. When confronting narrow passages, VITM-50-700 or VITM-30-700 have also been utilized.

Concerning various factory environments, the following W_s are taken into account (see Figure A1):

- Area with single obstacle: Map_1 (small-size obstacle), Map_2 (medium-size polygonal obstacle), Map_3 (medium-size circular obstacle)
- Space with dividers (Map_4)

- Narrow passage (Map_5)
- Curvilinear passage: Map6
- Area with regular multi-obstacles: Map_7, Map_8 (two groups), Map_9 (different size)
- Area with irregular multi-obstacles: Map_10, Map_11 (two groups), Map_12 (circular), Map_13 (different shapes)
- Hybrid with wide space: Map_14
- Corridor and dividers: Map_15
- Hybrid narrow space environments: Map_16, Map_17 (including all the previous environments)

In this study, a research platform developed by our team (ERICCA, Equipe de Recherche en Integration Cao-CAlcul) is utilized to perform the mesh generation for different environments. This platform is based on the Unified Topological Model [22] and automatic meshing and remeshing tools, as explained in [34–37]. The computer for using this platform and simulation was Intel Core i7-4790 RAM 32GiB. Since the main task is to study the effect of the discretized cell types and sizes on the generated path by the global planner, the required map preprocessing is performed by the operator. For instance, an STP file (the STP is a standard format for exchange of modelling data) is used to define the boundary edges of obstacles and the exterior frame of the environment. This preprocess is similar to changing a point cloud, as sensor data, into a continuous curve of the obstacle's boundary. Using the STP file as an input, various representations (QUAD, ITM, and VITM) with different sizes can be produced by the mentioned platform. A rectangle area of 10,000 × 5000 (mm) is utilized as the environment for all the performed simulations. Various types of obstacle distribution (single and multi-obstacles) are carried out regularly and randomly in this environment. A fixed scenario, navigation between points (300, 300) as the start and (9600, 4400) as the goal, is utilized for all the tests. The reason for selecting this path is that AGV needs to turn around the obstacles to pass the optimal length. After the generation of each map, the UTM platform makes it possible to perform global path planning and find the length, distance, and complexity in order to calculate the cost function for different representations. The running time of the utilized search algorithm is calculated by means of the number of visited cells in the graph search. This strategy is selected in order to be independent of coding details while comparing different cases.

5.2. Result—Environment and Representations

Consider an environment with a single medium obstacle as an example, which has been discretized by the three representations (QUAD, ITM, and VITM) with different sizes. Figure 2 represents the optimal path obtained by A* solely for one size of each representation. The scenario is to find a path between the start (300, 300) and goal (9600, 4400) points without considering inflation. Table 1 reports the length, minimum distance to obstacles, complexity, and cost of each case. From this table, ITM-700 has reached the lowest cost followed by VITM-100-700 while the GRID 200 × 100 has led to the highest cost. Therefore, ITM-700 and VITM-100-700 perform better than GRID 200 × 100 in terms of finding an optimal path in an environment with medium-size obstacles in parallel with the global frame.

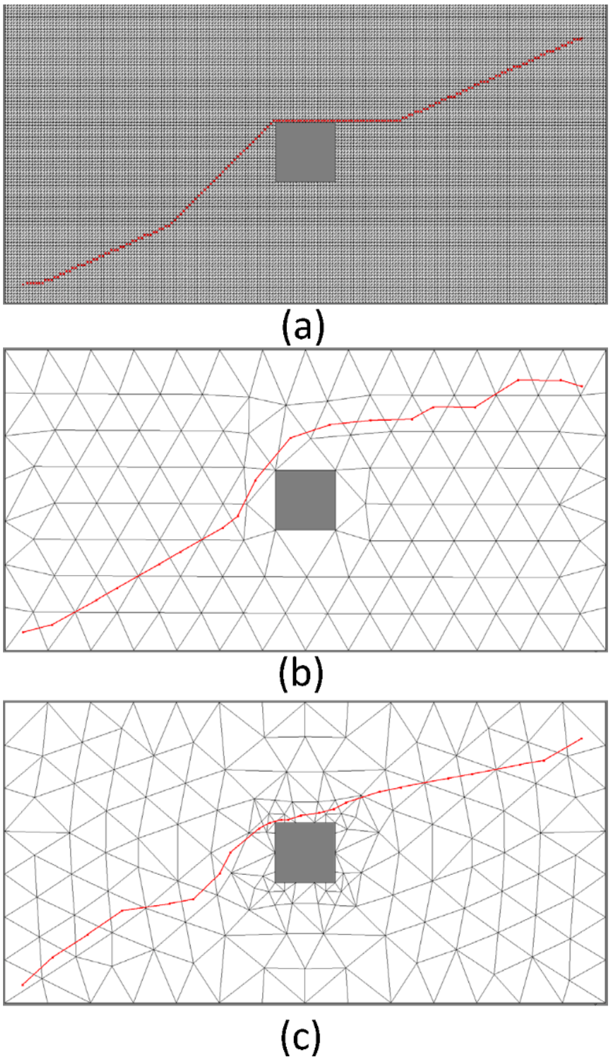

Figure 2. Different paths for a fixed scenario in an environment according to different representations. (**a**) A GRID 200 × 100. (**b**) An ITM-700. (**c**) A VITM-100-700.

Table 1. The length, minimum distance to obstacle, complexity, and cost function values for three sample representations for a sample map with a fixed scenario.

Representations			Calculated Parameters			f (Single Criteria)
Item	**Type**	**Size**	**Length (mm)**	**Min Distance (mm)**	**Complexity**	**Cost Function**
1	QUAD	GRID 200 × 100	10,992	25	8323	0.145
2	ITM	700	10,749	155	134	0.006
3	VITM	100_700	10,521	19	177	0.007

Similar scenarios have been performed for various environments and cell sizes, as shown in Figure A2. The considered environments are the different distributions variously mentioned in Section 5.1. Moreover, different cell sizes of QUAD, ITM, and VITM, as explained earlier, are considered. The complete result is presented in Table A1, and for simplicity, only the best representation of each environment (minimum f values among others) is shown in Table 2. According to this table, a coarse ITM (like ITM-700 or ITM-500) seems to be a good candidate for almost all environments except the hybrid environment where a VITM (VITM-100-700) shows better performance. It is worth reminding the reader that the detailed results, as in Table 1, are presented in the Appendix for all environments. The discussed results have been obtained by considering the pure decomposition effect without an inflation layer. In the next sub-section, the results concerning the effect of robot size or inflation layer are explored.

Table 2. Best representation for each environment according to f, cost function values without inflation.

MAP	Environment Description	Representation with the Lowest f Value	MAP	Environment Description	Representation with the Lowest f Value
Map_1	Small obstacle	ITM-700	Map_9	Regular different size	ITM-700
Map_2	Medium polygonal obstacle	ITM-700 ITM-500 VITM-100-700	Map_10	Irregular obstacles	ITM-700
Map_3	Medium circular obstacle	ITM-700 ITM-500 VITM-100-700	Map_11	Irregular two groups	ITM-700
Map_4	Space with dividers	ITM-700	Map_12	Irregular circular obstacles	ITM-700
Map_5	narrow passage	ITM-700	Map_13	Irregular different shape obstacles	ITM-700
Map_6	Curvilinear passage	ITM-700	Map_14	Hybrid with wide-spaced obstacles	ITM-700
Map_7	Regular obstacles	ITM-700	Map_15	Corridor and dividers	ITM-700
Map_8	Regular two groups	ITM-700	Map_16,17	Hybrid narrow-space environments	VITM-28-700 VITM-100-700

5.3. Result—Inflation Effect on the Representations

In practice, it is necessary to consider the robot size in path planning. This is normally done by imposing an inflation layer around the obstacles. In this regard, the tests performed in the previous sub-section are repeated herein, considering the inflation layer.

Firstly, the test regarding the environment with medium-size obstacles is repeated by considering a robot size diameter of 354 mm (turtle bot 2 [38]). Figure 3 shows the obtained optimal path for this new case. The red layer around the obstacle and exterior boundary is not accessible by the robot. Therefore, the movements between the neighbors are affected by this layer, and the results will be changed. The new values considering this limitation are given in Table 3 for the three different representations. From this table, VITM-100-700 reaches the best performance (f_{reps} : 0.009) compared to the other two cases. This shows that considering the inflation layer influences the choice of the best representation, due to the change in complexity.

Secondly, all the previously considered environments of different types and sizes are repeated, considering the inflation layer. The obtained results are shown in Table 4. In this new analysis, as opposed to the case in which inflation was ignored, it is seen that many ITM representations have not been able to find the path, and they have been replaced by the VITM and QUAD representations. Furthermore, it has been necessary for the VITM to use finer triangles in the narrow passages that have not been blocked completely. Decreasing the size of triangles (increasing their number) can increase the chance of finding more triangles whose centers are inside the free space, and movement between them and their children is possible. Some of the generated paths are presented in Figure 4. According to Table 4, coarse ITM shows the best performance for a single obstacle or hybrid environment, with enough wide space (for Map_1, Map_3, and Map_14). Regarding the corridor and regular multi-obstacles, the coarse grid is the best method. In irregular multi-obstacles and a curvilinear passage, a VITM indicates the best performance.

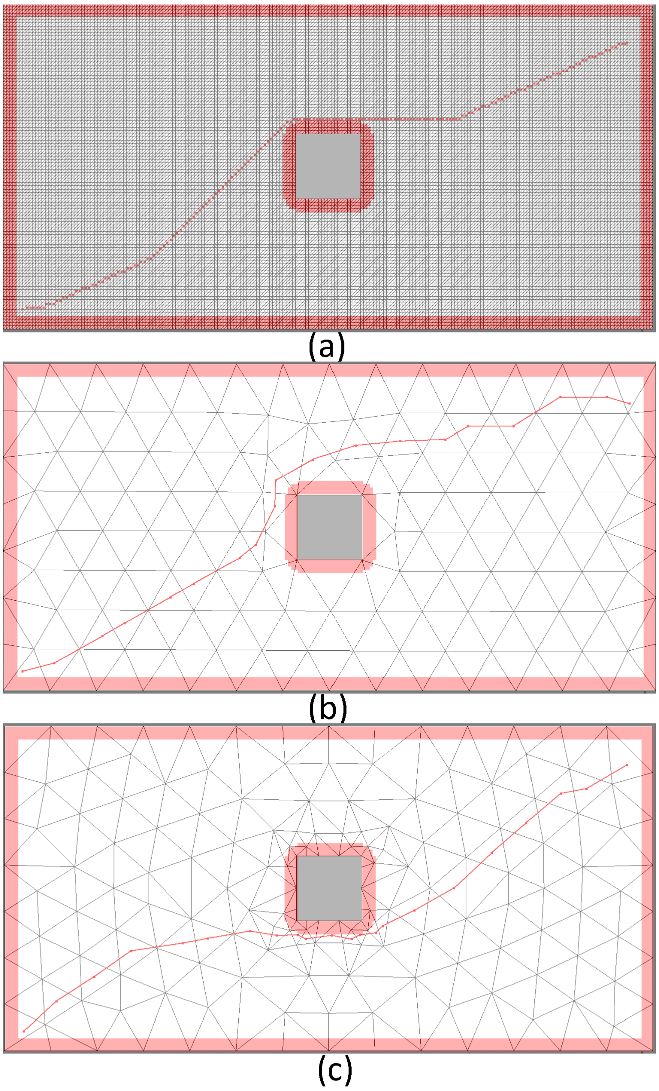

Figure 3. Different paths for the medium-size obstacle scenario considering the inflation layer. (**a**) A GRID 200 × 100. (**b**) An ITM-700. (**c**) A VITM-100-700.

Table 3. The length, minimum distance to obstacle, complexity, and cost function values for three sample representations considering the inflation layer.

Representations			**Calculated Parameters**			**F (Single Criteria)**
Item	**Type**	**Size**	**Length (mm)**	**Min Distance (mm)**	**Complexity**	**Cost Function**
1	QUAD	GRID 200 × 100	10,992	22	7555	0.132
2	ITM	700	10,878	98	532	0.013
3	VITM	100_700	10,469	48	327	0.009

Table 4. Best representation for each environment according to f, cost function values with consideration of the inflation layer.

MAP	Environment Description	Representation with the Lowest f Value	MAP	Environment Description	Representation with the Lowest f Value
Map_1	Small obstacle	ITM-500 VITM-100-700	Map_9	Regular different size	ITM-700 GRID 100 × 50
Map_2	Medium polygonal obstacle	VITM-100-700	Map_10	Irregular obstacles	VITM-30-700
Map_3	Medium circular obstacle	ITM-700	Map_11	Irregular two groups	ITM-500
Map_4	Space with dividers	ITM-700	Map_12	Irregular circular obstacles	ITM-700
Map_5	narrow passage	VITM-50-700	Map_13	Irregular different shape obstacles	VITM-100-700 GRID 100 × 50
Map_6	Curvilinear passage	VITM-50-700	Map_14	Hybrid with wide-spaced obstacles	ITM-700
Map_7	Regular obstacles	GRID 100 × 50	Map_15	Corridor and dividers	VITM-50-700 GRID 100 × 50
Map_8	Regular two groups	VITM-100-700 GRID 100 × 50	Map_16,17	Hybrid narrow-space environments	GRID 120 × 60 VITM-28-700 VITM-50-700 GRID 160 × 80

To find the effect of inflation on each type of representation, Table 5 shows the average cost function (f values) for all environments in two conditions, without inflation and with an inflation layer. Their ratio shows that ITM is more sensitive to the inflation layer, and misses its benefit. QUAD representation shows better behavior when using inflation.

Table 5. Average of cost function value on all environments, without and with inflation, and its ratio.

Reps	$f_{without\_infl}$	f_{with_infl}	$f_{with}/f_{without}$
QUAD	0.114	0.089	0.78
ITM	0.038	0.217	5.71
VITM	0.016	0.036	2.25

To compare the complexity of finding a path for each type of representation, we have calculated the average complexity (number of visited cells when graph searching) of each type of representation in Table 6. If any size could not give a path, a smaller size has been used to produce one. Overall, VITM shows the lowest complexity as compared to other types.

Table 6. Average of complexity value for all environments and each representation type.

Reps	QUAD	ITM	VITM
Average Complexity	3958	4244	1453

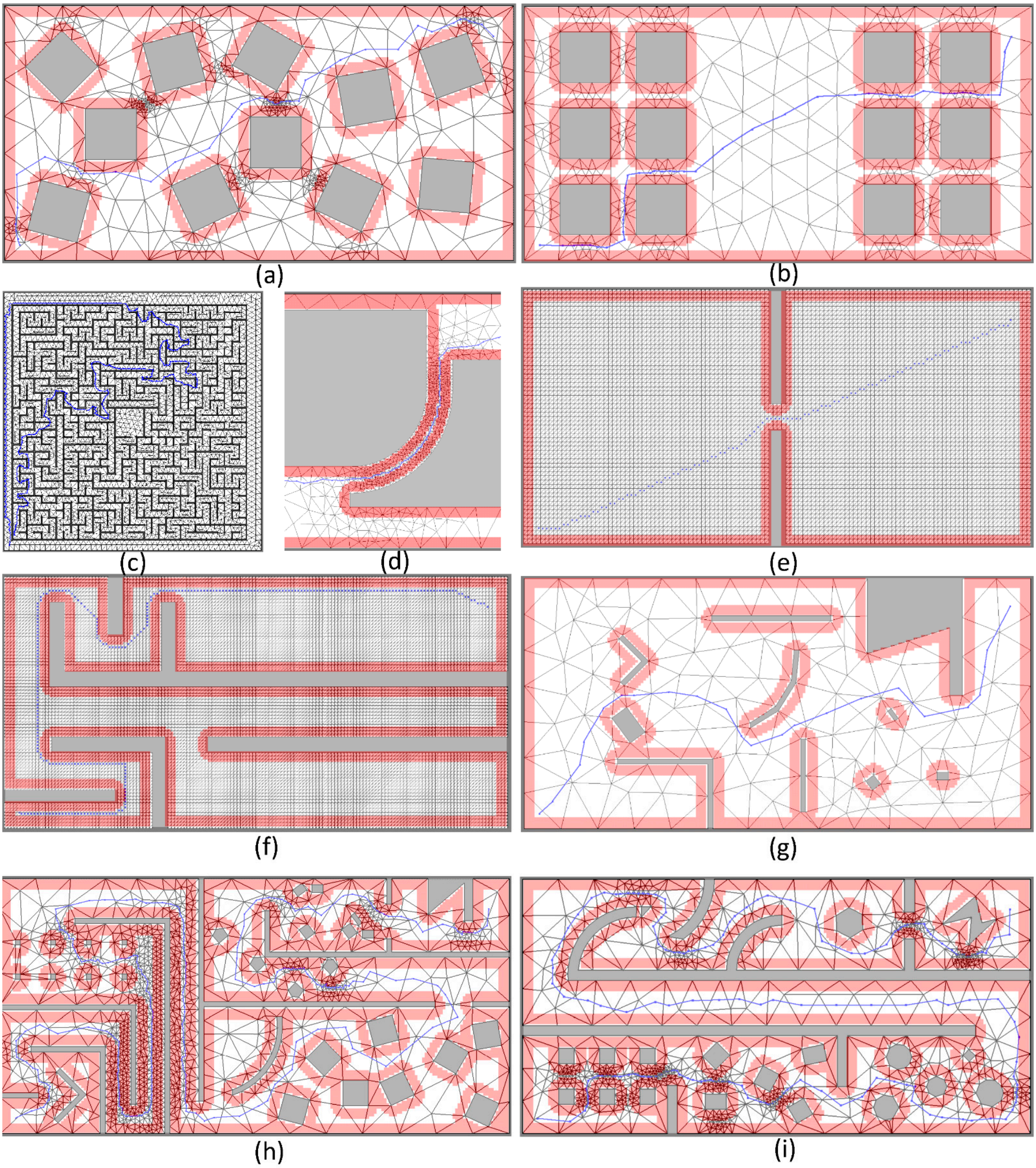

Figure 4. Different paths for some environments. (**a**) Blue path for Map_10-VITM-30-700. (**b**) Blue path for Map_8-100-700. (**c**) Blue path showing the capability of ITM in a maze map. (**d**) Blue path passing a narrow curvilinear passage by VITM method. (**e**) Blue path for a narrow passage by GRID 120 × 60 (**f**) Path in the corridor by GRID 160 × 80 (**g**) Representation ITM-700 with the generated path on a wide hybrid map (**h**,**i**) Two hybrid-narrow-passage environments with VITM representation.

5.4. Results—Discussion

Finding the best representation for an AGV inside a factory according to different environment encounters during its navigation is an important part of the use of autonomous agents. Different representations give different path. A good path has a short length with a maximum distance to obstacles that can be calculated quickly. Without knowing this path, a mobile robot can be an unfeasible solution for a factory, along with increasing energy consumption due to a less than optimal path, lowering efficiency by inappropriate planning time and increased cost due to the high risk of an inadequate distance to obstacles (high probability of collision). By considering all these important parameters of path planning, this work defined a single criterion and checked it for different types and sizes of representation for the various environments inside a factory.

The obtained results show that for various types of environments, such as single or multi obstacles and corridor-like, some level of coarse ITM is the predominant strategy for achieving a compromise solution for length, safety, and computation time; this verifies the work in [14], which shows that refinement is not necessary for some motion planning. However, what has not been considered is that, when considering the inflation layer around obstacles, VITM and GRID representations are better. It is clear that, by increasing the resolution or decreasing the cell size, the complexity increases, and in most cases, the nearest distance to the obstacles decreases, with some average improvement in length. According to this result, when the robot is of a fixed size in a static environment, and it is possible to apply first the robot size to the map, it is better to use the ITM method with coarse size like [16] in which inflation has been applied as offset during the first mapping. However, when there are the following situations, it is better to use VITM or GRID representations:

- In multi-robot sizes conditions, when is not possible to apply the offsetting method.
- Some changes in the environment make it necessary to apply online inflation to the map, and not by offsetting the obstacles in the primary map.
- There are narrow passages for a robot which makes ITM representation useless.

This benchmark also demonstrated that varying the size of the triangular mesh with a different strategy for refinement could be a competitive method when ITM cannot find the answer. Of course, this needs to be more refined when using the elements in an area with narrow passages. The other benefit of VITM is the more straightforward implementation relative to the quadtree method for neighbor finding, such as regular triangulation.

Robotic navigation in a real factory with multi AGVs with non-circular shapes is a more complicated task. This paper surveyed the effect of environment and map representation types on path planning considering a single circular AGV. Although a circle can cover any shape as a peripheral circle, it cannot obtain a path in some narrow passages. According to the robot's footprint, this limitation leads to consideration of a more complex definition of the inflation layer. In addition, to show the clear behavior of various cell shapes in response to A* search graph, we did not consider any path smoothing, which causes a non-optimal path from the point of energy consumption.

6. Conclusions and Future Work

In using an AGV as a feasible solution in a factory in response to the fourth industrial revolution, it is necessary to consider a flexible multi-objective path planner to give better answers according to the changes in the environment. Hence, this paper investigates the impact of two important factors, environment model representation and robot size, on the performance of an AGV. In this regard, a multi-objective cost function, considering length, complexity, and minimum distance to obstacles, is formulated for a perfect circular robot with a fixed path. Subsequently, three types of representation, namely QUAD, ITM, and VITM, with different cell sizes are employed to model the environment. Moreover, an inflation layer is imposed on the discretized view to check its influence on the performance. A* is utilized to find the optimal path for the different cell representations and sizes with and without the inflation layer. The obtained results show that a coarse ITM is a good candidate for a relatively unchanging environment with fixed robot size and wide space. However, when it is possible to repeatedly change the factory's plan and use different 'robot sizes, VITM and GRID representations give better performance by applying an inflation layer, especially in narrow spaces. In comparing triangle and square representations, in about 59% of industrial environments, constant or variable size triangle shapes are the only best for the discretization of that specific environment.

Future work can pursue the following paths, which will bring important practical application in the field of AGV environment modeling and navigation. Consideration of dynamic environments will enrich the optimization function with more complex criteria using a solution such as right-hand traffic, and applying some simultaneous constraints on the navigation of a narrow passage. Furthermore, the results of this paper can be used as an efficient map server that selects the best representation according to the environment. This can bring a smarter AGV platform to industry.

Author Contributions: Conceptualization, J.-C.C., V.F. and S.K.; Methodology, A.M., J.-C.C., V.F. and S.K.; Software, V.F. and A.M.; Validation, A.M., J.-C.C., V.F. and S.K.; Formal Analysis, A.M.; Investigation, A.M., J.-C.C., V.F. and S.K.; Resources, J.-C.C., V.F. and S.K.; Data Curation, A.M.; Writing—Original Draft Preparation, A.M.; Writing—Review & Editing, A.M., J.-C.C., V.F. and S.K.; Visualization, A.M.; Supervision, J.-C.C., V.F. and S.K.; Project Administration, S.K.; Funding Acquisition, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Natural Sciences and Engineering Research Council of Canada, grant Number CRSNG RDCPJ 518029-18.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the Natural Sciences and Engineering Research Council of Canada, Noovelia and the Foundation de l'Université du Québec à Trois-Rivières.

Conflicts of Interest: The authors have no conflict of interest for this work.

Appendix A

- Figure A1: All defined environments
- Figure A2: Sample of various representation types and sizes
- Table A1: Result without consideration of inflation for all environments and representations
- Table A2: Result with consideration of inflation for all environments and representations

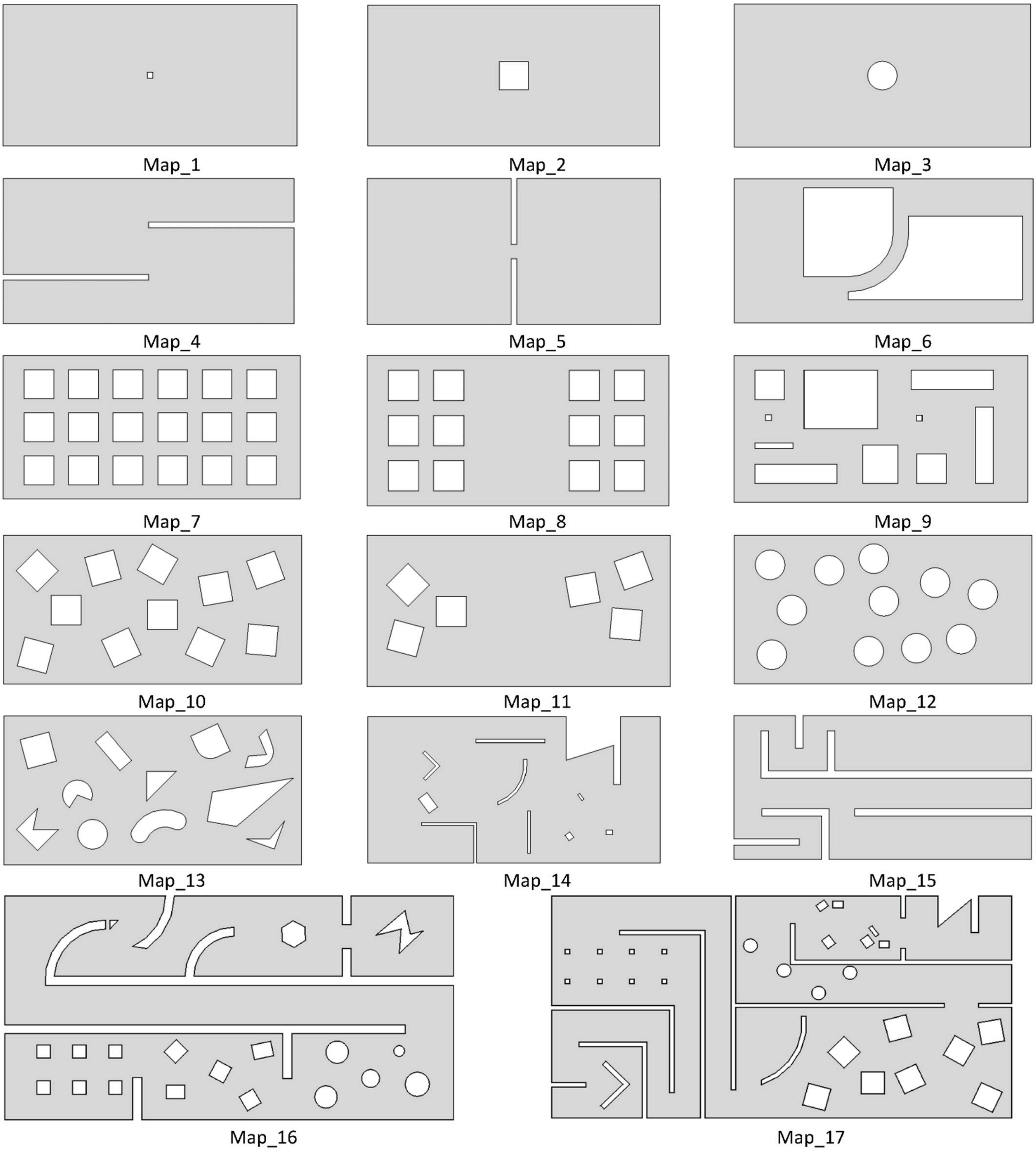

Figure A1. All defined environments.

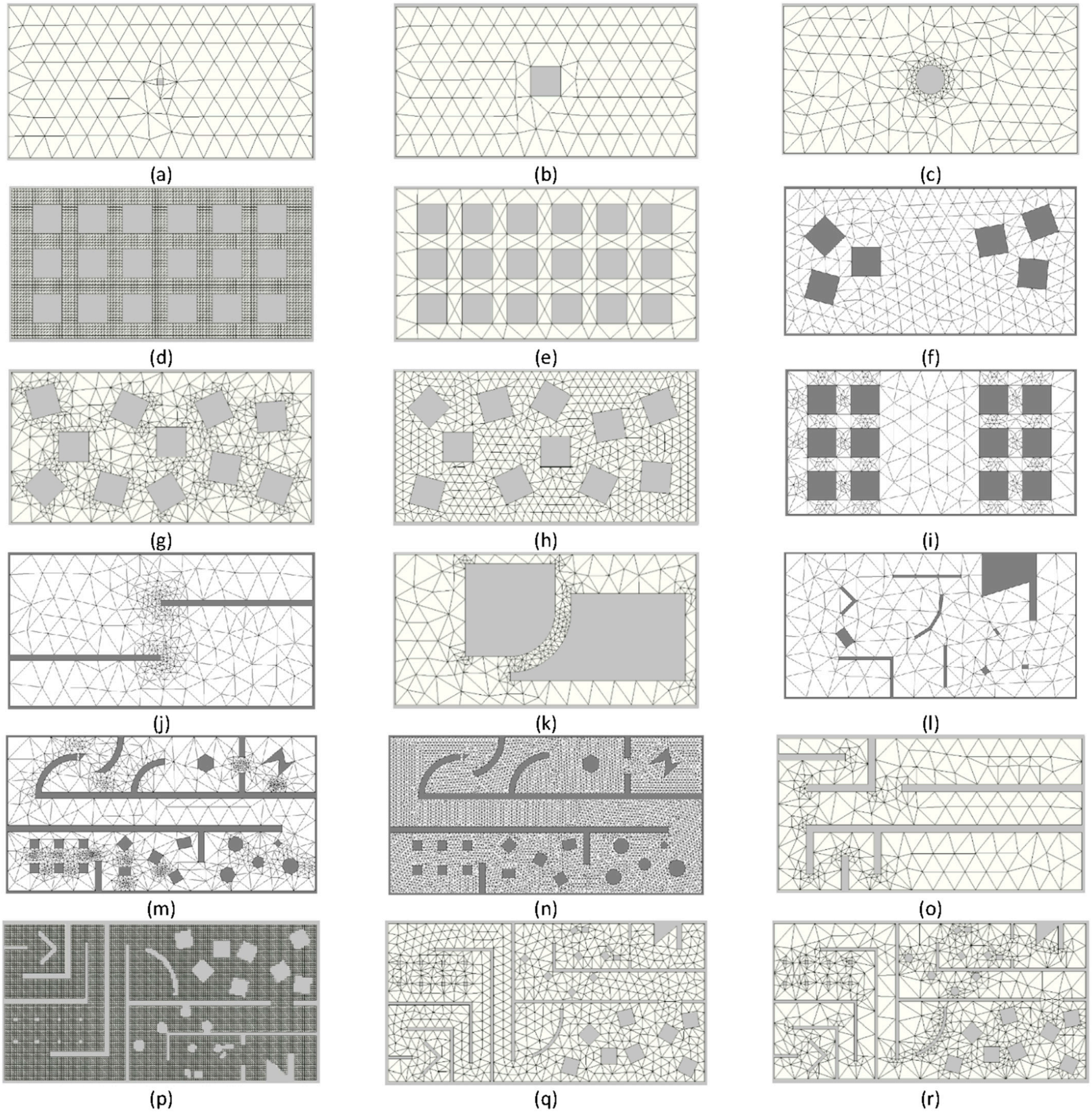

Figure A2. Various environments with different representation types and sizes of QUAD, ITM and VITM. (**a**) Map_1-ITM-700, (**b**) Map_2-ITM-100, (**c**) Map_3-VITM-100-700, (**d**) Map_7-GRID 200 × 100 (**e**) Map_7-ITM-300, (**f**) Map_11-ITM-500, (**g**) Map_10-VITM-100-700, (**h**) Map_10-ITM-300, (**i**) Map_8-VITM-100-700, (**j**) Map_4-VITM-100-700, (**k**) Map_6-VITM-100-700, (**l**) Map_14-ITM-700, (**m**) Map_16-VITM-30-700, (**n**) Map_16-ITM-100, (**o**) Map_15-VITM-100-700, (**p**) Map_17-GRID 200 × 100, (**q**) Map_178-ITM-300, (**r**) Map_17-VITM-100-700.

Table A1. The length, minimum distance to obstacle, complexity, and cost function values for all representations without considering the inflation layer in the same scenario.

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	Size	Inflation	Length mm	Min Distance mm	Complexity	f Cost Function
Map_1	QUAD	GRID 100 × 50	0	10,986	75	2269	0.042
	QUAD	GRID 120 × 60	0	11,022	125	3160	0.056
	QUAD	GRID 160 × 80	0	11,007	75	5463	0.095
	QUAD	GRID 200 × 100	0	10,992	25	8359	0.143
	ITM	700	0	10,451	133	114	0.005
	ITM	500	0	10,433	275	168	0.006
	ITM	300	0	10,381	46	358	0.010
	ITM	100	0	10,329	275	1644	0.031
	VITM	100_700	0	10,292	1	118	0.008
Map_2	QUAD	GRID 100 × 50	0	10,986	25	2251	0.042
	QUAD	GRID 120 × 60	0	11,022	25	3206	0.058
	QUAD	GRID 160 × 80	0	11,007	25	5452	0.097
	QUAD	GRID 200 × 100	0	10,992	25	8323	0.145
	ITM	700	0	10,749	155	134	0.006
	ITM	500	0	10,375	25	153	0.006
	ITM	300	0	10,484	46	369	0.010
	ITM	100	0	10,570	125	3030	0.055
	VITM	100_700	0	10,521	19	177	0.007
Map_3	QUAD	GRID 100 × 50	0	10,986	25	2255	0.042
	QUAD	GRID 120 × 60	0	11,022	25	3049	0.056
	QUAD	GRID 160 × 80	0	11,007	25	5269	0.093
	QUAD	GRID 200 × 100	0	10,992	25	8117	0.141
	ITM	700	0	10,521	17	106	0.006
	ITM	500	0	10,559	75	190	0.007
	ITM	300	0	10,456	3	444	0.012
	ITM	100	0	10,399	19	2191	0.041
	VITM	100_700	0	10,394	40	173	0.007

Table A1. *Cont.*

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	Size	Inflation	Length mm	Min Distance mm	Complexity	f Cost Function
Map_4	QUAD	GRID 100 × 50	0	12,416	25	2695	0.051
	QUAD	GRID 120 × 60	0	12,409	25	3707	0.069
	QUAD	GRID 160 × 80	0	12,340	25	6468	0.117
	QUAD	GRID 200 × 100	0	12,293	25	10,078	0.179
	ITM	700	0	12,565	116	164	0.007
	ITM	500	0	12,438	75	295	0.010
	ITM	300	0	12,222	75	747	0.017
	ITM	100	0	12,031	13	5613	0.102
	VITM	100_700	0	11,953	15	412	0.012
Map_5	QUAD	GRID 100 × 50	0	10,986	75	2102	0.039
	QUAD	GRID 120 × 60	0	11,022	25	2917	0.053
	QUAD	GRID 160 × 80	0	11,007	25	5064	0.090
	QUAD	GRID 200 × 100	0	10,992	25	7812	0.136
	ITM	700	0	10,294	87	92	0.005
	ITM	500	0	10,285	125	153	0.006
	ITM	300	0	10,339	133	308	0.009
	ITM	100	0	10,379	116	2316	0.043
	VITM	100_700	0	10,404	125	220	0.007
Map_6	QUAD	GRID 100 × 50	0	11,630	25	1579	0.053
	QUAD	GRID 120 × 60	0	11,657	25	2134	0.070
	QUAD	GRID 160 × 80	0	11,592	25	3726	0.116
	QUAD	GRID 200 × 100	0	11,519	25	5740	0.175
	ITM	700	0	11,109	46	94	0.009
	ITM	500	0	11,403	75	188	0.012
	ITM	300	0	11,048	46	360	0.017
	ITM	100	0	11,065	14	2859	0.091
	VITM	100_700	0	11,052	25	275	0.015

Table A1. *Cont.*

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	Size	Inflation	Length mm	Min Distance mm	Complexity	f Cost Function
Map_7	QUAD	GRID 100 × 50	0	12,386	25	2289	0.066
	QUAD	GRID 120 × 60	0	12,380	25	2995	0.085
	QUAD	GRID 160 × 80	0	12,327	25	5502	0.150
	QUAD	GRID 200 × 100	0	12,173	25	8388	0.225
	ITM	700	0	12,760	46	190	0.012
	ITM	500	0	12,357	20	226	0.013
	ITM	300	0	12,233	11	552	0.021
	ITM	100	0	11,798	0.4	4428	0.130
	VITM	100_700	0	12,031	45	513	0.020
Map_8	QUAD	GRID 100 × 50	0	11,566	25	2224	0.054
	QUAD	GRID 120 × 60	0	11,640	25	3418	0.080
	QUAD	GRID 160 × 80	0	11,521	25	5532	0.126
	QUAD	GRID 200 × 100	0	11,511	25	8836	0.199
	ITM	700	0	11,938	20	167	0.009
	ITM	500	0	11,465	27	242	0.011
	ITM	300	0	11,444	39	611	0.019
	ITM	100	0	11,181	0	4737	0.117
	VITM	100_700	0	11,584	5	435	0.015
Map_9	QUAD	GRID 100 × 50	0	11,408	25	1705	0.046
	QUAD	GRID 120 × 60	0	11,461	25	2483	0.064
	QUAD	GRID 160 × 80	0	11,373	25	4069	0.102
	QUAD	GRID 200 × 100	0	11,343	25	6520	0.160
	ITM	700	0	11,207	31	120	0.008
	ITM	500	0	11,108	2	190	0.011
	ITM	300	0	10,921	13	379	0.015
	ITM	100	0	10,878	10	2793	0.072
	VITM	100_700	0	10,925	17	315	0.013

Table A1. *Cont.*

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	Size	Inflation	Length mm	Min Distance mm	Complexity	f Cost Function
Map_10	QUAD	GRID 100 × 50	0	11,239	25	1458	0.037
	QUAD	GRID 120 × 60	0	11,189	25	2252	0.055
	QUAD	GRID 160 × 80	0	11,222	25	3771	0.089
	QUAD	GRID 200 × 100	0	11,201	25	5759	0.132
	ITM	700	0	11,105	14	119	0.008
	ITM	500	0	11,149	45	209	0.010
	ITM	300	0	11,292	2	583	0.020
	ITM	100	0	10,814	3	2507	0.061
	VITM	100_700	0	10,854	32	440	0.015
Map_11	QUAD	GRID 100 × 50	0	11,156	25	1650	0.036
	QUAD	GRID 120 × 60	0	11,120	25	2498	0.053
	QUAD	GRID 160 × 80	0	11,119	25	4214	0.086
	QUAD	GRID 200 × 100	0	11,119	25	6395	0.128
	ITM	700	0	10,500	13	95	0.006
	ITM	500	0	10,615	45	140	0.007
	ITM	300	0	10,519	32	356	0.011
	ITM	100	0	10,550	18	2998	0.062
	VITM	100_700	0	10,596	21	317	0.010
Map_12	QUAD	GRID 100 × 50	0	11,069	25	1651	0.039
	QUAD	GRID 120 × 60	0	11,091	25	2321	0.053
	QUAD	GRID 160 × 80	0	11,058	25	4066	0.089
	QUAD	GRID 200 × 100	0	11,033	25	6316	0.135
	ITM	700	0	10,531	4	94	0.007
	ITM	500	0	10,454	24	153	0.008
	ITM	300	0	10,369	29	230	0.009
	ITM	100	0	10,522	24	2026	0.046
	VITM	100_700	0	10,398	42	299	0.011

Table A1. *Cont.*

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	Size	Inflation	Length mm	Min Distance mm	Complexity	f Cost Function
Map_13	QUAD	GRID 100 × 50	0	11,239	25	1479	0.036
	QUAD	GRID 120 × 60	0	11,189	25	2366	0.055
	QUAD	GRID 160 × 80	0	-	-	-	-
	QUAD	GRID 200 × 100	0	11,201	25	5999	0.132
	ITM	700	0	11,048	12	126	0.008
	ITM	500	0	11,026	44	160	0.008
	ITM	300	0	11,011	23	494	0.015
	ITM	100	0	10,891	14	3802	0.086
	VITM	100_700	0	11,063	21	254	0.010
Map_14	QUAD	GRID 100 × 50	0	11,903	25	3025	0.060
	QUAD	GRID 120 × 60	0	11,855	25	4270	0.083
	QUAD	GRID 160 × 80	0	11,918	25	7623	0.145
	QUAD	GRID 200 × 100	0	11,892	25	11,932	0.225
	ITM	700	0	11,568	42	214	0.008
	ITM	500	0	11,469	46	327	0.010
	ITM	300	0	11,573	46	771	0.019
	ITM	100	0	11,346	9	5936	0.114
	VITM	100_700	0	11,590	5	298	0.010
Map_15	QUAD	GRID 100 × 50	0	17,192	25	3118	0.068
	QUAD	GRID 120 × 60	0	17,246	25	4422	0.093
	QUAD	GRID 160 × 80	0	17,031	25	7472	0.153
	QUAD	GRID 200 × 100	0	16,769	25	11,676	0.235
	ITM	700	0	17,764	31	209	0.011
	ITM	500	0	18,192	46	353	0.014
	ITM	300	0	17,480	29	919	0.025
	ITM	100	0	16,593	8	7053	0.145
	VITM	100_700	0	16,786	1	403	0.020

Table A1. *Cont.*

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	**Size**	**Inflation**	**Length mm**	**Min Distance mm**	**Complexity**	**f Cost Function**
Map_16	QUAD	GRID 200 × 100	0	29,830	25	13,978	0.303
	ITM	100	0	30,072	0	8215	0.219
	VITM	28_700	0	30,183	5	1538	0.045
Map_17	QUAD	GRID 120 × 60	0	32,876	25	5157	0.119
	QUAD	GRID 160 × 80	0	32,308	25	9484	0.206
	QUAD	GRID 200 × 100	0	32,075	25	15,000	0.318
	ITM	300	0	32,204	5	1207	0.039
	ITM	100	0	31,732	3	2455	0.064
	ITM	50	0	30,498	0	9232	0.216
	VITM	100_700	0	31,494	1	1095	0.037

Table A2. The length, minimum distance to obstacle, complexity, and cost function values for all representations considering the inflation layer in the same scenario (Robot diameter = 354 mm).

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	**Size**	**Inflation**	**Length mm**	**Min Distance mm**	**Complexity**	**f Cost Function**
Map_1	QUAD	GRID 100 × 50	1	10,986	48	2160	0.040
	QUAD	GRID 120 × 60	1	11,022	4	3017	0.055
	QUAD	GRID 160 × 80	1	11,007	4	5222	0.092
	QUAD	GRID 200 × 100	1	10,992	22	7978	0.137
	ITM	700	1	10,560	98	367	0.010
	ITM	500	1	10,433	98	331	0.009
	ITM	300	1	10,472	98	581	0.013
	ITM	100	1	10,329	98	1845	0.034
	VITM	100_700	1	10,362	98	345	0.009

Table A2. *Cont.*

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	Size	Inflation	Length mm	Min Distance mm	Complexity	f Cost Function
Map_2	QUAD	GRID 100 × 50	1	10,986	48	2101	0.040
	QUAD	GRID 120 × 60	1	11,022	48	2995	0.055
	QUAD	GRID 160 × 80	1	11,007	22	4987	0.089
	QUAD	GRID 200 × 100	1	10,992	22	7555	0.132
	ITM	700	1	10,878	98	532	0.013
	ITM	500	1	10,492	48	457	0.011
	ITM	300	1	10,597	48	749	0.016
	ITM	100	1	10,635	10	3714	0.067
	VITM	100_700	1	10,469	48	327	0.009
Map_3	QUAD	GRID 100 × 50	1	10,986	48	2029	0.038
	QUAD	GRID 120 × 60	1	11,022	10	2842	0.052
	QUAD	GRID 160 × 80	1	11,007	10	4902	0.087
	QUAD	GRID 200 × 100	1	10,992	22	7458	0.130
	ITM	700	1	10,671	98	324	0.009
	ITM	500	1	10,566	4	464	0.012
	ITM	300	1	10,519	98	622	0.014
	ITM	100	1	10,508	4	2869	0.053
	VITM	100_700	1	10,641	67	673	0.015
Map_4	QUAD	GRID 100 × 50	1	13,099	48	2387	0.046
	QUAD	GRID 120 × 60	1	13,076	4	3396	0.064
	QUAD	GRID 160 × 80	1	13,126	4	6016	0.110
	QUAD	GRID 200 × 100	1	13,151	4	9240	0.166
	ITM	700	1	12,777	81	928	0.021
	ITM	500	1	12,633	10	1365	0.028
	ITM	300	1	12,988	81	1742	0.035
	ITM	100	1	12,751	4	7671	0.138
	VITM	100_700	1	12,686	10	1334	0.028

Table A2. *Cont.*

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	Size	Inflation	Length mm	Min Distance mm	Complexity	f Cost Function
Map_5	QUAD	GRID 100 × 50	1	-	-	-	-
	QUAD	GRID 120 × 60	1	11,022	48	2665	0.049
	QUAD	GRID 160 × 80	1	11,007	4	4611	0.083
	QUAD	GRID 200 × 100	1	10,992	22	7057	0.124
	ITM	700	1	-	-	-	-
	ITM	500	1	-	-	-	-
	ITM	300	1	-	-	-	-
	ITM	100	1	10,402	4	2554	0.048
	VITM	50_700	1	10,409	4	681	0.016
Map_6	QUAD	GRID 100 × 50	1	-	-	-	-
	QUAD	GRID 120 × 60	1	-	-	-	-
	QUAD	GRID 160 × 80	1	-	-	-	-
	QUAD	GRID 200 × 100	1	11,900	4	4607	0.143
	ITM	500	1	-	-	-	-
	ITM	300	1	-	-	-	-
	ITM	100	1	-	-	-	-
	ITM	50	1	11,668	4	13,978	0.418
	VITM	50_700	1	11,545	4	1111	0.040
Map_7	QUAD	GRID 100 × 50	1	12,756	48	730	0.026
	QUAD	GRID 120 × 60	1	12,800	4	1188	0.038
	QUAD	GRID 160 × 80	1	12,779	4	2049	0.061
	QUAD	GRID 200 × 100	1	12,797	22	2828	0.080
	ITM	700	1	-	-	-	-
	ITM	500	1	-	-	-	-
	ITM	300	1	-	-	-	-
	ITM	100	1	13,176	4	4933	0.136
	VITM	100_700	1	16,061	4	1706	0.054

Table A2. *Cont.*

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	Size	Inflation	Length mm	Min Distance mm	Complexity	f Cost Function
Map_8	QUAD	GRID 100 × 50	1	11,703	48	1523	0.039
	QUAD	GRID 120 × 60	1	11,668	4	2193	0.054
	QUAD	GRID 160 × 80	1	11,690	4	3892	0.091
	QUAD	GRID 200 × 100	1	11,668	22	5802	0.132
	ITM	700	1	-	-	-	-
	ITM	500	1	-	-	-	-
	ITM	300	1	-	-	-	-
	ITM	100	1	11,854	4	5765	0.132
	VITM	100_700	1	12,003	4	1138	0.031
Map_9	QUAD	GRID 100 × 50	1	11,584	48	1095	0.032
	QUAD	GRID 120 × 60	1	11,604	4	1580	0.044
	QUAD	GRID 160 × 80	1	11,581	4	2656	0.069
	QUAD	GRID 200 × 100	1	11,584	22	4125	0.103
	ITM	700	1	12,842	48	969	0.029
	ITM	500	1	12,047	4	1250	0.036
	ITM	300	1	11,964	22	1613	0.044
	ITM	100	1	11,584	4	4978	0.124
	VITM	100_700	1	11,972	4	1432	0.041
Map_10	QUAD	GRID 100 × 50	1	-	-	-	-
	QUAD	GRID 120 × 60	1	13,171	4	2037	0.052
	QUAD	GRID 160 × 80	1	12,162	4	2461	0.061
	QUAD	GRID 200 × 100	1	12,211	4	3731	0.089
	ITM	500	1	-	-	-	-
	ITM	300	1	-	-	-	-
	ITM	100	1	-	-	-	-
	ITM	50	1	12,065	4	13,897	0.313
	VITM	30_700	1	12,547	4	1485	0.039

Table A2. *Cont.*

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	Size	Inflation	Length mm	Min Distance mm	Complexity	f Cost Function
Map_11	QUAD	GRID 100 × 50	1	11,404	10	1541	0.034
	QUAD	GRID 120 × 60	1	11,258	4	2295	0.049
	QUAD	GRID 160 × 80	1	11,274	4	3894	0.080
	QUAD	GRID 200 × 100	1	11,284	4	5876	0.135
	ITM	700	1	-	-	-	-
	ITM	500	1	11,074	4	851	0.021
	ITM	300	1	-	-	-	-
	ITM	100	1	11,214	4	6390	0.128
	VITM	100_700	1	12,754	4	2012	0.044
Map_12	QUAD	GRID 100 × 50	1	11,915	22	1333	0.033
	QUAD	GRID 120 × 60	1	11,465	4	1791	0.043
	QUAD	GRID 160 × 80	1	11,494	4	3110	0.070
	QUAD	GRID 200 × 100	1	11,411	22	4630	0.101
	ITM	700	1	11,602	4	853	0.023
	ITM	500	1	11,796	4	1123	0.029
	ITM	300	1	10,895	10	1119	0.028
	ITM	100	1	11,533	4	4676	0.102
	VITM	28_700	1	11,449	4	1263	0.032
Map_13	QUAD	GRID 100 × 50	1	12,425	10	1878	0.045
	QUAD	GRID 120 × 60	1	12,205	4	2780	0.065
	QUAD	GRID 160 × 80	1	12,256	4	4808	0.108
	QUAD	GRID 200 × 100	1	12,316	4	7430	0.164
	ITM	700	1	-	-	-	-
	ITM	500	1	-	-	-	-
	ITM	300	1	-	-	-	-
	ITM	100	1	12,133	4	9511	0.208
	VITM	100_700	1	12,306	4	1797	0.044

Table A2. *Cont.*

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	Size	Inflation	Length mm	Min Distance mm	Complexity	f Cost Function
Map_14	QUAD	GRID 100 × 50	1	14,540	22	2421	0.050
	QUAD	GRID 120 × 60	1	13,245	10	3860	0.076
	QUAD	GRID 160 × 80	1	14,370	4	5985	0.117
	QUAD	GRID 200 × 100	1	13,247	22	10,293	0.195
	ITM	700	1	12,918	4	1345	0.030
	ITM	500	1	14,584	4	1573	0.035
	ITM	300	1	14,258	48	2399	0.050
	ITM	100	1	12,786	4	10,729	0.203
	VITM	50_700	1	13,208	10	1717	0.037
Map_15	QUAD	GRID 100 × 50	1	18,640	10	2073	0.048
	QUAD	GRID 120 × 60	1	18,717	4	3107	0.069
	QUAD	GRID 160 × 80	1	18,525	4	5312	0.112
	QUAD	GRID 200 × 100	1	18,651	4	8180	0.168
	ITM	700	1	-	-	-	-
	ITM	500	1	-	-	-	-
	ITM	300	1	-	-	-	-
	ITM	100	1	18,584	4	10,910	0.221
	VITM	50_700	1	18,918	4	1701	0.042
Map_16	QUAD	GRID 120 × 60	1	35,929	4	2661	0.071
	QUAD	GRID 200 × 100	1	35,954	4	6878	0.159
	ITM	100	1	36,679	4	11,402	0.253
	VITM	28_700	1	36,430	4	2801	0.074

Table A2. *Cont.*

Map Name	Representations		S (300, 300) G (9600, 4400)	Parameters			Single Criteria
	Type	Size	Inflation	Length mm	Min Distance mm	Complexity	f Cost Function
Map_17	QUAD	GRID 120 × 60	1	-	-	-	-
	QUAD	GRID 160 × 80	1	37,557	4	4020	0.098
	QUAD	GRID 200 × 100	1	38,084	4	6242	0.143
	ITM	300	1	-	-	-	-
	ITM	100	1	-	-	-	-
	ITM	50	1	37,727	4	28,135	0.588
	VITM	50_700	1	38,756	4	3180	0.081

References

1. De Ryck, M.; Versteyhe, M.; Debrouwere, F. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *J. Manuf. Syst.* **2020**, *54*, 152–173. [CrossRef]
2. Oyekanlu, E.A.; Smith, A.C.; Thomas, W.P.; Mulroy, G.; Hitesh, D.; Ramsey, M.; Kuhn, D.J.; Mcghinnis, J.D.; Buonavita, S.C.; Looper, N.A. A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5G-based smart manufacturing applications. *IEEE Access* **2020**, *8*, 202312–202353. [CrossRef]
3. Fragapane, G.; Ivanov, D.; Peron, M.; Sgarbossa, F.; Strandhagen, J.O. Increasing flexibility and productivity in Industry 4.0 production networks with autonomous mobile robots and smart intralogistics. *Ann. Oper. Res.* **2020**, 1–19. [CrossRef]
4. Martínez-Barberá, H.; Herrero-Pérez, D. Autonomous navigation of an automated guided vehicle in industrial environments. *Robot. Comput.-Integr. Manuf.* **2010**, *26*, 296–311. [CrossRef]
5. Eykhoff, P. *System Identification*; Wiley: New York, NY, USA, 1974; Volume 14.
6. Thrun, S. Robotic mapping: A survey. *Explor. Artif. Intell. New Millenn.* **2002**, *1*, 1.
7. Moravec, H.; Elfes, A. High resolution maps from wide angle sonar. In Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 25–28 March 1985; pp. 116–121.
8. Botsch, M.; Kobbelt, L.; Pauly, M.; Alliez, P.; Lévy, B. *Polygon Mesh Processing*; CRC Press: Boca Raton, FL, USA, 2010.
9. Marton, Z.C.; Rusu, R.B.; Beetz, M. On fast surface reconstruction methods for large and noisy point clouds. In Proceedings of the 2009 IEEE International Conference on Robotics And Automation, Kobe, Japan, 12–17 May 2009; pp. 3218–3223.
10. Latombe, J.-C. *Robot Motion Planning*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 124.
11. Fujimura, K. Path planning with multiple objectives. *IEEE Robot. Autom. Mag.* **1996**, *3*, 33–38. [CrossRef]
12. Rañó, I.; Minguez, J. Steps towards the automatic evaluation of robot obstacle avoidance algorithms. In *Proc. of Workshop of Benchmarking in Robotics, in the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*; IEEE: Beijing, China, 2006; Volume 88, pp. 90–91.
13. Tsardoulias, E.; Iliakopoulou, A.; Kargakos, A.; Petrou, L. A review of global path planning methods for occupancy grid maps regardless of obstacle density. *J. Intell. Robot. Syst.* **2016**, *84*, 829–858. [CrossRef]
14. Plaku, E.; Kavraki, L.E.; Vardi, M.Y. Impact of workspace decompositions on discrete search leading continuous exploration (DSLX) motion planning. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 3751–3756.
15. Aggarwal, R.; Kumar, M. Chance-Constrained Approach to Optimal Path Planning for Urban UAS. In *AIAA Scitech 2020 Forum*; American Institute of Aeronautics and Astronautics, Inc.: Orlando, FL, USA, 2020; p. 0857.
16. Liu, Y.; Jiang, Y. Robotic Path Planning Based on a Triangular Mesh Map. *Int. J. Control. Autom. Syst.* **2020**, *18*, 2658–2666. [CrossRef]
17. van Toll, W.; Triesscheijn, R.; Kallmann, M.; Oliva, R.; Pelechano, N.; Pettré, J.; Geraerts, R. Comparing navigation meshes: Theoretical analysis and practical metrics. *Comput. Graph.* **2020**, *91*, 52–82. [CrossRef]
18. Kallmann, M. Shortest Paths with Arbitrary Clearance from Navigation Meshes. In *Symposium on Computer Animation*; Eurographics Association: Madrid, Spain, 2010; pp. 159–168.
19. Oliva, R.; Pelechano, N. A generalized exact arbitrary clearance technique for navigation meshes. In *Proceedings of Motion on Games*; Association for Computing Machinery: New York, NY, USA, 2013; pp. 103–110.

20. Guimarães, R.L.; de Oliveira, A.S.; Fabro, J.A.; Becker, T.; Brenner, V.A. ROS navigation: Concepts and tutorial. In *Robot Operating System (ROS)*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 121–160.
21. Fernandes, E.; Costa, P.; Lima, J.; Veiga, G. Towards an orientation enhanced astar algorithm for robotic navigation. In Proceedings of the 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, 17–19 March 2015; pp. 3320–3325.
22. Cuillière, J.-C.; Francois, V. Integration of CAD, FEA and topology optimization through a unified topological model. *Comput.-Aided Des. Appl.* **2014**, *11*, 493–508. [CrossRef]
23. Kumar, G.S.; Kalra, P.K.; Dhande, S.G. Curve and surface reconstruction from points: An approach based on self-organizing maps. *Appl. Soft Comput.* **2004**, *5*, 55–66. [CrossRef]
24. Stanimirovic, I.P.; Zlatanovic, M.L.; Petkovic, M.D. On the linear weighted sum method for multi-objective optimization. *Facta Acta Univ.* **2011**, *26*, 49–63.
25. Ravankar, A.; Ravankar, A.A.; Kobayashi, Y.; Hoshino, Y.; Peng, C.-C. Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges. *Sensors* **2018**, *18*, 3170. [CrossRef]
26. Lamini, C.; Benhlima, S.; Elbekri, A. Genetic algorithm based approach for autonomous mobile robot path planning. *Procedia Comput. Sci.* **2018**, *127*, 180–189. [CrossRef]
27. Rekleitis, I.; Bedwani, J.-L.; Dupuis, E. Autonomous planetary exploration using LIDAR data. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 3025–3030.
28. Kallmann, M. Path planning in triangulations. In Proceedings of the IJCAI Workshop on Reasoning, Representation, and Learning in Computer Games, Edinburgh, Scotland, 31 July 2005; pp. 49–54.
29. Mark, d.B.; Otfried, C.; Marc, v.K.; Mark, O. *Computational Geometry Algorithms and Applications*; Spinger: Berlin/Heidelberg, Germany, 2008.
30. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
31. Liu, X.; Gong, D. A comparative study of A-star algorithms for search and rescue in perfect maze. In Proceedings of the 2011 International Conference on Electric Information and Control Engineering, Wuhan, China, 15–17 April 2011; pp. 24–27.
32. Fabbri, R.; Costa, L.D.F.; Torelli, J.C.; Bruno, O.M. 2D Euclidean distance transform algorithms: A comparative survey. *ACM Comput. Surv. (CSUR)* **2008**, *40*, 1–44. [CrossRef]
33. Lau, B.; Sprunk, C.; Burgard, W. Efficient grid-based spatial representations for robot navigation in dynamic environments. *Robot. Auton. Syst.* **2013**, *61*, 1116–1130. [CrossRef]
34. Cuillière, J.-C. An adaptive method for the automatic triangulation of 3D parametric surfaces. *Comput.-Aided Des.* **1998**, *30*, 139–149. [CrossRef]
35. François, V.; Cuilliere, J.-C. 3D automatic remeshing applied to model modification. *Comput.-Aided Des.* **2000**, *32*, 433–444. [CrossRef]
36. Cuillière, J.-C.; François, V.; Lacroix, R. A new approach to automatic and a priori mesh adaptation around circular holes for finite element analysis. *Comput.-Aided Des.* **2016**, *77*, 18–45. [CrossRef]
37. François, V.; Cuillière, J.-C. Automatic mesh pre-optimization based on the geometric discretization error. *Adv. Eng. Softw.* **2000**, *31*, 763–774. [CrossRef]
38. Clearpath Robotics Inc. Turtlebot-2-Open-Source-Robot. Available online: https://clearpathrobotics.com/turtlebot-2-open-source-robot/ (accessed on 9 April 2022).