UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

CARTE HYBRIDE À BASE DE TUILES POUR UNE PLANIFICATION DE TRAJECTOIRE EFFICACE CHEZ LES ROBOTS AUTONOMES DANS DES ENVIRONNEMENTS VASTES ET ENCOMBRÉS

THÈSE PRÉSENTÉE
COMME EXIGENCE PARTIELLE DU
DOCTORAT EN INGÉNIERIE (CONCENTRATION GÉNIE MÉCANIQUE)

PAR
AHMADREZA MEYSAMI

June 2026

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

TILED HYBRID REPRESENTATION MAP FOR EFFICIENT PATH PLANNING IN AUTONOMOUS ROBOTS OPERATING IN LARGE, CLUTTERED ENVIRONMENTS

A THESIS PRESENTED
IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE
DOCTOR OF PHILOSOPHY IN ENGINEERING (MECHANICAL ENGINEERING CONCENTRATION)

BY
AHMADREZA MEYSAMI

Juin 2026

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

EN DOCTORAT EN INGÉNIERIE (CONCENTRATION GÉNIE MÉCANIQUE) (Ph.D.)

Direction de recherche :

Sousso Kelouwani	directeur de recherche
Jean-Christophe Cuillière	codirecteur de recherche
Vincent François	codirecteur de recherche

Jury d'évaluation

Pierre-Yves Lajoie	évaluateur externe (Département de génie informatique et génie logiciel, École Polytechnique de Montréal)
Nguyen Tan Sy	évaluateur externe (Département de mathématiques, d'informatique et de génie, UQAR)
Luc Laperrière	président du jury (DGM, UQTR)
Vincent François	codirecteur de recherche (DGM, UQTR)
Jean-Christophe Cuillière	codirecteur de recherche (DGM, UQTR)
Sousso Kelouwani	directeur de recherche (DGM, UQTR)

Thèse soutenue le 6 05 2026

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

DOCTOR OF PHILOSOPHY IN ENGINEERING (MECHANICAL ENGINEERING CONCENTRATION) (Ph.D.)

Research Supervision:

Sousso Kelouwani	research director
Jean-Christophe Cuillière	research co-director
Vincent François	research co-director

Evaluation jury

Pierre-Yves Lajoie	external evaluator (Department of Computer Engineering and Software Engineering, École Polytechnique de Montréal)
Nguyen Tan Sy	external evaluator (Department of Mathematics, Computer Science and Engineering, UQAR)
Luc Laperrière	jury president (DME, UQTR)
Vincent François	research co-director (DME, UQTR)
Jean-Christophe Cuillière	research co-director (DME, UQTR)
Sousso Kelouwani	research director (DME, UQTR)

Thesis defended on 6 05 2026

Résumé

La planification efficace de trajectoires pour les véhicules autoguidés (AGVs) dans des environnements à grande échelle constitue un défi majeur pour les systèmes de navigation autonome, nécessitant un équilibre entre la longueur du trajet, l'efficacité computationnelle et la distance par rapport aux obstacles. Les cartes traditionnelles à représentation unique peinent à fournir des solutions optimales dans de vastes environnements en raison de limitations computationnelles, d'une augmentation de la longueur des trajets et de préoccupations liées à la sécurité. Les approches hybrides existantes se concentrent principalement sur la réduction des coûts de calcul, sans exploiter pleinement les avantages des différentes représentations.

Cette recherche introduit une nouvelle carte hybride en mosaïque (carte combinant des éléments quadrilatéraux et triangulaires), pilotée par l'IA, qui sélectionne dynamiquement la représentation la plus efficace pour chaque région d'un grand environnement. L'approche utilise un modèle de réseau de neurones convolutifs (CNN) entraîné, démontrant des performances supérieures comparées à l'utilisation exclusive des représentations TRI (triangulaire) ou QUAD (quadrilatérale). Le modèle identifie la taille de tuile optimale en évaluant la contribution de chaque type de représentation, supprimant ainsi le besoin de métriques prédéfinies reliant la distribution des obstacles à la sélection de représentation.

Des expériences ont été menées avec le même robot dans divers environnements, y compris des simulations à grande échelle utilisant le framework ROS et des scénarios réels. Les tests ont été réalisés avec une carte à deux tuiles, le robot opérant dans un espace restreint. La méthode proposée a maintenu une augmentation de la longueur du trajet inférieure ou égale à 10 % dans certains scénarios, tout en réduisant la complexité de 25 % par rapport à la représentation TRI, une méthode couramment utilisée en planification de trajectoire. Elle n'a engendré qu'une augmentation de 4 % de la longueur du trajet par rapport à la représentation QUAD largement adoptée en robotique, offrant ainsi une solution efficace sans compromis significatif sur la distance.

Les contributions de cette recherche incluent l'introduction d'un indice de contribution, permettant d'analyser de vastes environnements pour déterminer la taille de tuile la plus efficace parmi plusieurs options. L'étude prend également en compte l'impact des couches d'inflation sur l'efficacité des différentes représentations et propose l'utilisation d'un maillage triangulaire variable comme approche concurrente aux maillages QUAD et triangulaires grossiers. Ce travail établit un cadre généralisable pour la planification efficace de trajectoires dans des environnements vastes et encombrés, adaptable à

différents types de représentations, d'algorithmes de recherche et de stratégies de cartographie. Contrairement aux études précédentes qui se concentrent uniquement sur les métriques computationnelles, les environnements de petite taille ou des dimensions de robot prédéfinies, cette approche prend en compte divers facteurs environnementaux, différents types de robots et l'effet des couches d'inflation (couche élargie autour des obstacles, tenant compte de la taille du robot et d'une marge de sécurité). La méthodologie proposée améliore significativement l'efficacité de la planification de trajectoires dans de grands environnements, tout en assurant sécurité et performance computationnelle, ce qui en fait un outil précieux pour les systèmes autonomes en conditions réelles.

Abstract

Efficient path planning of automated guided vehicles (AGVs) in large-scale environments is a critical challenge for autonomous navigation systems, requiring a balance between path length, computational efficiency, and obstacle clearance. Traditional single-representation maps struggle to provide optimal solutions in large environments due to computational limitations, increased path lengths, and safety concerns. Existing hybrid mapping approaches primarily focus on reducing computational costs but do not fully leverage the advantages of different representations.

This research introduces a novel AI-driven Tiled Hybrid Representation Map (a map combining quadrilateral and triangular elements) that dynamically selects the most efficient representation for each region in a large environment. The approach utilizes a trained Convolutional Neural Network (CNN) model, demonstrating superior performance compared to using only TRI (triangular) or QUAD (quadrilateral) representations. The model identifies the optimal tile size by evaluating the contribution of each representation type, removing the need for predefined metrics linking obstacle distribution to representation selection.

Experiments were conducted on the same robot in various environments, including large-scale simulations using the ROS framework and real-world scenarios. The testing utilized a two-tile map setup, with the robot operating under space constraints. The proposed method maintained a path length increase of no more than 10% for specific scenarios, while achieving a 25% reduction in complexity compared to the TRI representation, a commonly used method for pathfinding. The approach also resulted in only a 4% longer path length than the widely adopted QUAD representation in robotics, providing an efficient solution without significant sacrifice in path length.

The contributions of this research include the introduction of a contribution index, which aids in analyzing large environments to find the most efficient tile size among various options. The study also considers the impact of inflation layers on the effectiveness of different representations and proposes the use of a variable triangular mesh as a competitive approach to QUAD and coarse triangular meshes. This work establishes a generalizable framework for efficient path planning in large, cluttered environments, adaptable across various types of representations, search algorithms, and mapping strategies. Unlike previous studies that focus only on computational metrics, small-scale environments, or predefined robot sizes, this approach considers diverse environmental factors, various robot sizes, and the impact of inflation layers. The proposed methodology significantly enhances path planning efficiency in large environments while maintaining safety and computational performance, making it a valuable tool for real-world autonomous systems.

Acknowledgment

My soul, life, and even destiny resemble the subject of my thesis—a tiled division, where each tile is filled and shaped by different individuals, both in small and large ways, yet together, they have formed the path that brought me here.

I wish to express my heartfelt gratitude, tile by tile, starting from the coordinates (0,0)—my parents, Khosro and Fereshteh and Armin my twin brother—whose immense self-sacrifice laid the foundation of my journey. Moving toward (N,N), I am deeply grateful to my family—my wife Farnoush and my daughter Ronia —whose loves and supports have been my greatest motivation through all challenges.

Along this path, I extend my appreciation to my dear friends Mehdi and Mohsen, who helped me find a place at UQTR, and Jean-Christophe, the first professor to trust me and offer an opportunity to return to the world of science after several years. Sousso has not only been an exceptional supervisor but also a mentor who taught me how to blend kindness with wisdom while guiding me into the realm of AI. Vincent, with his generosity, empowered me to bring my ideas to life through coding, bridging the language barrier between us—not with French, but with C++. Ali, a compassionate friend, has been a source of invaluable support and ideas.

Finally, but by no means least, I am grateful to all my colleagues at IRH and ERRICA, the most welcoming and supportive team I have ever encountered.

This journey would not have been possible without the support, kindness, and guidance of so many incredible people.

Table of Contents

Résumé..... vii
Abstract..... ix
Acknowledgment..... x
Table of Contents..... xi
List of Tables..... xiii
List of Figures..... xiv
List of Symbols..... xv
Chapter 1.Introduction..... 1
1.1 Context and Motivation..... 1
1.2 Literature study..... 2
1.2.1 Environment Representations..... 2
1.2.2 Path Planning Algorithms..... 7
1.2.3 Impact of map representation on path planning efficiency..... 15
1.2.4 Environmental Complexity (or difficulty) Metrics..... 18
1.2.5 Hybrid Maps..... 20
1.2.6 Large Maps & Scalability..... 23
1.3 Synthesis of literature review..... 24
1.4 Problem statement..... 25
1.5 Objectives..... 26
1.6 Assumptions..... 27
1.7 Methodology overview..... 28
1.7.1 First Stage:..... 29
1.7.2 Second Stage:..... 30
1.7.3 Third Stage:..... 31
1.8 Thesis structure..... 33
Chapter 2. Evaluating Mesh Representations for Path Planning in Indoor Environments..... 34
2.1 Introduction..... 34
2.2 Article 1..... 35
2.3 Methodology..... 35
2.4 Summary of analysis:..... 35
2.5 Article..... 37
2.6 Conclusion:..... 67
Chapter 3. A Hybrid Representation Map..... 68

3.1 Introduction..... 68
3.2 Article 2 68
3.3 Methodology..... 69
3.4 Summary of analysis..... 71
3.5 Article..... 72
3.6 Conclusion:..... 92
Chapter 4. A Tiled Large Map..... 93
4.1 Introduction..... 93
4.2 Article 3 93
4.3 Methodology..... 93
4.4 Summary of Analysis..... 94
4.5 Article..... 95
4.6 Conclusion..... 117
Chapter 5. Conclusion and Recommendations..... 118
5.1 Conclusion..... 118
5.2 Contributions..... 120
5.3 Recommendations for future directions..... 122
5.3.1 Triangle navigation approach..... 122
5.3.2 Incorporating Path Smoothing..... 123
5.3.3 Adaptive Tile Sizing..... 123
5.3.4 Comparing a variable irregular triangular mesh with the proposed hybrid approach..... 124
5.3.5 Online re-classification of tiles under environmental change..... 125
5.3.6 Incorporating uncertainty and SLAM into the hybrid framework..... 125
Chapter 6. References..... 126

List of Tables

Table 1: different environment parameters.....18
Table 2: Explain structured link gap problem objectives methodology.....29
Table 3: Summary of paper comparisons.....119

List of Figures

Figure 1: Robot navigation framework....1
Figure 2 : Environment representation formats 2
Figure 3: Examples of common 2D environment representation formats 3
Figure 4: A binary grid map 4
Figure 5: Grid-based mapping....5
Figure 6: Applications of polygonal mesh representations....6
Figure 7: Modelling Mars terrain with irregular triangular mesh 8
Figure 9: Search graph global planner, A* approach....9
Figure 10: Sampling based global planner, RRT approach 10
Figure 11: *Illustration of local path planning* 11
Figure 12:Local planner approaches....12
Figure 13: Inflation layer approach....12
Figure 14: Path smoothing 13
Figure 15: Several smoothing techniques....14
Figure 16: Grid and triangle representation 15
Figure 17: Several options for considering representative points within triangular cells....17
Figure 18: Funnel algorithm 18
Figure 19: environment diversity....19
Figure 20: The hybrid map combines 2D occupancy with 2.5D terrain layers 21
Figure 21: A hybrid layered map 22
Figure 22: Illustration of a LOGnet transition 22
Figure 23: Hybrid mapping 23
Figure 24: Synthesis of literature review 25
Figure 25: A large environment with definitions of different areas and tile concept....28
Figure 26: Methodology of the first paper....30
Figure 27: Methodology of the second paper 31
Figure 28: Methodology of the third paper 32
Figure 29: Hierarchical Progression of Tiled Hybrid Representation Map based on three papers....32
Figure 30: Simplified architecture of the convolutional neural network (CNN) 69
Figure 31: Comparison between hierarchical uniform tile division 124

List of Symbols

2D	Two-dimensional
2.5D	Two-dimensional map with height/elevation values.
3D	Three-dimensional
A*	A-star (heuristic graph search algorithm)
AGV	Automated Guided Vehicles
API	Application Programming Interface
CA	Cellular Automata
CDT	Constrained Delaunay Triangulation
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CV	Coefficient of Variation
DWA	Dynamic Window Approach
Env	Environment
FEM	Finite Element Modeling
GIS	Geographic Information System
G	Goal point
H	Overall map entropy
HYMM	Hybrid Metric Maps
HYB-C	Hybrid Conformal
HYB-NC	Hybrid Non-Conformal
ITM	Irregular Triangle Mesh
JPS	Jump Point Search
L_{map}	Large Map
M_{rep}	Map with a Specific Representation
MAGIC	Mesh Automatic Generation with Integrated CAD
NURBS	Non-Uniform Rational B-Splines
PRM	Probabilistic Roadmaps
QUAD	Quadrangle Representation (Uniform Type)
REP	Representation
ROS	Robot Operating System
RRT	Rapidly exploring Random Tree

S	Start Point
SLAM	Simultaneous Localization and Mapping
STL	*STereoLithography* (triangular surface mesh file format)
TEB	Timed-Elastic Band
TRI	TRI Representation (Irregular Coarse Type)
VFH	Vector Field Histogram
VITM	Variable Irregular Triangle Mesh

Chapter 1. Introduction

1.1 Context and Motivation

Automated guided vehicles (AGVs), with their broad application in factories, are becoming an integral element of the future workforce. Their autonomy requires tools for sensing real-world environments, storing inferred information, and using it for tasks like path planning. These interconnected modules form the core of robot navigation, illustrated at a traditional level in Figure 1.

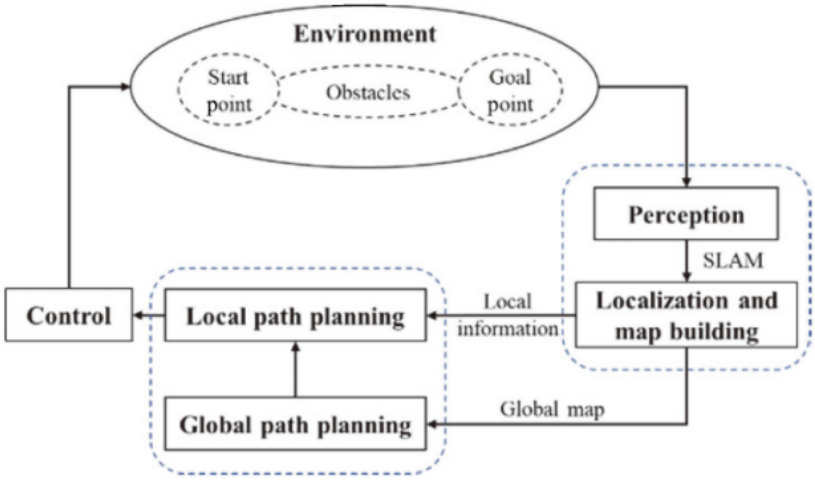


Figure 1: Robot navigation framework (Zhu & Zhang, 2021)

In this presented pipeline, environment representation — or more narrowly, map representation — is a core concept that flows throughout the pipeline with a considerable effect on the efficiency of global path planning. In addition, in large, cluttered indoor environments such as factories, efficient path planning is more challenging due to constraints like limited computational resources, safety, expensive equipment, and budget limitations. Without efficiency, autonomous robots cannot be helpful and reliable tools. One of the less-studied areas is the influence of different environment types on planning efficiency, which can vary with different obstacle distributions in an indoor factory. Furthermore, current single representations cannot offer a complete solution for this problem; therefore, this motivates us to develop adaptive mapping strategies.

1.2 Literature study

The literature review explores foundational work and recent advances in environment representations, path planning algorithms, hybrid maps, efficiency & complexity metrics, and large maps particularly in the context of autonomous mobile robot navigation.

1.2.1 Environment Representations

In the world of autonomous robots, any interaction with the environment requires translating spatial information into a form usable by path planning methods, known as an environment representation. An environment representation is a structured model of space, typically built from basic spatial units such as points, polygons, or voxels, to describe environmental data for navigation and planning. (Figure 2)



Figure 2 : Environment representation formats (Zhang et al., 2023)

The choice of environment representation format depends on the intended application. In robotics, these applications are generally categorized into two major groups: indoor and outdoor. For indoor applications—which are the primary focus of this thesis—2D spatial representation formats are more commonly used due to the prevalence of planar surfaces. Several examples of 2D spatial representation methods commonly used in robotics and geographic information systems (GIS) are illustrated in the Figure 3.

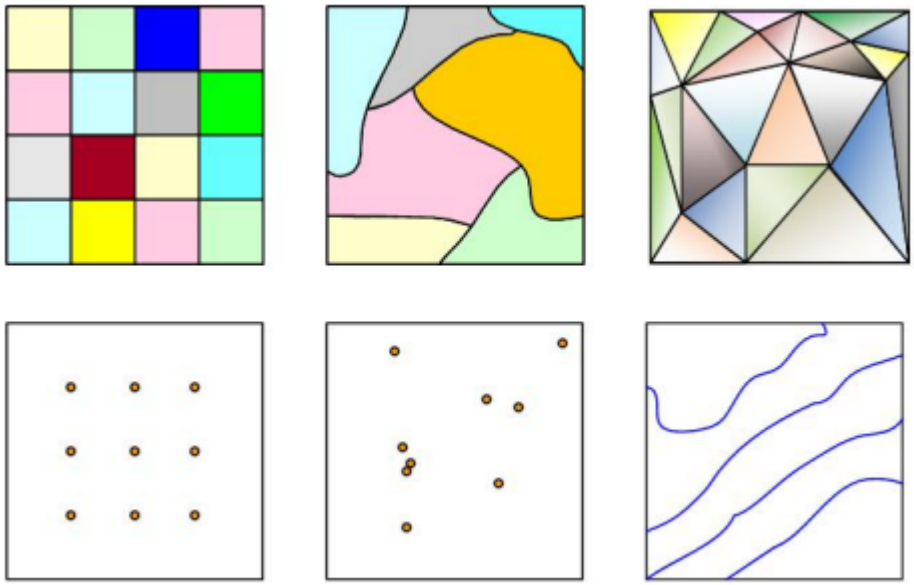


Figure 3: Examples of common 2D environment representation formats used in robotics and GIS, including grid-based, region-based, triangular mesh, point-based, feature-based, and contour-based approaches, each highlights a different way of modelling space, where each unit of representation contains information related to a specific application. (Mostafavi, Beni, & Hins-Mallet, 2009)

In indoor mobile robot applications, where path planning is a primary requirement, spatial information is primarily used to identify traversable areas for navigation. To achieve this, formats such as point clouds, surfaces, and volumetric representations are employed to capture spatial data from sensors. This information is then encoded into structured formats—a process known as mapping.

Mapping, in another sense, refers to a data structure that stores organized spatial data. For example, a grid map stores occupancy information in each cell using binary values (e.g., 0/1 or free/occupied, black/white – Figure 4). Similarly, a polygon map represents obstacles by saving the coordinates of polygon corners in a structured format. Point clouds consist of a set of coordinate points sampled from surfaces in the environment. For 2D navigation, the Z-dimension is often set to zero, reducing the data to a planar representation.



Figure 4: A binary grid map (0 or free space and 1 or occupied space)

Each data structure type, or simply mapping method, behaves differently in processing information through path planning algorithms to find an optimal and safe path within a feasible time. This is due to the fact that each representation contains fundamental information about the environment, distinguishing between occupied and free areas. Additionally, it should consider the free area as a potential traversable space through various strategies such as discretization. This highlights that each mapping method employs a different approach to represent the environment, incorporating critical parameters that explicitly or implicitly define the free area and its connection to occupied regions. A global path planning algorithm must understand how the free areas are distributed to determine a feasible solution across the entire environment. Each mapping method is suited to specific applications, and its effectiveness depends on factors such as environment size, obstacle distribution, and computational constraints.

One widely used mapping method, the occupancy grid map, is notable for its simplicity and robustness, particularly in autonomous indoor navigation. Researchers have addressed some of its limitations, and many platforms support this type of map, making it a common choice for industrial projects. These maps build environments cell by cell independently (Figure 5 - a). In narrow spaces among obstacles, small cells are required for precision, yet these same small cells must also represent open areas far from obstacles. This results in data structures containing excessive redundant data, making them inefficient to store, update, and process in large-scale maps. Although hierarchical representations like quad-tree (Figure 5 - b) and octree mappings attempt to address this issue, their search and update queries are more complex than uniform grid methods. This challenge is intensified in dynamic environments, where maps must be incrementally updated as surroundings change. Moreover, discretization errors at the boundaries of obstacles persist, even with fine cell grids, particularly in high-precision applications. Advancements in robotics technology, which enable new applications, necessitate more advanced environment representations. This factor motivates the exploration of alternative representations.

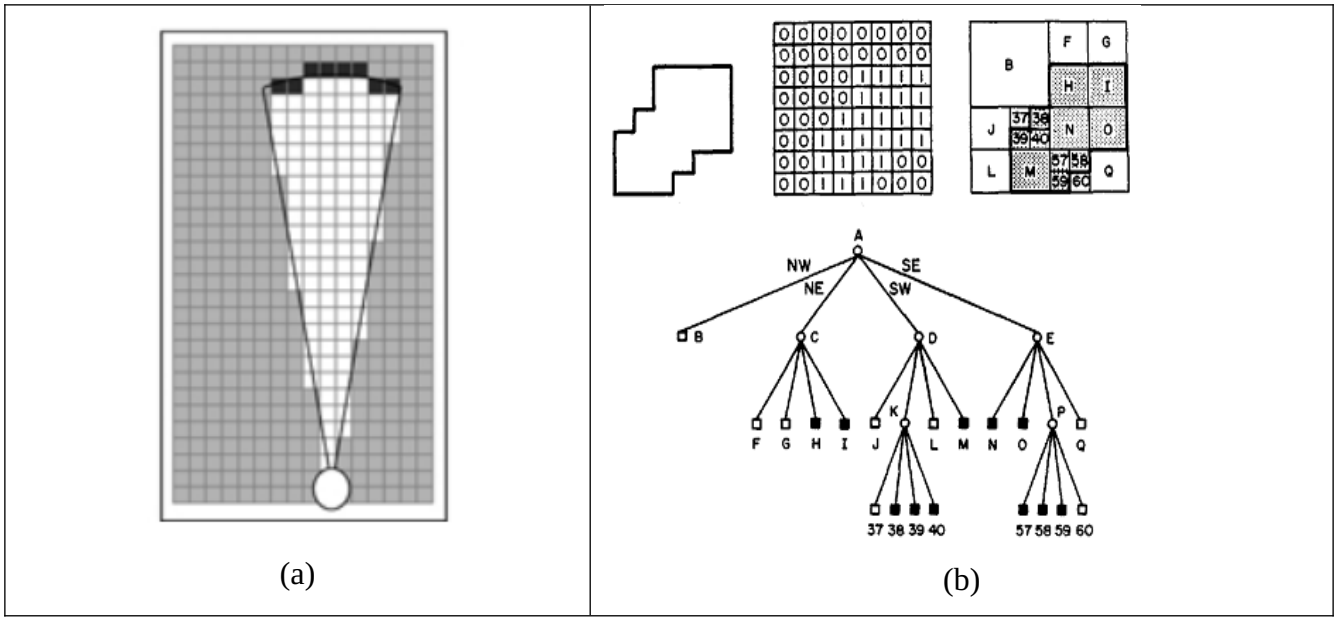


Figure 5: Grid-based mapping. (a) Occupancy grid mapping (Thrun, 2003). (b) Quadtree mapping (Kraetzschmar, Gassull, & Uhl, 2004)

One such alternative is the polygonal mesh which are widely used in reconstruction (Gemme, 2005), computer graphics (Shirley, 2009), and computational methods like finite element modelling (FEM) (Liu, 2013) — an integrated application of all three is illustrated in Figure 6. This representation offers a more compact database, enabling better human interaction and environmental manipulation. Its key advantage lies in accurately representing object boundaries with fewer elements than grid-based methods and the ability to apply elements of varying sizes as needed.

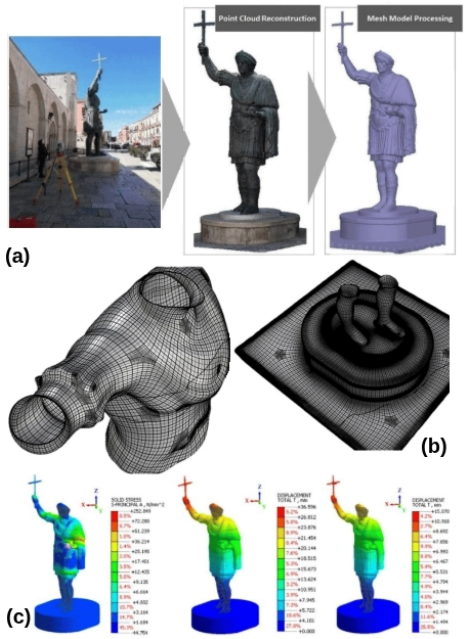


Figure 6: Applications of polygonal mesh representations. (a) 3D reconstruction workflow from point cloud to mesh model (b) polygonal meshes used in computer graphics (c) finite element analysis (FEM) results.(Alfio, V.S., 2022)

Triangular meshes have been applied to 3D terrain modelling and navigation, such as Mars exploration. In outdoor terrains with irregular objects, surface meshing effectively represents environmental geometry and topology. (Figure 7). However, in indoor environments with more planar features, triangular meshes are less popular for robotics navigation, where grid methods dominate. Most applications of triangular meshes are found in video games and virtual reality, where computational efficiency outweighs navigation accuracy. Coarse triangular meshes are often used for faster processing in these industries.

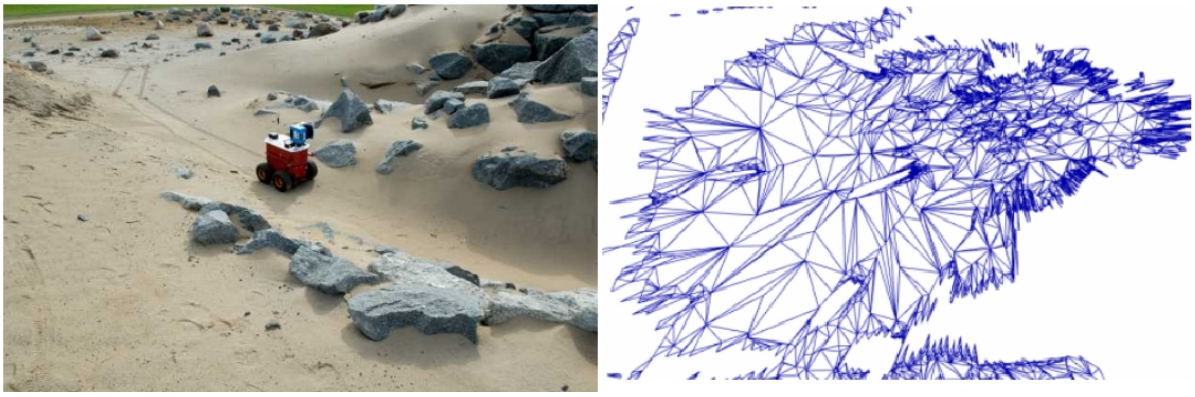


Figure 7: Modelling Mars terrain with irregular triangular mesh (Rekleitis, Bedwani, & Dupuis, 2009)

In real robotics applications, especially those involving expensive equipment and safety concerns, triangular meshes face significant challenges near obstacles. For example, our experiments revealed that paths generated by global planners using triangular meshes were sometimes incompatible with local planners, causing robots to become stuck near obstacles. Reducing mesh size can alleviate this issue but negates the primary advantage of triangular meshes—their efficiency in finding fast paths.

This issue becomes even more pronounced in cluttered, large-scale maps, which require both rapid pathfinding and precise navigation near obstacles. Although both triangle and quadrangle representations can vary in resolution depending on the application, in practice, triangular meshes are often applied at coarser resolutions in navigation meshes used in the game industry, while grid-based methods in robotics are typically implemented at finer resolutions to ensure accuracy and compatibility with local planners. While one advantage of triangular meshes is their ability to adapt mesh size according to local obstacle density, refinement of the mesh in specific areas such as narrow passages or corridors can increase computational cost.

1.2.2 Path Planning Algorithms

Path planning refers to the process of generating a traversable route for a robot within an environment. The path is typically represented as a sequence of connected nodes derived from the underlying map representation (e.g., grid cells or mesh centroids). Path planning, which utilizes environmental information, is typically divided into two approaches: global and local planners, which serve distinct but complementary roles. A global planner uses static information to compute an optimal geometric path, while a local planner attempts to follow this path, taking into account the robot's kinematic constraints and adapting to any dynamic changes in the environment.

Building on this distinction, global path planning has been studied extensively, with a variety of methods developed to compute feasible and efficient paths from a start to a goal location. These methods are commonly categorized into graph-based and sampling-based approaches, each operating in a different representation of the environment. Graph-based methods construct an explicit search graph directly from the map, which discretizes the environment into connected nodes and edges. In contrast, sampling-based methods operate in the robot's configuration space, defined as the set of all possible states (combinations of position and orientation) that the robot can occupy. While the map represents the physical layout of obstacles and free regions, the configuration space captures the feasible motion space considering the robot's geometry and kinematic constraints (LaValle, 2006). Figure 8 illustrates

two distinct views: the map and the configuration space, each used in a different path planning approach—search-based and sampling-based.

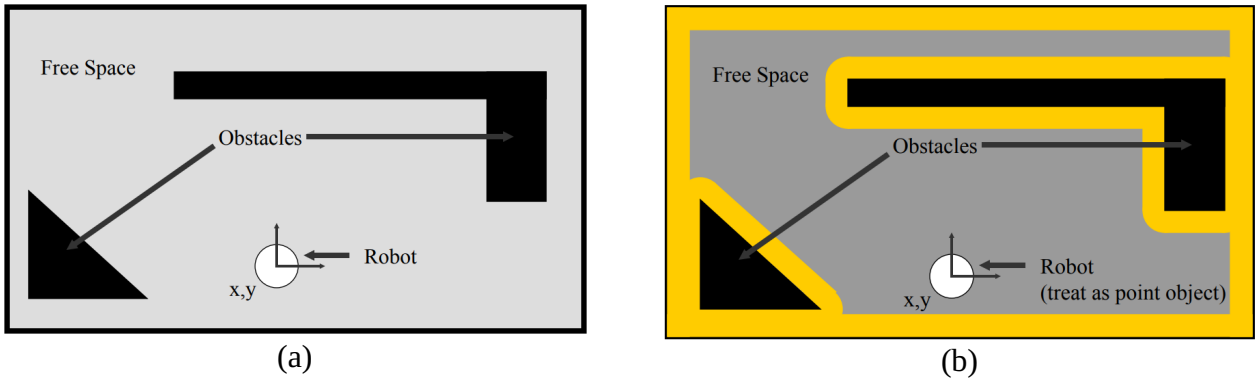


Figure 8: Difference between the map and the configuration space: (a) the physical map showing free space and obstacles with a circular robot, and (b) the corresponding configuration space, where the robot is represented as a point and obstacles are expanded to account for the robot's size. (Lozano-Perez, 1983)

Among the most widely used graph-based algorithms are Dijkstra's and A (Figure 9), which guarantee deterministic and optimal solutions in structured, low-dimensional environments where the map is explicitly represented (Hart et al., 1968) and (Dijkstra, 1959). In these methods, the environment is discretized into a graph G=(V , E), where each node* $v \in V$ *represents a free cell and each edge* $e \in E$ *represents a connection between adjacent nodes. The planner searches this graph to find the minimum-cost and collision-free path from a start node to a goal node. In Dijkstra's algorithm, the total path cost f(n) is evaluated solely based on the accumulated cost g(n) from the start node. A* extends this by incorporating a heuristic function h(n) that estimates the remaining cost to the goal, combining both terms as* f(n)=g(n)+h(n) which significantly improves efficiency in structured maps. These planners rely directly on the map's explicit structure, making them predictable and robust, though potentially computationally expensive when applied to large or high-resolution grids.

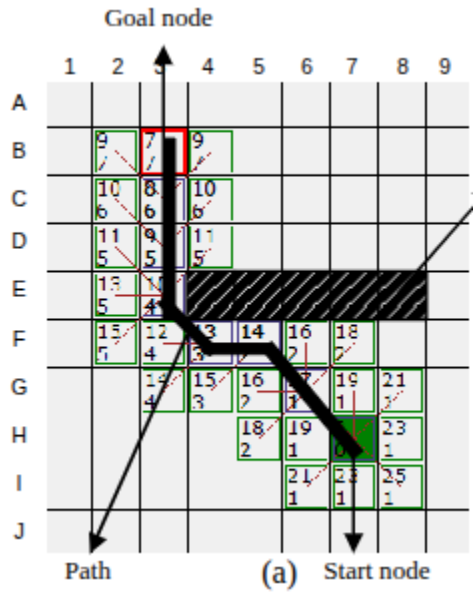


Figure 9: Search graph global planner, A approach (Setiawan, 2014)*

In contrast, sampling-based methods such as Rapidly-exploring Random Trees (RRT) (Figure 10) (LaValle, 1998) and Probabilistic Roadmaps (PRM) (Kavraki et al., 1996) operate in the robot's configuration space C rather than a pre-defined grid. Instead of constructing a full graph G(V,E) over the entire environment, they incrementally build a sparse roadmap by generating random samples $q_i \in C_{free}$ representing feasible (collision-free) configurations. Neighboring samples q_i,q_j are locally connected if the straight-line path between them lies entirely within C_{free}, forming an edge $e_{ij} \in E$. A global path is then obtained by connecting the start configuration q_{start} and goal configuration q_{goal} through this network of connected nodes. These approaches are well suited to high-dimensional configuration spaces (e.g., manipulators with many degrees of freedom) or uncertain environments, since they avoid exhaustive map discretization and efficiently explore complex, continuous spaces. However, in large, fully known indoor environments with complex obstacle layouts, these methods often require extensive parameter tuning and are less practical for tasks where navigation performance must reliably balance multiple objectives such as path length, computation time, and obstacle clearance. Consequently, graph-based methods remain preferable for navigation in structured and well-mapped settings.

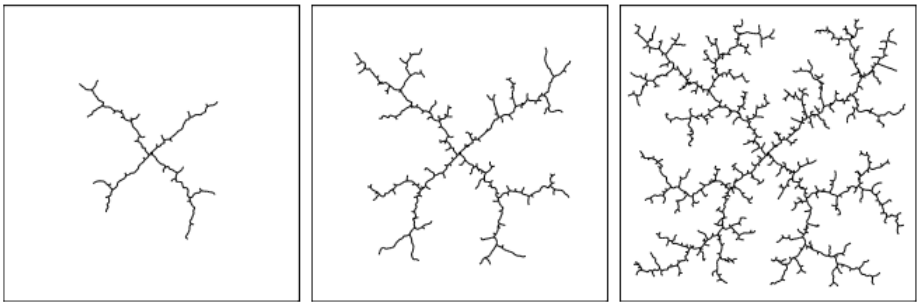


Figure 10: Sampling based global planner, RRT approach (LaValle, 1998)

Next, researchers have explored various refinements to make these planning algorithms more efficient. In terms of handling computational complexity, improvements in path planning algorithms have been extensively explored. For instance, algorithms like JPS(A)* accelerate grid-based search by skipping intermediate nodes and pruning unnecessary neighbors (Duchoň et al., 2014), while Quick-RRT* while Quick-RRT* improves sampling-based planning by biasing samples toward deeper tree regions, enabling faster convergence in maze-like environments (Jeong, Lee, & Kim, 2019). Comparisons between graph-based and sampling-based planners suggest that each approach has strengths depending on environmental conditions. Studies have shown that A* (Hart, Nilsson, & Raphael, 1968) often outperforms RRT (LaValle, 1998) in static environments, whereas RRT is more effective in dynamic settings (Zammit & Van Kampen, 2022). Work by Noreen et al. (2018) has successfully enhanced the performance of RRT* by introducing an adaptive bias (AB) mechanism, improving efficiency in cluttered environments.

Building on these developments, it is important to note that although sampling-based planners, particularly RRT*, have gained significant traction in high-dimensional and dynamic environments, their inclusion as comparison baselines would not serve the objectives of this thesis. Because RRT* operates directly in continuous configuration space without relying on a discretized geometric map, it cannot reflect the influence of different map representations—such as triangular or quadrilateral meshes—which is the central variable under investigation. Furthermore, sampling-based planners are known to exhibit sensitivity to sampling distribution, variance in path quality across runs, lack of guaranteed convergence time in large maps, susceptibility to narrow passages, and substantial parameter-tuning requirements, all of which complicate a fair, representation-focused comparison.

Beyond global strategies, the efficiency of path planning also depends heavily on the local planner’s ability to adapt trajectories in real time. The role of the local planner is to refine the global path into a smooth, collision-free trajectory using real-time sensor input. Like global planning, several methods

exist to perform this task. For example, the Potential Field method (Hwang, 1992) simulates attraction toward the goal and repulsion from obstacles (Figure 11-a), providing a simple framework that can, however, get trapped in local minima. Bug algorithms (Yufka, 2009) guide the robot along obstacle edges until a direct path is found (Figure 11-b), offering simplicity but sometimes leading to inefficient paths.

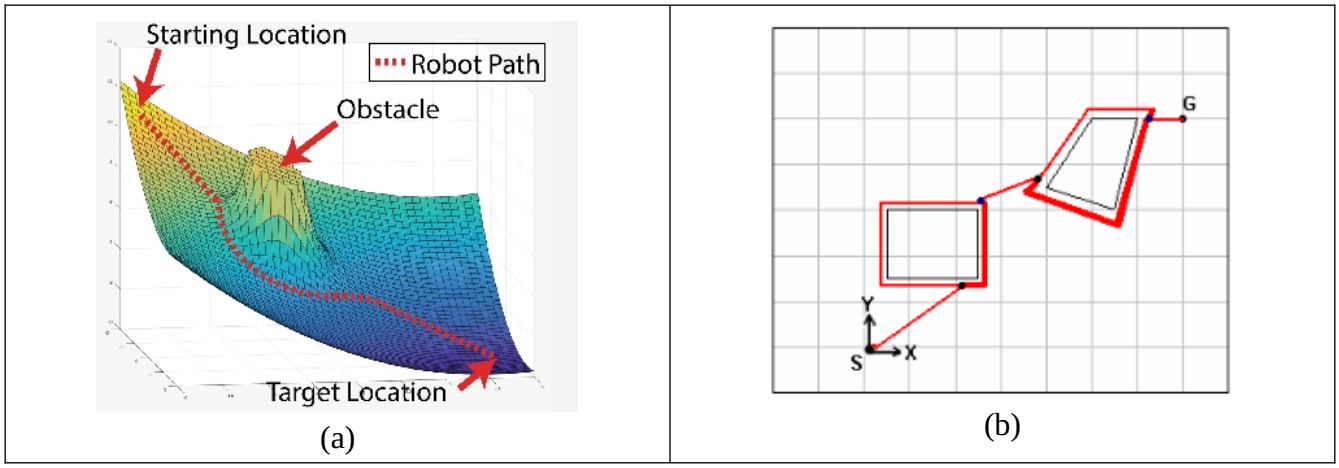
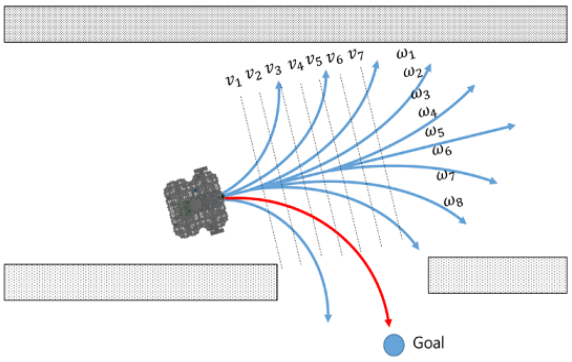
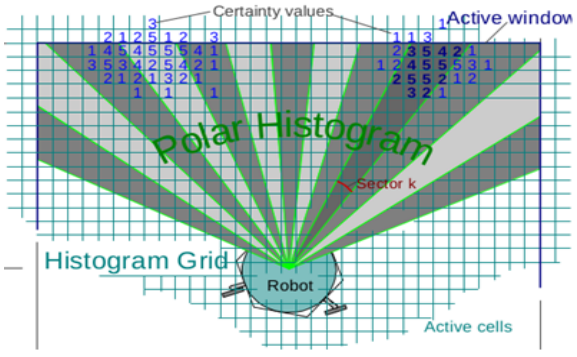


Figure 11: Illustration of local path planning using (a) the Potential Field method (Woods, 2015) (b) Bug1 algorithm (Yufka, 2009).

The Dynamic Window Approach (DWA) – (Figure 12 - a) – selects motion commands by evaluating possible velocity profiles within the robot's limits, allowing for fast local avoidance, although it depends heavily on cost function tuning. Timed-Elastic Band (TEB) uses a deformable path model over time to compute feasible, smooth trajectories, performing well with dynamic constraints but at the cost of higher computation in complex environments. Another method is the Vector Field Histogram (VFH) – (Figure 12 - b) – which creates a polar grid to identify open directions quickly, offering fast reactions but limited success in highly cluttered environments. In this thesis, the choice of a local planner serves mainly to validate the proposed mapping strategy within a real robotic setup, such as in the ROS framework.





(a) (b)

Figure 12:Local planner approaches (a) DWA (Pyo, Cho, Jung, & Lim, 2017) (b) the vector field histogram (Borenstein & Koren, 1991)

An important consideration in both global and local planning is the robot's physical size, which directly influences the feasibility and safety of planned paths. In global planning, this is typically addressed by inflating obstacles using the robot's effective radius plus a safety buffer (as shown in Figure 13), where the effective radius accounts for circular or non-circular footprints. This process is formally related to configuration space (C-space) expansion, where obstacles are grown to reflect the robot's shape and dimensions (Latombe, 1991). Local planners often consider the robot's exact footprint during trajectory generation, enabling finer collision avoidance and improved maneuverability in dynamic settings.

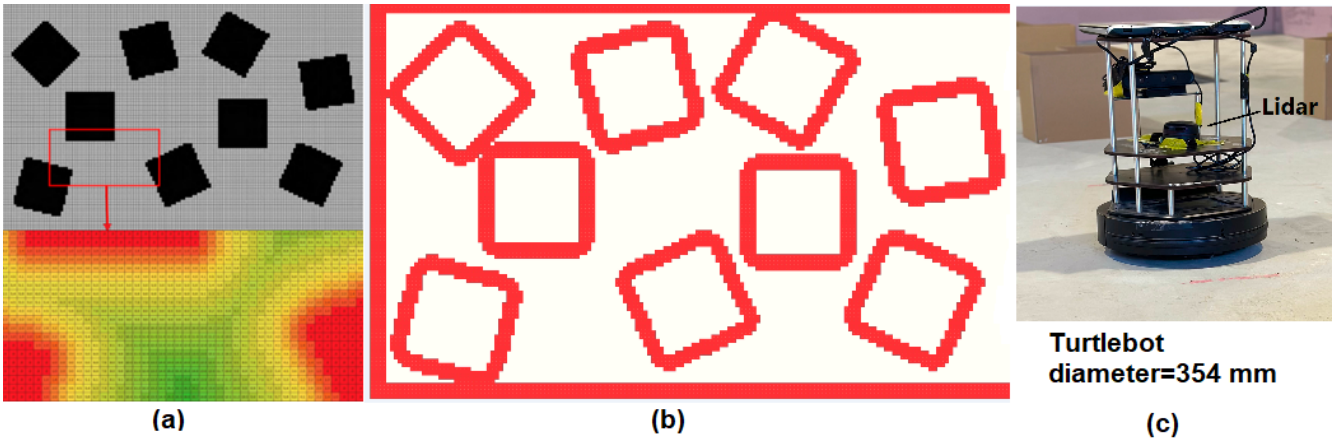


Figure 13: Inflation layer approach. (a) calculation of distance (b) an applied layer (c) Real robot

To refine the computed path, smoothing techniques (Figure 14) are often applied at the global planning stage . These methods aim to eliminate sharp turns and create shorter, more natural-looking paths by modifying the geometry of the path while still avoiding obstacles.

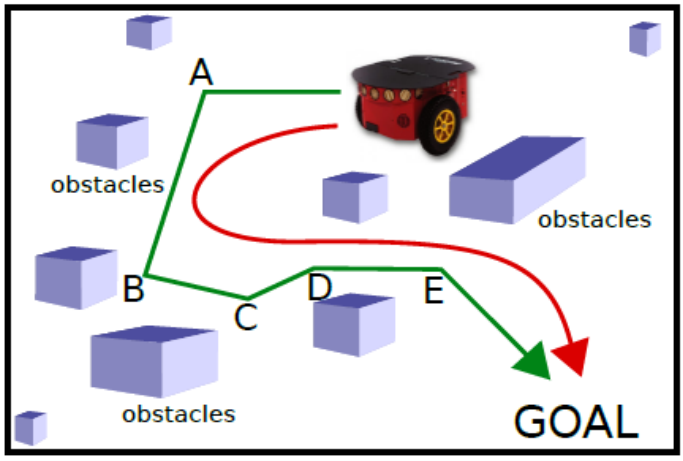


Figure 14: Path smoothing (Ravankar, Ravankar, Kobayashi, Hoshino, & Peng, 2018)

As summarized by (Ravankar, Ravankar, Kobayashi, Hoshino, & Peng, 2018), these methods can be broadly categorized into shortcut-based, optimization-based, and curve-fitting approaches. Shortcut-based methods iteratively replace sections of the path with direct line segments if they remain collision-free, thereby reducing unnecessary intermediate points. Optimization-based methods minimize curvature or path length subject to obstacle constraints. Curve-fitting approaches generate smooth parametric curves through or around control points to achieve natural and continuous trajectories. Several techniques have been shown in Figure 15. For example Bézier curves (*Farin, 2002*), which generate smooth polynomial paths through control points, and B-spline smoothing (*Piegl & Tiller, 1997*), which creates a continuous and adjustable curve based on a series of control nodes, offering both flexibility and control over curvature (Figure 15-c). Another widely used method is NURBS (Non-Uniform Rational B-Splines) (*Piegl & Tiller, 1997*), which extends B-splines by incorporating weights and rational terms, allowing for even more precise control of complex curves and surfaces. However, in this work, we deliberately excluded any global path smoothing, in order to perform a fair and unbiased comparison between different environment representations—triangular and quadrangular—by isolating their effect on the raw output of the path planning process, without additional improvements introduced by post-processing techniques.

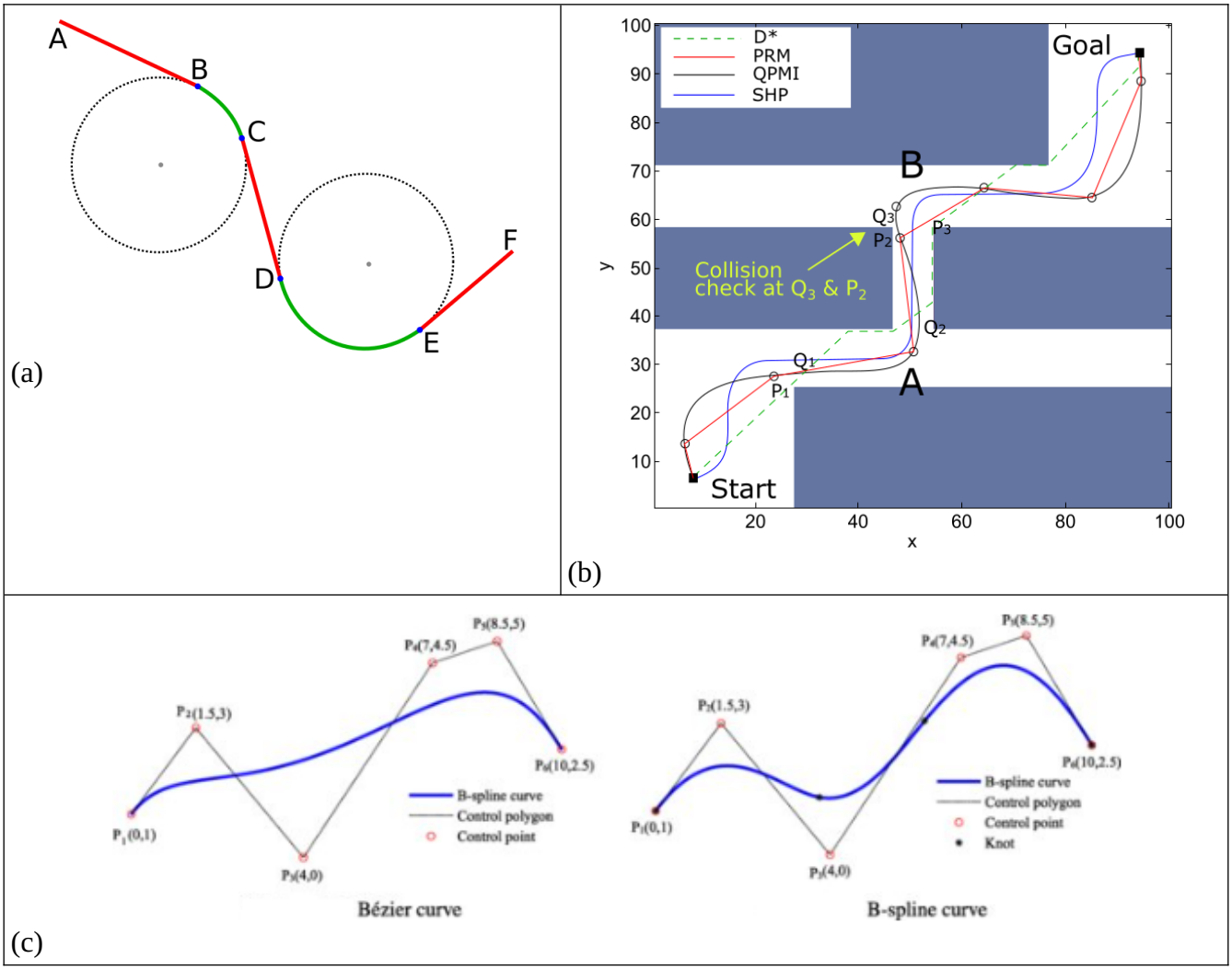


Figure 15: Several smoothing techniques. (a) Dubin's curve, generating the shortest smooth path under turning constraints (Ravankar, Ravankar, Kobayashi, Hoshino, & Peng, 2018) (b) Comparison of path smoothing techniques—QPMI and SHP—applied to a baseline D* and PRM path; QPMI produces a quadratic polynomial path, while SHP generates a spline-based smooth curve through key points. (Ravankar, Ravankar, Kobayashi, Hoshino, & Peng, 2018) (c) Bézier and B-spline Curves (Do, 2019)

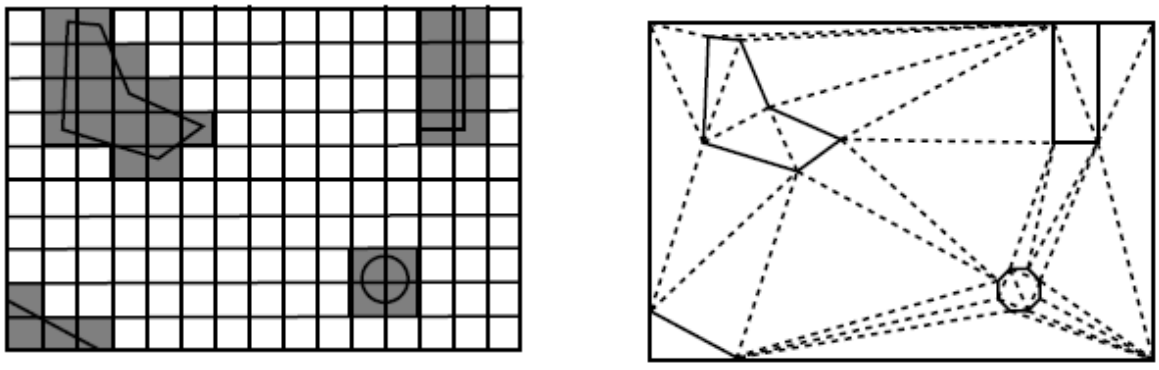
In robotics, path and trajectory are distinct but closely related concepts. A path refers to a spatial sequence of points that defines movement from a start to a goal position, while a trajectory is a time-parameterized version of the path that incorporates speed and timing, allowing the robot's actuators to execute the motion within environmental and physical constraints. This work focuses on path planning, not trajectory generation; however, the resulting paths have been tested in real-world scenarios to evaluate their performance when executed by a mobile robot. Therefore, the primary concern is ensuring path efficiency before sending it to a local planner and control layers (a module responsible for converting planned paths into executable velocity commands for the robot) which will be converted

into a trajectory. Efficient path planning aims to generate collision-free routes that are also optimized for length, safety margins, and computational cost, as the quality of these paths directly impacts their feasibility in real-world environments.

1.2.3 Impact of map representation on path planning efficiency

Path planning efficiency can be characterized by evaluating multiple key parameters, including path length, number of turns or rotations, computational time, and safety margins such as distance from obstacles (Tsardoulias, Iliakopoulou, Kargakos, & Petrou, 2016). Achieving efficient path planning becomes even more challenging when these factors—path length, computational load, and obstacle clearance—must be balanced simultaneously within a given navigation scenario. This multidimensional view of efficiency ensures that the resulting path is not only optimal in geometric terms but also practical for real-time execution in robotic systems.

To enhance the efficiency of path planning algorithms, the choice of environment representation plays a critical role. As mentioned earlier, one of the most widely adopted methods, the occupancy grid mapping approach (Moravec & Elfes, 1985), known for its simplicity and robustness, making it highly compatible with graph-based search algorithms. However, alternative discretization method, triangular meshes have also gained attention due to their flexibility in representing irregular geometries. Both methods aim to partition space into manageable cells, yet they differ fundamentally in terms of cell shape, connectivity, and their efficiency in modelling free and occupied areas (Figure 16). These structural differences directly influence the computational cost and quality of planned paths, making the choice of representation a key factor in overall path-planning efficiency.



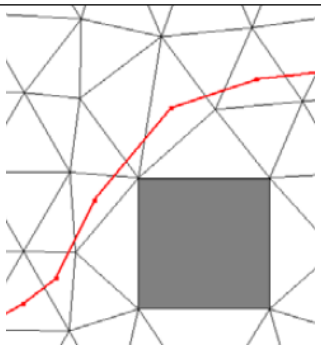
(a) Grid representation with uniform square cells (b) Triangle representation for the same environment

Figure 16: Grid and triangle representation (Demyen & Buro, 2006)

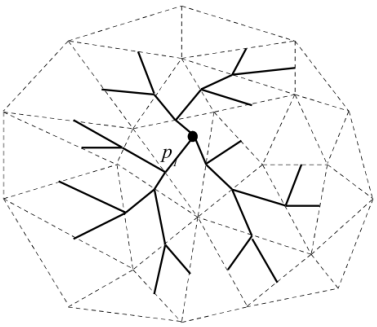
While triangular meshes have proven effective in outdoor and unstructured environments—such as in Mars exploration—path planning for indoor wheeled robots, including Automated Guided Vehicles (AGVs), introduces unique challenges. Accurately defining traversable areas becomes more difficult in confined and cluttered indoor environments. This difficulty arises from factors such as narrow corridors, high obstacle density, and the need to maintain strict safety clearances in confined spaces, which limit mesh simplification and increase computational demands. Although triangular mesh tessellation is widely used in simulations and computer graphics, it involves trade-offs between computational cost and path accuracy. Some strategies attempt to improve accuracy by refining the mesh, but this often leads to increased computational demands, particularly for large environments. Adaptive meshing and simplification methods help alleviate this cost, though they typically introduce significant preprocessing overhead. Moreover, using coarse or unrefined triangles does not necessarily guarantee lower computational effort—especially when safety constraints are critical. In these cases, additional checks for safe movement between neighboring triangles may increase the processing load.

Each cell represents a specific area within the environment, defined by its boundaries. For small, uniformly shaped square cells, geometric properties such as the distance to neighboring cells or nearby obstacles remain nearly the same throughout the entire cell. This allows the center of the cell to be used as a good approximation for calculations such as distance estimation across the entire cell. However, for irregular or larger triangular cells, using the triangle's centroid as a representative point is not always sufficient, especially in processes where safety and obstacle proximity are critical. Due to these limitations, various methods have been proposed to define representative points within triangular cells or to modify pathfinding approaches accordingly. Several well-known methods in the literature are as follows:

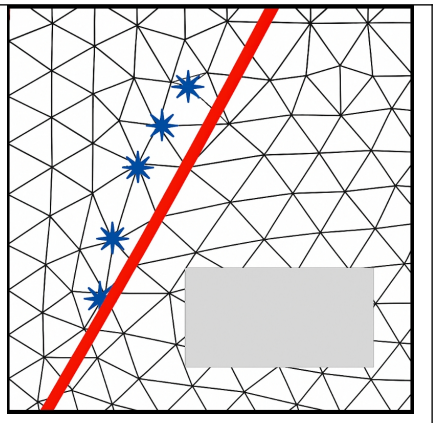
- Consideration of the center of triangles as the graph node for movement between triangles (Demyen & Buro, 2006)
- Consideration of the middle edges of each triangle for movement inside each element (Kallmann, 2005)
- Consideration of triangle vertex as a node and move on the triangle's edge (Y. Liu & Jiang, 2020)



(a) Center of triangles



(b) Middle edge (Kallmann, 2005)



(c) Blue stars, selected vertex, red line a path generated by a simplified algorithm (Y. Liu & Jiang, 2020)

Figure 17: Several options for considering representative points within triangular cells

The first approach involves using the center of each triangle as a representative point. This is applied in two different ways. One method, similar to the current study, processes triangular cells using procedures analogous to those applied to square grid cells - for example, computing traversal cost - while adding extra steps to account for safety margins. The other method employs a funnel-algorithm (Hershberger & Snoeyink, 1994) to find the shortest path through a sequence of connected triangles. It helps to find a safe and local shortest path in a triangle tessellation (Figure 18). Much of the foundational work in this area is attributed to (Kallmann, 2005). Similar to earlier methods, a dual graph of a triangulation is searched using algorithms like Dijkstra or A*, resulting in a triangle corridor where a local shortest path can be extracted. In gaming, optimality is often secondary to computation speed, and path quality is mainly used to simulate intelligent behavior. Thus, coarse meshes like Constrained Delaunay Triangulation (CDT) (Koefoed-Hansen & Brodal, 2012) are commonly used. However, these often yield local optima due to limited node resolution. In robotics, where energy efficiency and path length are critical, global or near-optimal paths are preferred. Refinement of CDT offers one such solution, though it's rarely explored—except in clearance-based planning (Kallmann, 2010) , where agent size is considered, and constrained edges are refined to handle narrow passages. A notable study comparing CDT with other mesh types in terms of computational efficiency is (Plaku, Kavraki, & Vardi, 2008), showing CDT's advantage in reducing the need for fine-tuning. However, it does not address path optimality.

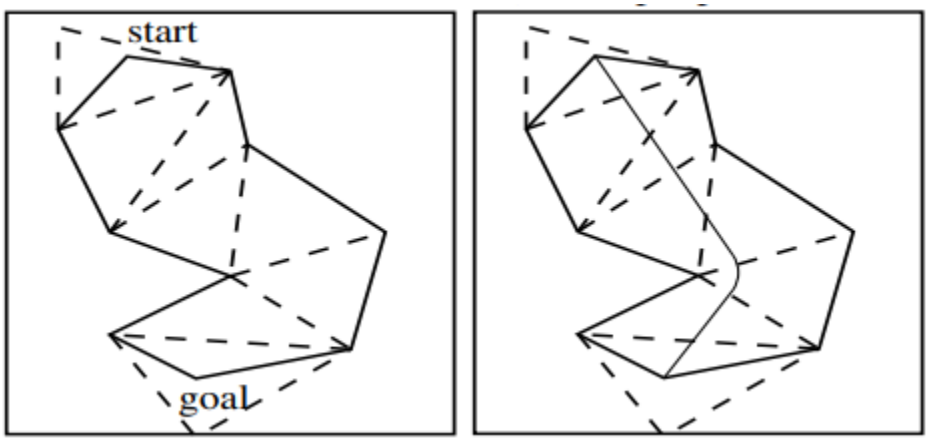


Figure 18: Funnel algorithm (Demyen & Buro, 2006)

1.2.4 Environmental Complexity (or difficulty) Metrics

As map representation strongly influences path-planning efficiency and its performance varies with environment diversity, defining metrics for environmental complexity or difficulty remains an active area of research in autonomous navigation. Environment diversity refers to the variation in obstacle types, density, layout patterns, and free space distribution within a navigable space. A rough classification of different environment parameters has been presented in Table 1 with some illustrations in Figure 19.

Table 1: different environment parameters

Parameters	Description	Examples
Obstacle Density	Refers to how crowded the environment is	Sparse: wide warehouse aisles Dense: storage rooms
Obstacle Shape/Size	Geometry and scale of objects in the environment	Regular: shelves, walls Irregular: furniture, boxes
Spatial Layout	Arrangement of space and obstacle organization	Structured: corridors, hallways Unstructured: open-plan
Navigation Constraints	Features that affect robot maneuverability	Narrow: doorways, tight passages Wide: open lobbies

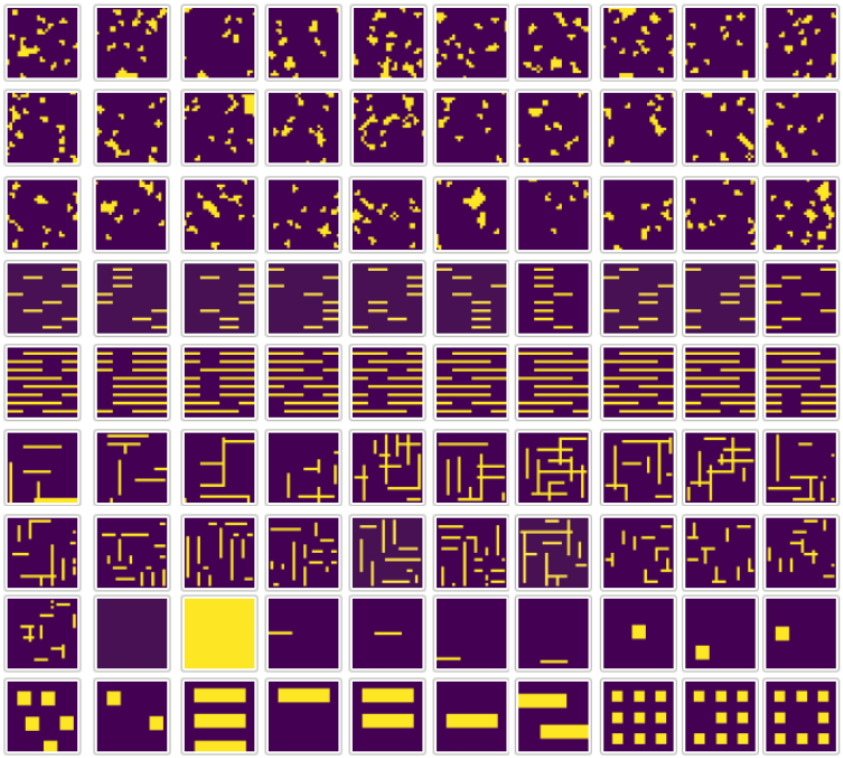


Figure 19: environment diversity

Previous studies have attempted to analyze the effect of environment difficulty on path planning performance, with some focusing on parameters like obstacle shape, obstacle distribution, and distance to the nearest obstacles (Tsardoulias et al., 2016). Entropy-based methods have been proposed to quantify environmental complexity in grid-based representations. Introduced by (Anderson and Yang, 2007) and (Held, Lampe, and Chatila, 2006), these methods apply information theory to describe the variability in free, occupied, and unknown cells within a map. For a grid map with n cells, each cell i has an occupancy probability p_i. The overall map entropy, H is defined as:

$$H = -\sum_{i=1}^{n} p_i \log(p_i)$$

where H increases with greater variability in occupancy probabilities. In a binary occupancy grid, p_i typically denotes the probability of a cell being occupied (p_i=1) or free (p_i=0). High entropy indicates cluttered or irregular regions, while low entropy corresponds to uniform, open areas. This quantitative measure allows researchers to assess how spatial heterogeneity affects mapping accuracy and path planning efficiency. In other words, these studies demonstrate how environmental structure can be numerically characterized to analyze its effect on mapping and path planning performance. A main

drawback of entropy-based measures is that they overlook structural factors — such as narrow passages and regular or irregular obstacle arrangements — that significantly influence navigation complexity. More recent developments, such as Sartori, Ermacora, Pei, Zou, and Yu (2020), applied convolutional neural networks (CNNs) to estimate planning-difficulty metrics derived from similar map parameters, for example obstacle density and free-space connectivity. However, many of these studies overlook the role of navigation scenarios—such as choice of start and goal positions—and fail to propose solutions for handling environment diversity, meaning variations in obstacle layout and connectivity. This diversity directly affects the performance of mapping and path planning strategies, especially in large or cluttered environments.

Additionally, while some research links environmental difficulty to robot traversal time in benchmarking studies (Perille et al., 2020), these works typically assess the overall navigation system performance rather than isolating the impact of different representations on the global planner itself. Despite these efforts, no standardized performance metric has been established across the field. Moreover, as noted by Wang, Zhang, Ma, Li, Ma, Meng, and Meng (2021), hybrid methods that blend learning-based and classical planning approaches often struggle with scalability in large environments.

1.2.5 Hybrid Maps

One of a promising direction in improving path planning efficiency is hybridization, which can be categorized into two main approaches: (1) combining different algorithms to leverage their respective advantages, and (2) integrating multiple mapping techniques to optimize representation. The first approach has been explored in works that combine Voronoi diagrams for safe travel with fast marching methods for smooth path generation (Garrido, Moreno, & Blanco, 2008). However, these methods often require parameter tuning, making them less adaptable to highly variable environments (Tahir et al., 2018).

The second approach, which involves using hybrid mapping approaches have been extensively explored as a means of combining the advantages of multiple map representations to enhance navigation performance (Buschka, 2005). Foundational work by Thrun and Bücken (1996) and later Thrun (1998) pioneered the integration of two distinct map types, setting a precedent for subsequent hybrid mapping studies. Others, such as Hybrid Metric Maps (HYMM) (Guivant et al., 2004), combine feature-based and grid representations to optimize path search complexity. Although AutoInspect (Staniaszek et al., 2024) does not explicitly identify as a hybrid system, it effectively merges

topological maps for navigation with point clouds for localization using a pose graph, thereby uniting metric accuracy with topological planning efficiency.

In outdoor environments, Liu et al. (2023) propose a hybrid map that fuses 2D and 2.5D representations, combining the simplicity of 2D grids with the detailed elevation modeling of 2.5D maps (Figure 20). This approach improves both safety and efficiency in complex terrain and demonstrates the viability of applying hybrid mapping strategies beyond indoor contexts.

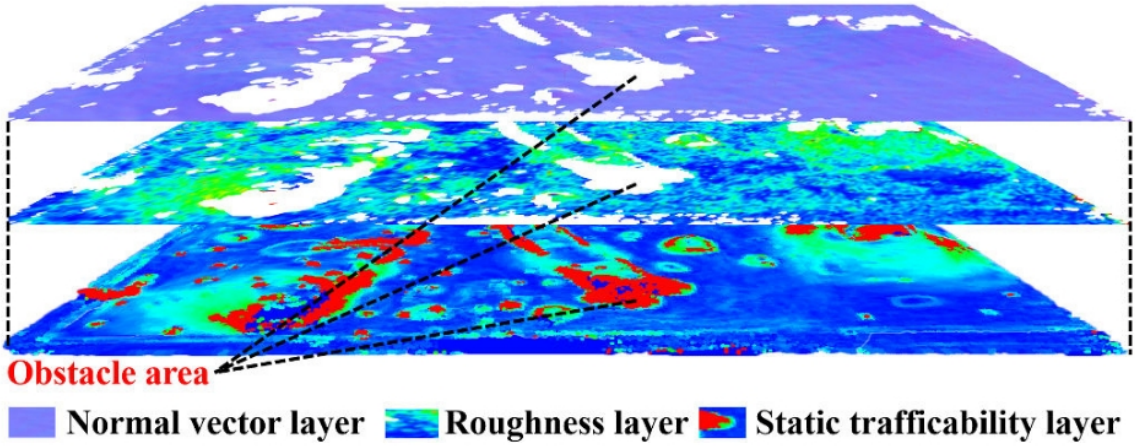


Figure 20: The hybrid map combines 2D occupancy with 2.5D terrain layers—normal vectors and roughness—to improve path planning by capturing both surface orientation and unevenness, enabling safer and more efficient navigation in unstructured environments. (J. Liu et al., 2023)

Other works support this idea by integrating grid and feature-based maps to optimize planning efficiency (Zhong, Tian, Hu, & Peng, 2020). For example, Wang and Chen (2018) suggest a three-layer hybrid system comprising a scene layer (overall layout of the environment), a feature layer (salient landmarks for localization), and a metric layer (precise geometric information for path planning) (Figure 21). However, this method requires manual annotation of turning points and provides no explicit safety framework for navigation near obstacles. Similarly, Song, Liu, Chen, Xu, and Xiong (2024) present a hybrid topological-feature map that employs two types of nodes - one for localization and another for path planning - but it still relies on manually tuned hyperparameters, which limits adaptability in highly cluttered or large environments. Earlier work by Nieto, Guivant, and Nebot (2004) demonstrates how hybrid metric-topological maps can reduce computational load while improving mapping quality by combining coarse topological connectivity with locally detailed metric submaps.

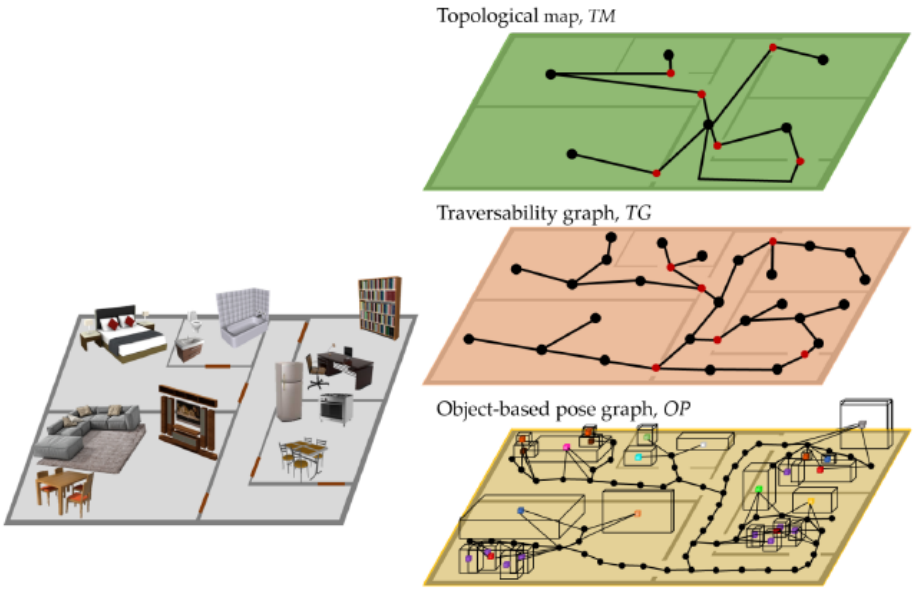


Figure 21: A hybrid layered map (Blázquez, 2021)

The hybrid paradigm is not limited to spatial representations. For instance, LOGnets (Malkin, 1990), introduce a hybrid graph in which each node stores a local spatial map around a landmark, while the edges encode motion commands—such as specific turns or moves—rather than purely geometric connections. This design mirrors human-like navigation by directly linking environmental layout with the sequence of motions needed to traverse it. (Figure 22)

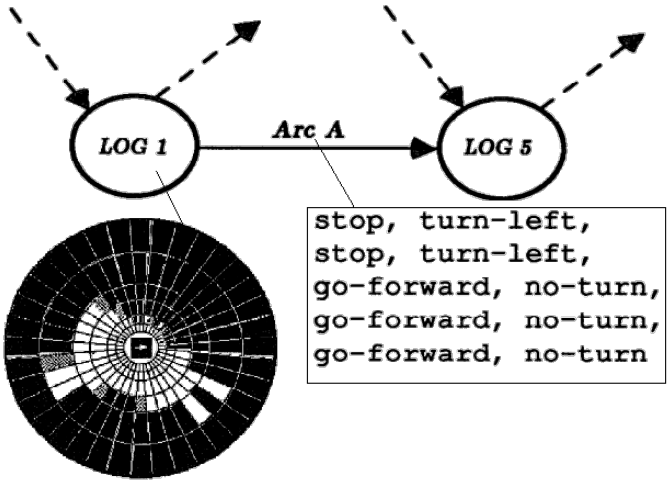


Figure 22: Illustration of a LOGnet transition: Arc A encodes motion commands between local occupancy grids LOC 1 and LOC 5, with directional actions derived from sensor data. (Malkin, 1990)

1.2.6 Large Maps & Scalability

Despite this progress, many hybrid approaches still fall short in accommodating the environmental diversity seen in large, cluttered areas. A unique example is the work by Bibby and Reid (2010), which tackles the complex conditions of marine navigation by using separate models for dynamic, static, and terrain features—Cubic Splines, Point Features, and Occupancy Grids, respectively (Figure 23). Similarly, recent advancements in autonomous underwater vehicles (Svishchev, 2025) use adaptive sensor fusion and hybrid map-matching to adjust sensing and localization strategies according to changing terrain and water conditions. This illustrates environment-aware decision-making, where the system adapts its planning based on the surrounding context rather than relying on fixed parameters.

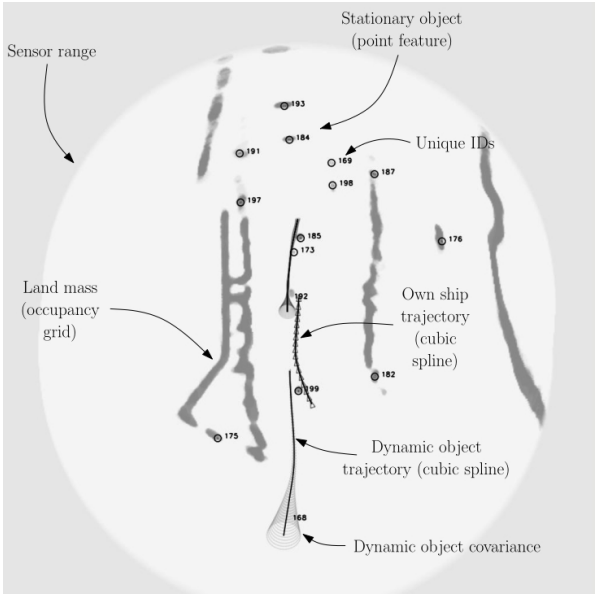


Figure 23: Hybrid mapping - the occupancy grid represents land masses, point features are used for stationary objects and cubic splines are used to represent the trajectories of dynamic objects (Bibby and Reid, 2010).

To address these challenges, a more adaptive strategy involves breaking large environments into smaller regions and selecting the most efficient representation locally for each, enabling both computational efficiency and planning flexibility across diverse conditions. The tile-based concept is well-established in various domains like GIS (Sample & Ioup, 2010), Google Maps (Stefanakis, 2017), and gaming (White, 2008), where it supports scalable data management. In robotics, tiles have been utilized to map and navigate large environments efficiently, as in the mapping of data centers (Mansley et al., 2011). Depending on the application, tiles may be large and few or small and numerous, allowing the resolution to be adjusted based on need.

1.3 Synthesis of literature review

The reviewed studies show that both environment representation and planning algorithms strongly influence path-planning performance. Graph-based planners such as A* remain the most reliable for large, well-mapped indoor spaces, while sampling-based methods offer flexibility in high-dimensional or uncertain settings, though they require careful tuning. In research on large environments, efforts to improve global planning have focused mainly on reducing computational cost, either by refining search algorithms or by introducing hybrid representations to organize environmental information more efficiently. These efforts, however, usually target a single performance metric—such as runtime—while giving less attention to multiple factors like safety, path length, and computational load that together determine overall efficiency.

The choice of map representation shows a similar pattern. Grid and triangular mesh maps each offer distinct advantages: triangular meshes are often applied at coarser resolutions in game-industry navigation meshes, whereas grid-based methods in robotics typically use finer resolutions to ensure accuracy and compatibility with local planners. When triangular meshes are adapted for robotic navigation, researchers frequently apply global or local refinement to improve safety and achieve near-optimal path lengths. Yet almost no study directly examines whether triangular or quadrangular representations are inherently better suited to a given environment.

Even studies that attempt to quantify environment complexity often rely on a small set of manually defined parameters—typically obstacle density, obstacle size variance, or free-space irregularity—to describe how difficult an environment is for navigation. While such parameters offer some insight into environment classification, they capture only limited aspects of spatial complexity and often fail to represent structural variability across large, heterogeneous environments. For instance, two regions with identical obstacle densities may differ substantially in navigability if one contains narrow corridors and the other wide open spaces. To address these limitations, recent AI-driven methods have sought to infer environmental difficulty directly from map data using learning-based models that extract spatial and topological features. However, these approaches remain limited to small environments and have not yet been effectively demonstrated in large-scale, cluttered settings where obstacle distribution and spatial organization vary significantly.

Overall, despite many algorithmic refinements and hybrid attempts, no single approach adapts map structure to local environmental diversity, leaving a critical gap that motivates the hybrid representation proposed in this thesis (Figure 24).

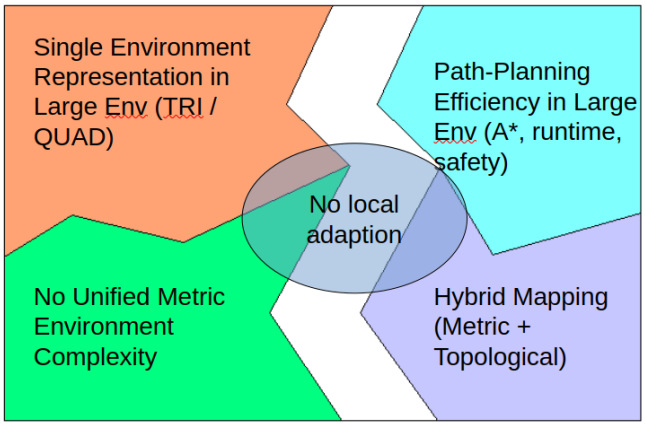


Figure 24: Synthesis of literature review highlighting the lack of local adaptation across mapping and path-planning approaches.

This synthesis reveals a clear research gap: existing map representations for global path planning are predominantly fixed and do not adapt to the local diversity of large, cluttered environments. Consequently, current approaches cannot consistently balance computational efficiency, path quality, and navigation safety across varying environmental conditions.

1.4 Problem statement

Based on this gap, efficient global path planning in large, static, and cluttered indoor environments is still a difficult challenge. Common map representations each have clear drawbacks: quadrangular grids are straightforward and safe but become computationally heavy on large maps, while triangular meshes capture free space compactly but can compromise safety near obstacles and are less straightforward to integrate with conventional grid-based planners, requiring additional steps for graph construction and search in large-scale applications. Most existing approaches commit to a single representation and therefore cannot adapt to the varied obstacle patterns and spatial changes found in real factories and warehouses. This lack of adaptability leads to unnecessary computation, lower efficiency, and reduced reliability for autonomous robots.

These limitations highlight the need for map representations that can adapt to environmental diversity rather than remain fixed—responding to local obstacle patterns and spatial characteristics to balance safety and efficiency. Based on the above discussion, the overall problem can be formulated as a main problem supported by three sub-problems. The main problem is that a single fixed map representation

is insufficient for efficiently handling diverse environmental patterns in large, cluttered environments. This primary issue can be decomposed into three sub-problems:

1. Unclear performance trade-offs between triangular (TRI) and quadrilateral (QUAD) maps
2. The absence of an effective hybrid mapping approach suitable for large environments
3. The lack of an established strategy for environment partitioning

1.5 Objectives

To address the main problem and its three sub-problems identified in Section 1.4, this thesis defines the following objective and sub-objectives. This thesis proposes a new approach for efficiently computing global paths in large, static, and cluttered indoor environments with varying obstacle distributions. To achieve this goal, the following objective and sub-objectives are defined:
Main Objective: Efficient computation of global paths in large, static, cluttered environments with varying obstacle distributions through an adaptive map representation.

The detailed key components introduced in the previous section—covering adaptive representation selection, tile analysis, metric definition, and integration into global path planning—are summarized here as three main sub-objectives.

Sub-objectives

1- TRI vs QUAD Benchmarking

Addressing Sub-problem 1 (unclear trade-offs):

> Characterize and benchmark triangular and quadrangular mesh representations for path planning, including the role of obstacle inflation, to evaluate their respective strengths and limitations. *(Paper 1)*

2- Hybrid Representation Map

Addressing Sub-problem 2 (lack of a hybrid approach):

> Develop and validate a tiled hybrid representation that adaptively selects the most efficient map structure (triangular or quadrangular) for different regions of large, cluttered environments with varying obstacle distributions. *(Paper 2)*

3- Analysis of Environment Partitioning for Local Adaptivity

Addressing Sub-problem 3 (no partitioning strategy):

Investigate the impact of tile partitioning size on path-planning efficiency, safety, and computational performance, and establish practical guidelines for optimal tiling strategies. *(Paper 3)*

1.6 Assumptions

This thesis is developed under the following assumptions:

- Static large cluttered indoor environments: The indoor environments under study are large, static, and cluttered, with obstacle layouts that remain unchanged during path planning.
- Effective Radius Model: The robot's shape is approximated by an effective radius to allow obstacle inflation for safety, accommodating both circular and non-circular platforms.
- No SLAM: The proposed method is limited to the known environment mapping component and is decoupled from the robot's localization system. No SLAM (Simultaneous Localization and Mapping) process is performed, and the focus is solely on mapping and planning.
- No path smoothing: No path-smoothing technique is applied in order to reveal the raw effect of map representation on planning performance.
- A* as the path planner: The A* algorithm is chosen as the path-planning method because of its popularity and established reliability.

At first glance, choosing a static environment, separating mapping from localization to avoid SLAM-related uncertainty, reducing the problem to a geometric map, and using a deterministic planner such as A may appear to place this work in a context where little remains to be solved for dynamic industrial environments. However, the goal of this thesis is not to address the entire navigation pipeline. The focus is specifically on improving the efficiency of global path planning within an existing and operational infrastructure.

Global path planning typically operates on a static map to compute a route from start to goal, while dynamic obstacle handling and trajectory generation are delegated to local planners. Therefore, assuming a static—or more precisely, an episodically static—environment is a deliberate choice that isolates and evaluates the impact of map representation on global planning performance.

To clearly isolate the effect of map representation on global planning performance, SLAM was intentionally omitted. This decision is motivated by two factors: first, triangular meshes as explicit geometric representations are not inherently suitable for managing uncertainty; second, introducing mapping uncertainty would compromise the validity of comparing different map representations. Therefore, localization is assumed to be known, allowing the study to focus solely on mapping.

1.7 Methodology overview

To achieve the main objective and the three sub-objectives defined in Section 1.5, the methodology is organized into three sequential stages, each corresponding to one sub-objective and one published paper. This thesis proposes a hybrid representation mapping approach to build an adaptable map representation that directly addresses the limitations identified in the problem statement. It proposes a learning-based tiled hybrid representation that responds to the diversity of real environments. Rather than relying on a single structure or predefined metrics, the method evaluates local obstacle patterns and spatial characteristics to select between triangular and quadrangular tiles for each region. This adaptive mechanism tailors the map structure to the unique demands of each environment segment, enabling more efficient global path planning. Figure 25 clarifies the concept of tiles in a large environment with varying obstacle distributions, segmented into regions commonly identified as free space, structured areas, random areas, and empty areas. Each tile exhibits distinct spatial characteristics that influence navigation difficulty. In this context, *free space* denotes traversable regions not occupied by obstacles, while *empty areas* represent zones with minimal obstacle density compared to cluttered regions. *Structured areas* contain regularly arranged obstacles, and *random areas* display irregular spatial distributions.

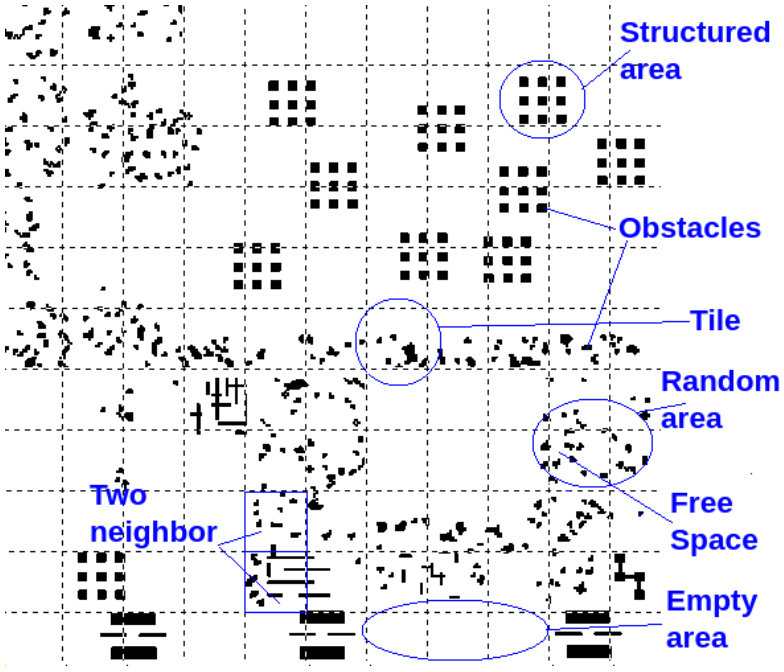


Figure 25: A large environment with definitions of different areas and tile concept. While free space denotes the robot's traversable domain within the configuration space, empty area refers to regions with near-zero obstacle density.

The proposed method is structured into the following key stages, each addressing specific aspects of the problem.

Table 2: Explain structured link gap problem objectives methodology

Research Gap	Research Problem	Research Objective	Methodology	Related paper
Existing fixed map representations cannot adapt to the local diversity of large, cluttered environments.	Unclear performance trade-offs between TRI and QUAD	TRI vs QUAD Benchmarking	Comparative benchmarking across environments, obstacle patterns, robot sizes, and planning scenarios.	Paper 1
	absence of an effective hybrid mapping approach suitable for large environments	Hybrid Representation Map	CNN-based tiled hybrid mapping that predicts the most efficient tile representation using supervised learning.	Paper 2
	The lack of an established strategy for environment partitioning	Analysis of Environment Partitioning for Local Adaptivity	Multi-scale tile-size experiments to assess their effects on computational efficiency, path quality, and safety	Paper 3

1.7.1 First Stage:

Stage 1 addresses Sub-objective 1 by benchmarking TRI and QUAD representations. The initial stage focuses on understanding the behaviour of the A* algorithm on triangular (TRI) representations. Multi-objective weighted variables are defined as metrics to compare the efficiency of paths generated by A*. These metrics are then applied to different environments to identify the types of environments where specific representations yield more efficient paths. The robot's size is also considered a critical parameter affecting safety, which is a key concern in robotics compared to gaming. Additionally, the impact of map resolution on navigation efficiency is analyzed. Benchmarks between TRI and quadrilateral (QUAD) representations are conducted across various environments and scenarios to evaluate the behavior of each map representation in terms of path-planning efficiency. (Figure 26)

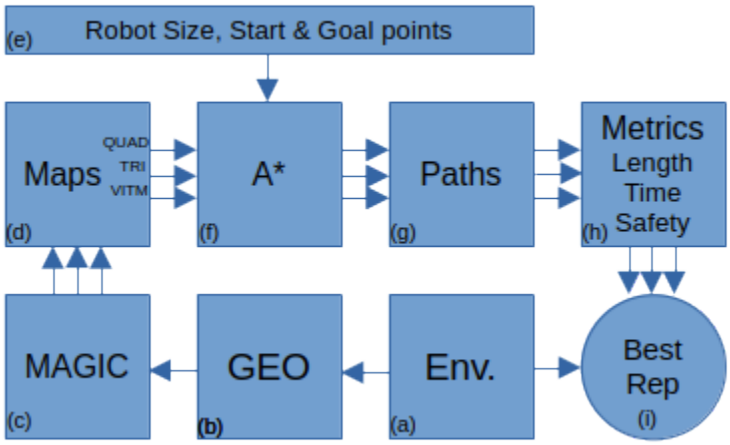


Figure 26: Methodology of the first paper. For a given environment (a), a corresponding geometry (b) is created, and the MAGIC platform (c) is used to generate multiple map types, including QUAD, TRI, and VITM (d). Various parameters such as robot size and start/goal positions are defined (e), and the A search algorithm (f) is applied to find paths (g) on each map. Metrics such as path length, computation time, and minimum distance to obstacles (h) are then calculated to compare the paths. The final goal (i) is to identify the most efficient representation for the environment based on these metrics.*

1.7.2 Second Stage:

Stage 2 addresses Sub-objective 2 by developing the tiled hybrid representation. This stage extends the study from small environments to large ones by introducing a tiling approach. The use of tiles is a well-established concept like Google Maps, where they enhance efficiency and scalability in handling large datasets. In robotics, tiles in (Mansley et al., 2011), are defined physically by the tile of the floor in a data center, the concept can be broadly applied as a general approach, similar to the divide-and-conquer strategy in computer science. The divide-and-conquer principle is a problem-solving strategy that breaks a large problem into smaller, more manageable subproblems, solves each one individually, and then combines the results to get the final solution.

A large map is divided into manageable tiles, each using either TRI or QUAD representation. Different types of hybrid representations are introduced and compared, considering the connections between tiles. The concept of diversity is applied using Cellular Automata (CA) to generate various shapes. A generative approach employing a convolutional neural network (CNN) is then used for supervised training of a model. This model predicts the most efficient representation for a given environment, removing the need to define metrics manually.

Path planning on a tiled map involved projecting a global path from a rough grid of the entire environment to determine local start and stop points within each tile. Path planning inside each tile with local start and stop and then combining them as output for a global path. In addition, The effect of

conformal versus non-conformal triangular mesh boundaries was also examined to assess how boundary alignment influences computational cost and search efficiency. Finally, an experimental study is conducted to validate the proposed method by testing the generated paths in a real-world environment. (Figure 27)

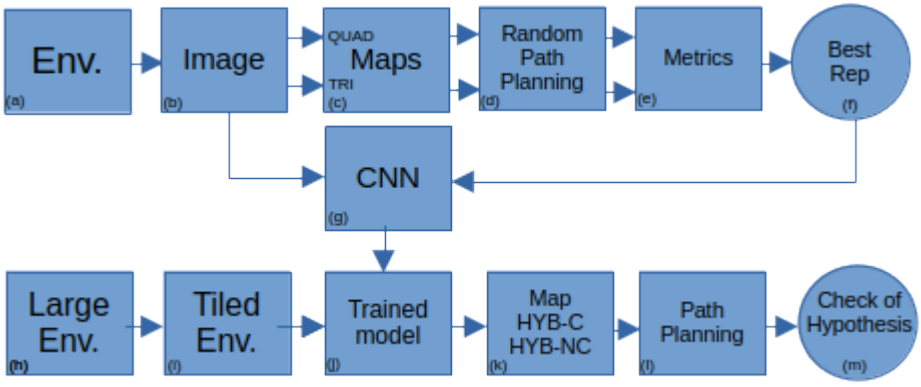


Figure 27: Methodology of the second paper. Similar to the pipeline in the first paper, steps (a) to (f) are used to label a given environment (a) with the most efficient representation based on a set of random path planning scenarios. These labeled images are then used to train a Convolutional Neural Network (CNN) (g). For a large environment (h), a tiling approach divides the space into fixed-size tiles. The trained model (i) determines the best representation for each tile. Based on these predictions, hybrid representation maps (HYB-C and HYB-NC) are constructed (k) and evaluated through path planning (l) to test the hypothesis (m) that a hybrid approach is more efficient than using a single representation.

1.7.3 Third Stage:

Stage 3 addresses Sub-objective 3 by investigating tile partitioning strategies. A divide-and-conquer strategy was first introduced in Paper 2, where the environment was partitioned into independent tiles. Because these tiles were initially unconnected, a procedure was needed to integrate the local paths into a single global route. The process began with a coarse global grid of the entire environment to compute a rough low-cost path, which was then used to determine local start and stop points inside each tile. This approach, however, required extra refinement of the local points and proved difficult to automate. Consequently, the divide-and-conquer idea was retained only for the initial stage, serving to select the most suitable representation (grid or triangle) for each tile.

In the third paper, global path planning was fully automated by eliminating the need for manually defined local start and goal points. A graph was built inside each tile—either a grid graph for quadrangular tiles or the dual graph of a triangular mesh for triangular tiles. These local graphs were then merged into a single global map by linking adjacent tiles through a nearest-neighbor method,

which added inter-tile edges between the closest border nodes. A visibility check confirmed that each connection was physically traversable, and a distance transform of the environment provided distance values for cost calculations and final edge validation.

Segmenting a large environment into uniform tiles allows for either a higher count of smaller tiles or fewer larger tiles, depending on the required resolution. The goal of final stage was to investigate the effect of tile size variation on the proposed hybrid map. A systematic approach is used to expand the tile size across a range of values, analyzing its impact on path-planning efficiency in different environments. This stage also explores how the trained CNN model can be utilized to analyze a map prior to path planning, enabling the selection of the most suitable representation template for optimal performance. (Figure 28)

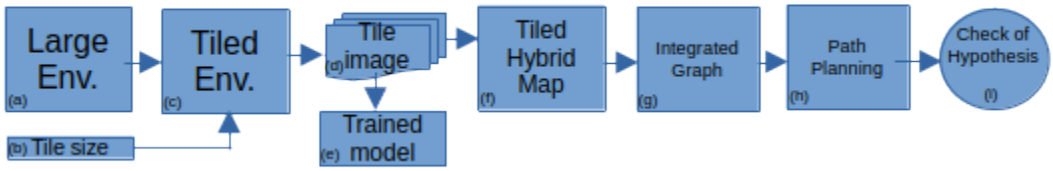


Figure 28: Methodology of the third paper. Similar to the pipeline in the second paper, but with the added flexibility to define tile size (b), allowing the generation of tile maps with varying sizes (c). For each tile (d), the trained model (e) selects the most efficient representation, resulting in a tiled hybrid map (f). This map is then converted into a unified, integrated graph for path planning using the different tile representations (g). Consistent path planning scenarios (h) are executed across different tile sizes to test the hypothesis (i) that the tiled map with the highest contribution from both representations yields the most efficient navigation performance.

The hierarchical approach by showing the core of each paper has been shown in Figure 29.

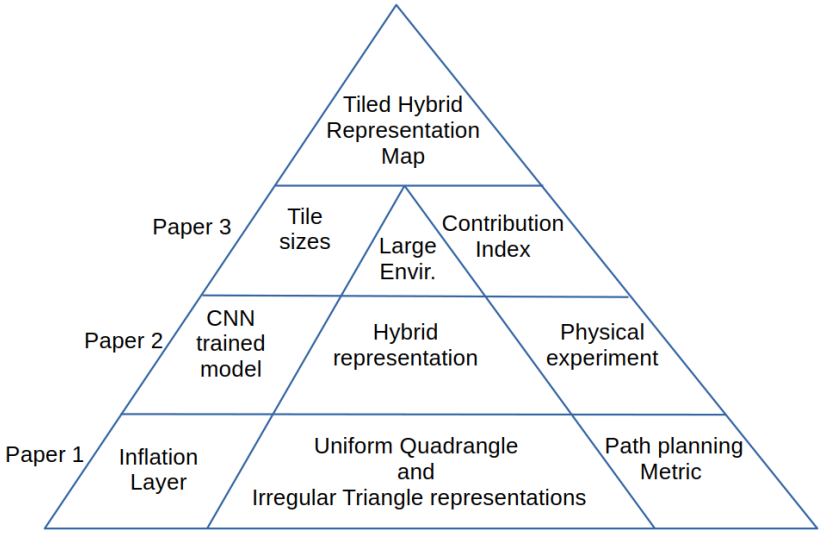


Figure 29: Hierarchical Progression of Tiled Hybrid Representation Map based on three papers

1.8 Thesis structure

The remainder of this thesis is organized as follows: Chapter 2 presents an accurate benchmark of TRI and QUAD representations across different environments. The chapter evaluates which representation provides more efficient path planning based on a defined metric and robot size, as detailed in *Paper 1*. Chapter 3, the core of this work, introduces a tiled hybrid representation map. It incorporates a trained model to identify the best representation for each tile. This chapter also includes an experimental study, documented in *Paper 2*. Chapter 4 expands the concept of the hybrid representation by exploring its application with varying tile sizes in a systematic manner. This approach enables the analysis of large maps to determine the most efficient map template before implementing it, without requiring scenario-based checks. The findings are presented in *Paper 3*. Chapter 5 concludes the thesis, summarizing the contributions, highlighting areas of improvement, and providing recommendations for future work.

Chapter 2. Evaluating Mesh Representations for Path Planning in Indoor Environments

2.1 Introduction

As discussed in the previous sections, the traditional mapping method, such as the occupancy grid (similar in behavior to QUAD), presents challenges as a robust representation for robotics. Similarly, using TRI, commonly known as Nav-Mesh in the game industry, encounters safety issues near obstacles. One common solution is to focus on improving path planning algorithms for each representation to address these problems, which is the most widely adopted approach in robotic path planning.

However, this work explores an alternative approach: employing a hybrid representation method that leverages both QUAD and TRI representations in different environments. This is based on the hypothesis—confirmed in the initial analysis of this study—that each representation produces a different path based on the distribution of obstacles. Consequently, the first critical question addressed in this work is identifying where QUAD and TRI provide better results in path planning.

Additionally, this work emphasizes the practical consideration of robot size as an inflation thickness for safety, which is a crucial factor in robotics. Whether or not to include this consideration is an essential decision for the continuation of this study.

Furthermore, defining and quantifying the concept of an efficient path is vital for comparing the performance of different representations. This foundational work, presented in the first paper, serves as the groundwork for developing a hybrid representation map in subsequent studies.

This chapter directly addresses sub-objective 1 of the thesis, which is to benchmark triangular (TRI) and quadrangular (QUAD) mesh representations to identify their environment-dependent behaviour in path planning.

It is important to highlight a point regarding the abbreviations used for the different triangular mesh types. For the first paper, two different types of TRI representations have been used:

- A near-constant-size element named ITM (Irregular Triangle Mesh)
- Variable-size elements named VITM (Variable Irregular Triangular Mesh)

In this work, the abbreviations ITM and VITM have been used instead of TRI. For the other sections, TRI has been used to represent only ITM with a coarse size.

2.2 Article 1: Investigating the Impact of Triangle and Quadrangle Mesh Representations on AGV Path Planning for Various Indoor Environments: With or Without Inflation

Journal: ROBOTICS (MDPI) (https://www.mdpi.com/journal/robotics)

Authors: Meysami, A.; Cuillière, J.-C.; François, V.; Kelouwani, S.

Accepted date: 10 April 2022

2.3 Methodology

The methodology in this paper involves a comparative analysis of triangle (TRI) and quadrangle (QUAD) mesh representations for path planning in various indoor environments. First, a set of diverse indoor environments with varying obstacle distributions is geometrically created, and three types of representations, including QUAD, ITM, and VITM meshes, are constructed for each environment to serve as the underlying mapping methods. The A* algorithm is employed as the path planning tool with additional calculations and checks applied to both TRI and QUAD meshes to generate paths. These include the calculation of visited cells in the A* method as an alternative metric for CPU time when finding a path and visibility checks for navigating between elements and common node neighbors near obstacles.

Metrics for path efficiency and safety are explicitly defined: efficiency is measured by path length and the computation time required to find the path, while safety is measured by the minimum clearance maintained from obstacles. An inflation layer, representing the impact of robot size (by using inflation thickness), is considered as a key parameter to evaluate safety and feasibility in narrow spaces. Multiple scenarios are then tested, with and without inflation, to study how the robot's size affects path planning for each mesh representation. Finally, TRI and QUAD meshes are compared based on the defined metrics across all scenarios to identify their strengths and weaknesses in path planning for each type of environment.

It is important to highlight a point regarding the abbreviations used for the different triangular mesh types.

2.4 Summary of analysis:

For the first set of scenarios, which consider the pure decomposition effect without incorporating an inflation layer, a coarse ITM proves to be a suitable option for most environments, except in hybrid environments where VITM demonstrates superior performance. Under these conditions, QUAD representation is absent.

However, when the robot size is considered—using an inflation layer to account for a diameter of 354 mm (e.g., TurtleBot 2)—the results shift significantly. This adjustment introduces limitations that affect movement between neighbors, altering the complexity and influencing the choice of the best representation. In this updated analysis, unlike the scenario without inflation, all ITM representations fail to find a path and are replaced by VITM and QUAD representations. Additionally, finer VITM triangles are required in narrow passages that are not fully obstructed. Reducing triangle size (and thereby increasing their number) enhances the likelihood of finding more triangles with centers located in free space, enabling feasible movement between them and their child elements.

In summary, coarse ITM delivers optimal performance in environments with a single obstacle or hybrid layouts featuring ample open space. For corridors and environments with regular multi-obstacles, the coarse grid is the most effective. Conversely, in irregular multi-obstacle setups and curvilinear passages, VITM emerges as the best-performing representation.

robotics

MDPI

2.5 Article

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

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

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

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 flexi- bility 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 [7his 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 represen- tation 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 defini- tion 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 topologi- cal 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.

1.1. 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}.

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

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:

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

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:

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

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

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

near obstacles. To do so, Mov c_i, c_j is defined as:

If 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:

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

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 $g(ci)$ is given by:

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

where 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 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:

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

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}:

$$\tau_{opt} = \{c_s, c_1, c_2, \ldots, c_g\}, \forall\ c_i \text{ and } c_j,$$

$$c_i = \mathrm{par}\ c_j \Longrightarrow L_{\tau_{opt}} \approx \min(L_\tau) \quad (8)$$

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*.

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

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

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

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

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

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:

$$D_{min,reps} = \min(DT(\tau)) \text{ which } \forall c_i \in \tau \bigvee cent(c_i) \in c_{RG,k} \quad (12)$$

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.

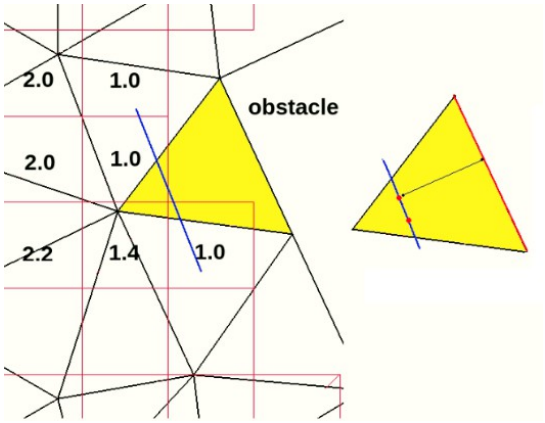


Figure 1. Methodology for calculation of distance when the checking point is not placed inside any cell. 1.0 shows the minimum number in a grid, but some values can be calculated based on the edge of nodes shared with obstacles boundaries.

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

$$\begin{cases} LN_{reps} = \frac{L_{\tau opt,\ reps}}{L_{max}} \\ CN_{reps} = \frac{C_{reps}}{C_{max}} \\ DN_{reps} = \frac{D_{min}}{\ } \end{cases} \quad (1$$

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 obsta- cle 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:

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

where ω_1, ω_2 and ω_3 are the weights. For simplicity, equal importance is considered for all weights: $\omega_1 = \omega_2 = \omega_3 = 0.333$.

1.2. Decomposition of Inflation Layer

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

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

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:

$$M_{reps} = \{\forall\, c_i (i = 1, 2, \ldots N);\ \sum c_i \approx W_{free};\ cent(c_i) \quad \text{inflation layer}\} \quad (16)$$

5-Results and Discussion

This section presents the results regarding the AGV performance in different condi- tions. 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.

1.3. 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 environ- ments 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 obsta- cle'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. Var- ious 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 select- ing 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.

1.4. 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 representa- tion. 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.

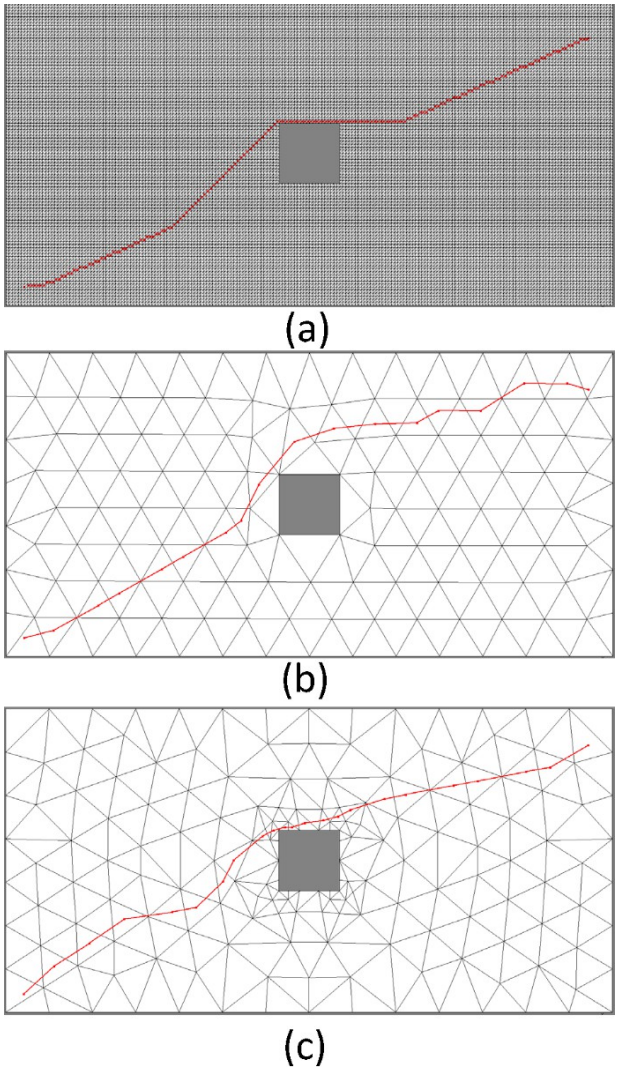


Figure 2. Different paths for a fixed scenario in an environment according to different representations. (**a**) A GRID 200 × 100. (**b**) An ITM-700. (**c**) A VITM-100-700.

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	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

1.5. 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.

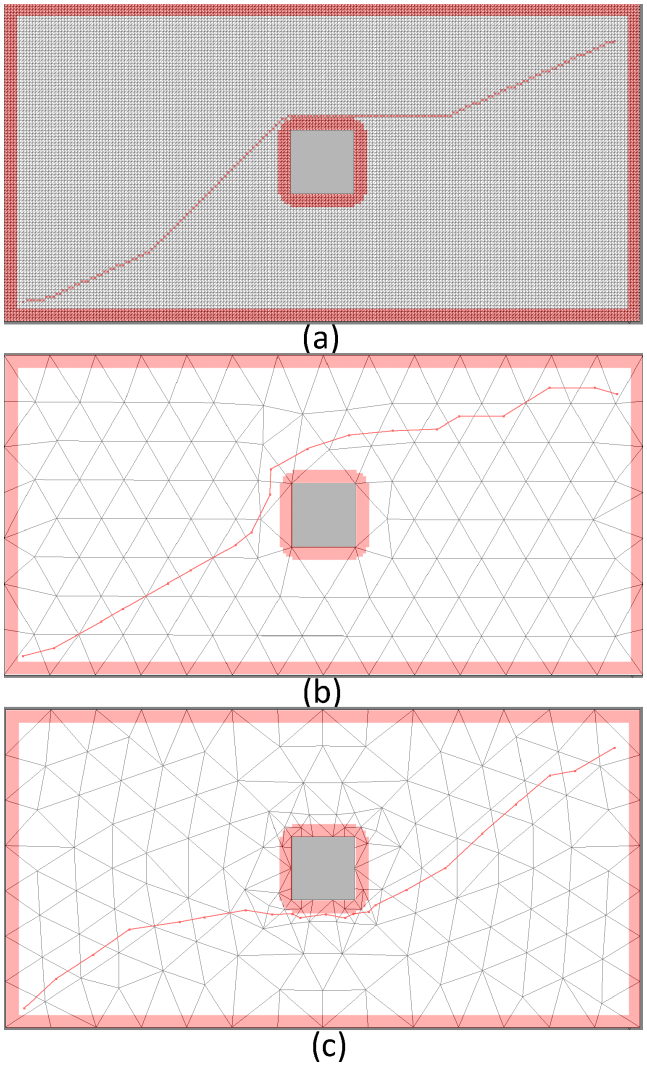


Figure 3. Different paths for the medium-size obstacle scenario considering the inflation layer. (**a**) A GRID 200 × 100. (**b**) An ITM-700. (**c**) A VITM-100-700.

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 Cost Function
Item	Type	Size	Length (mm)	Min Distance (mm)	Complexity	
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 considera- tion 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-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

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

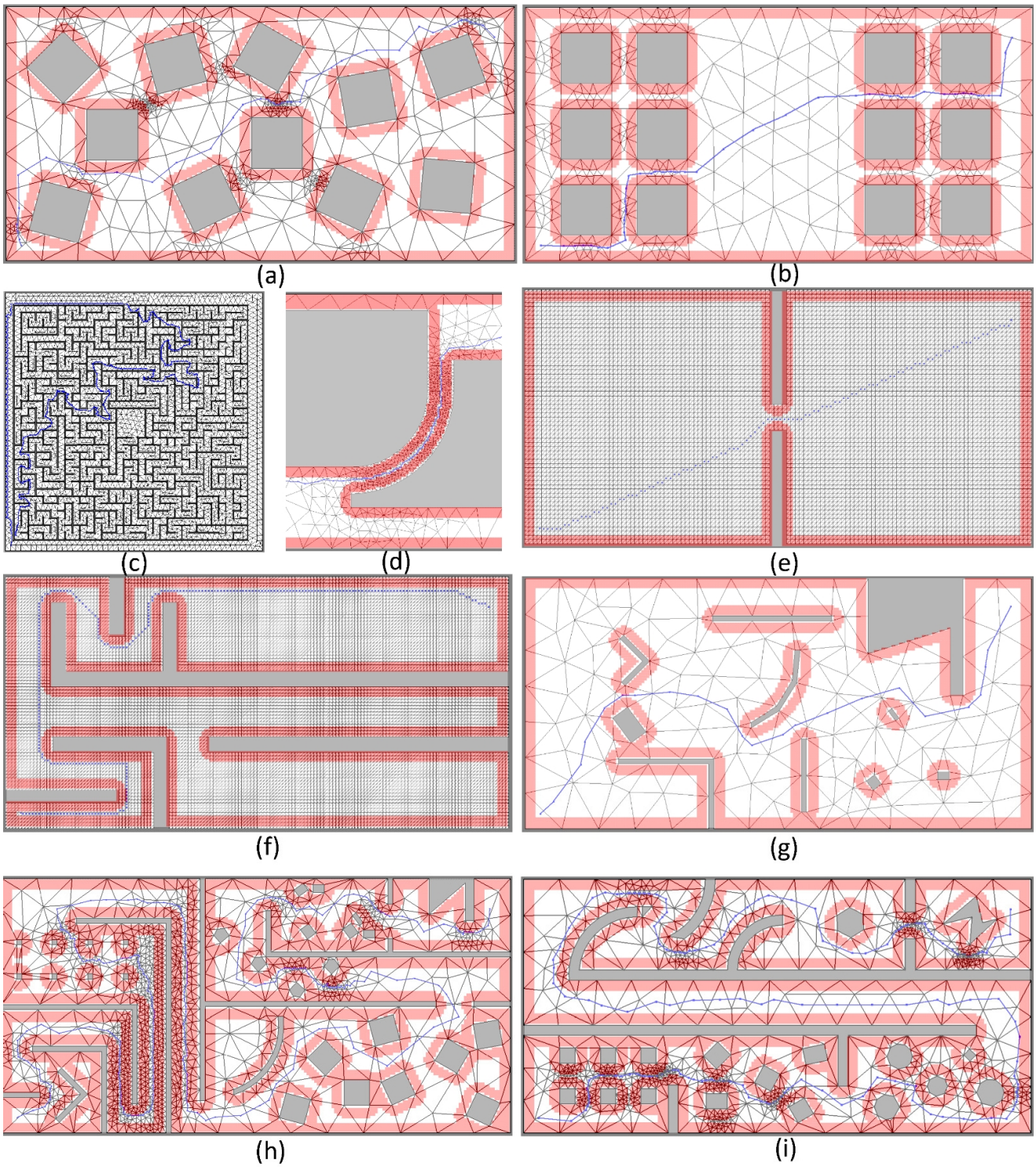


Figure 4. Different paths for some environments. (**a**) Blue path for Map_10-VITM-30-700. (**b**) Blue path for Map_8-100-700. (**c**) Blue path showing the capability of ITM in a maze map. (**d**) Blue path passing a narrow curvilinear passage by VITM method. (**e**) Blue path for a narrow passage 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.

1.6. 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.

7-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

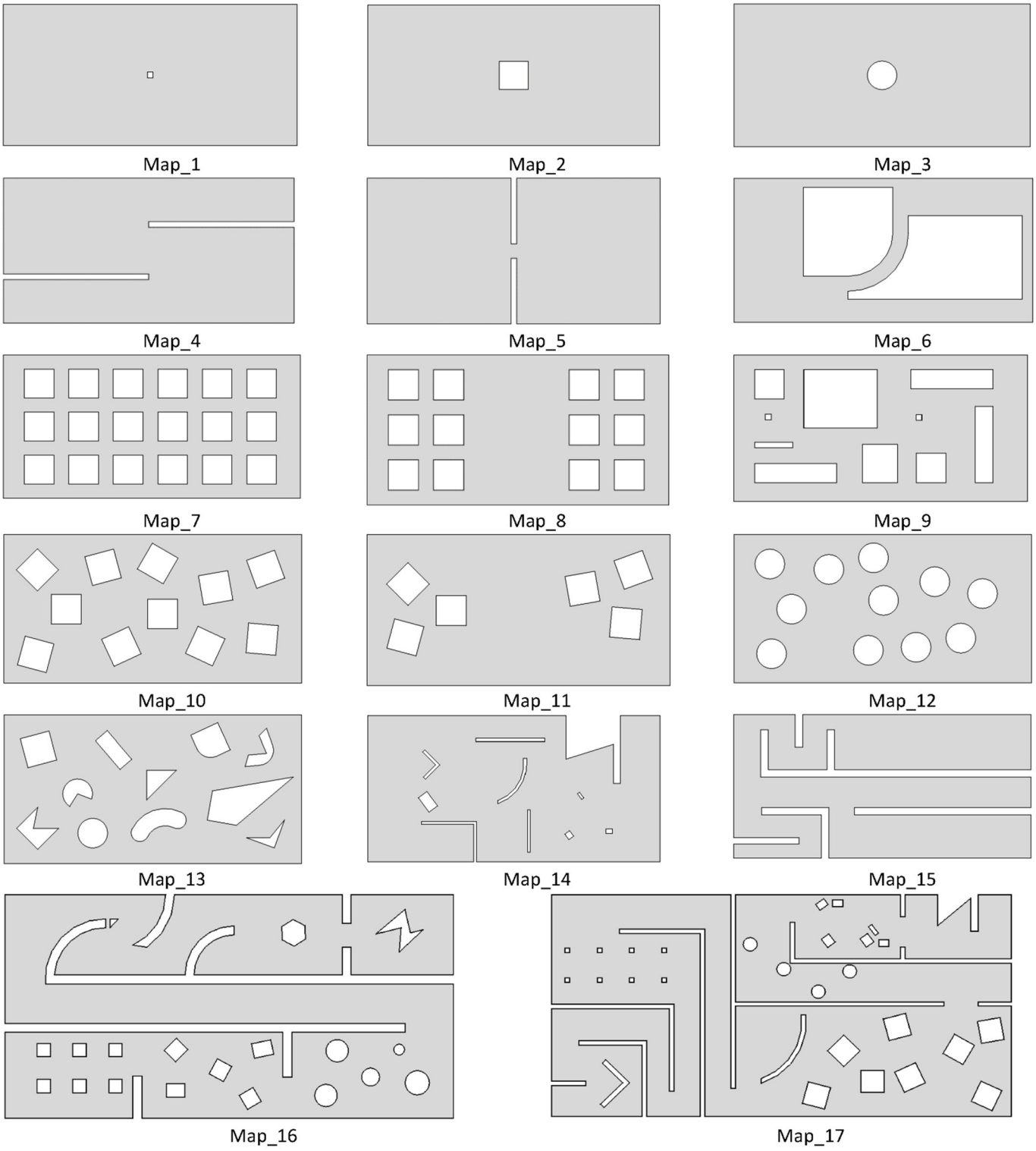


Figure A1. All defined environments.

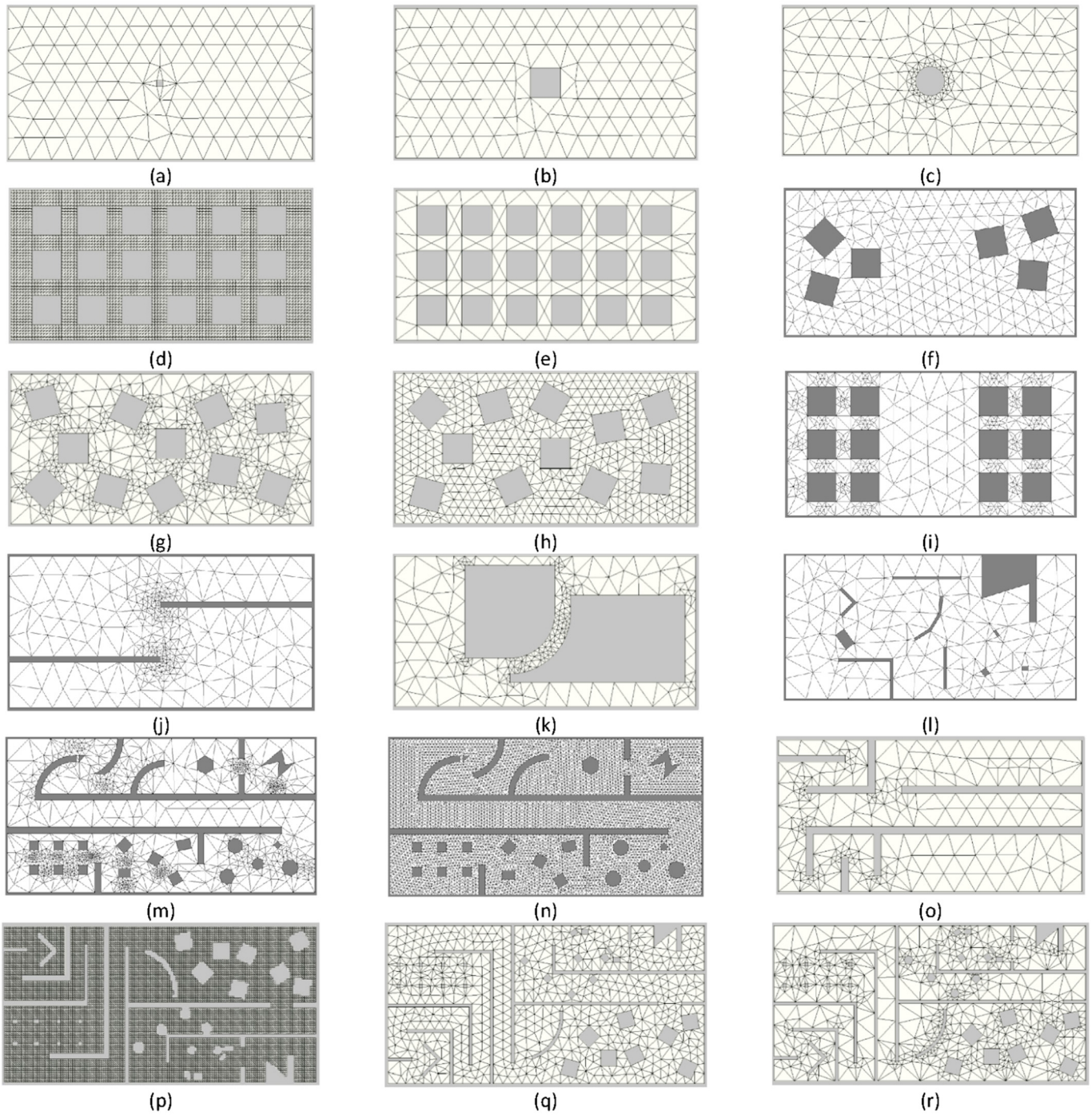


Figure A2. Various environments with different representation types and sizes of QUAD, ITM and VITM. (**a**) Map_1-ITM-700, (**b**) Map_2-ITM-100, (**c**) Map_3-VITM-100-700, (**d**) Map_7-GRID 200 × 100 (**e**) Map_7-ITM-300, (**f**) Map_11-ITM-500, (**g**) Map_10-VITM-100-700, (**h**) Map_10-ITM-300, (**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).

2.6 Conclusion:

The traditional mapping methods QUAD and TRI face limitations in robotics—QUAD fails in computationally intensive situations and TRI has safety issues near obstacles. This study emphasizing the need for selecting the appropriate representation to achieve efficient path planning. It confirms that each representation produces different paths based on obstacle distribution, addressing the question of which method performs better in different environments. Additionally, it demonstrates the critical role of robot size, introducing an inflation layer as a practical and effective approach for incorporating this factor compared to traditional methods that use obstacle offsets. Furthermore, this work establishes metrics for bench-marking the performance of the two representations, providing a systematic means to evaluate their impact on path planning.

Overall, the study after the confirmation of the hypothesis that the efficiency of each representation varies based on the environment, prepares the initial groundwork for constructing a hybrid representation map by integrating inflation layers and performance metrics for a practical platform.

Chapter 3. A Hybrid Representation Map

3.1 Introduction

In the previous chapter, the different behaviors of representations in path planning across various environments were shown. Now, as the core work of this thesis, a hypothesis is proposed that hybrid representations could provide a better solution for path planning in large maps. This is particularly critical for cluttered environments, as identified through the literature review and noted gaps. Therefore, the key to implementing a hybrid representation for large maps is determining where each type of representation should be applied.

However, for extensive maps, manually defining these areas is not feasible. Consequently, there emerged a need to establish a metric to analyze the environment, based on which the best representation could be determined. Nevertheless, as highlighted in the literature review, no universally agreed-upon metric exists among researchers. To address this, this work attempts a non-metric approach to relate environment types directly to the most suitable representation for path planning. Additionally, to ensure the practicality of the proposed mapping method, it was essential to validate the results under real-world conditions. Furthermore, hybrid representations must be clearly defined in transitional areas between regions with different characteristics.

This chapter directly addresses sub-objective 2 of the thesis, which is the development and evaluation of a Hybrid Representation Map for efficient global path planning in large and cluttered indoor environments.

3.2 Article 2: An efficient indoor large map global path planning for robot navigation

Journal: Expert Systems With Applications (ELSEVIER) (https://www.www.elsevier.com/locate/eswa)

Authors: Ahmadreza Meysami, Sousso Kelouwani, Jean-Christophe Cuilliere,
Vincent Francois, Ali Amamou, Bilel Allani

Accepted date: 30 January 2024

3.3 Methodology

Checking the presented hypothesis about the effectiveness of a hybrid representation map requires several stages.

The proposed method involves dividing a large environment into regular square-shaped tiles. For each tile, the best representation—either non-uniform triangle cell-based (TRI) or regular quadrangle cell-based (QUAD)—is selected based on predefined criteria for efficient path planning. The path search is then conducted using the A* algorithm. This approach eliminates the need for an additional stage to define the environment's difficulty, unlike other methods, by utilizing a trained model that predicts the best representation based on the environment's image. To accomplish this, a Convolutional Neural Network (CNN) has been trained using labeled environments.

The CNN used in this thesis is a lightweight 15-layer model adapted from a standard MATLAB small-image classification example (Figure 30). It processes 28×28 grayscale tiles through three repeated feature-extraction blocks, each consisting of a convolution layer (which learns local patterns using small filters), batch normalization (which stabilizes activations), a ReLU activation (which introduces nonlinearity), and max-pooling (which reduces spatial size while keeping the strongest features). The number of filters increases from 8 to 16 to 32 across the blocks, allowing the network to learn progressively more complex features. After feature extraction, a fully connected layer maps the learned representations to two output classes (TRI and QUAD), followed by a softmax layer that converts scores into probabilities and a classification layer using cross-entropy loss. This CNN serves only as a simple, well-tested preprocessing tool for tile labelling.

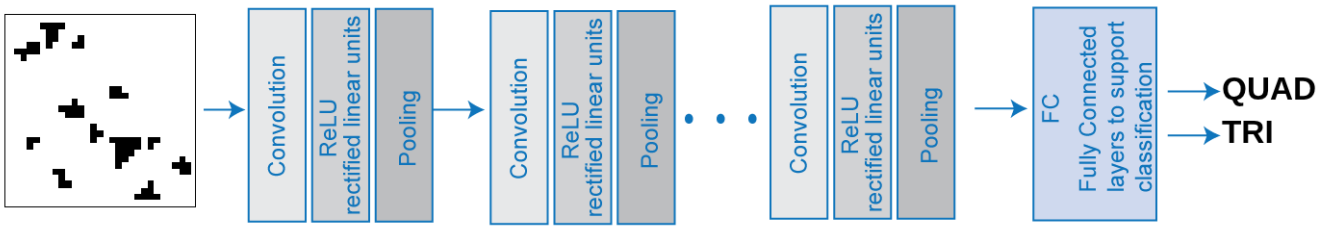


Figure 30: Simplified architecture of the convolutional neural network (CNN) used for tile classification. Adapted from (MathWorks, 2022)

The training stage serves as a generative method for producing various environments and includes an algorithm for labeling each environment. The methodology does not apply smoothing to the paths generated by A* to compare the plain behavior of each representation. The final step involves

calculating key parameters—path length, complexity, and minimum distance to obstacles—to evaluate the performance of the hybrid map in path planning.

The hybrid representation map can take different shapes based on the connection between tile representations due to the variability in mesh generation. Two methods can be used: $M_{HYB\text{-}C}$, where each tile is meshed following the mesh elements of its neighboring tiles, resulting in a conformal mesh throughout the environment, and $M_{HYB\text{-}NC}$, where each tile is independently meshed, leading to separate non-conformal mesh tiles. Mesh conformity is important in computational modeling, affecting the accuracy and complexity of the mesh generation and altering the behavior of the search algorithm due to changes in the definition of neighboring tiles. This study explores the impact of mesh conformity on the performance of global path planning, highlighting the differences between $M_{HYB\text{-}C}$ and $M_{HYB\text{-}NC}$ in generating integrated or distinct maps, respectively.

Path planning with A* varies depending on the type of hybrid representation map. In $M_{HYB\text{-}C}$, A* behaves similarly to traditional path planning on a single representation map, but with neighbor finding that accommodates both triangle and square cell-based elements. For $M_{HYB\text{-}NC}$, additional stages are needed because of the non-conformal mesh, which prevents direct neighbor finding between tiles. This study introduces M'_{QUAD}, a rough grid, to bridge this gap by identifying intersections and providing local start and goal points within each tile. A* is then applied independently to each tile using the optimal representation map, and the paths are assembled into a final global path.

The efficiency of path planning can vary depending on the project. Three criteria—path length, distance to obstacles, and path search complexity—allow for quantification through a unified metric. Additionally, Cellular Automata (CA) helps generate realistic environments for training a CNN model. This model learns from labeled images to predict the best representation for path planning. By comparing paths generated in environments using both M_{TRI} and M_{QUAD}, a decision can be made on the optimal representation for each environment, thus enhancing the overall path planning process.

The experiment aimed to demonstrate the practicability of the proposed approach in real environments by validating the results of performed simulations. Three environments similar to the GAZEBO simulation environments were implemented, each being a 10 m × 5 m area reflecting the lab's size constraints and using a combination of triangle and square maps as the best representation. A Turtlebot 2 robot was used with a 0.2 m inflation layer converted into a global path generation cost map. The real experiment utilized a SLAM Gmapping-generated map with intrinsic errors and uncertainties, while the simulation used a ground-truth map. The goal was to compare the correlation between the global path and the traveled distance. A total of 60 tests were conducted across 5 scenarios per environment, each with four different representations. The study reported values for traveled distance and traversal time,

and a comparison was made between the absolute traveled distance in simulations and experiments to validate accuracy.

3.4 Summary of analysis

For training, various environments were created as large maps, with the optimal tile representations (triangle or quadrangle) determined using a trained model. These environments featured diverse obstacle shapes and distributions, requiring different optimal representations across the map. A large map could range from simple configurations, such as two tiles with one optimal as TRI and the other as QUAD, to complex configurations with 100 tiles. Scenarios involved paths traversing at least two representation types, evaluated based on path length, distance to obstacles, and path search complexity. No smoothing was applied to paths.

Results showed that HYB-NC (Hybrid Non-Conformal) representation minimized path search complexity and demonstrated a low slope in the scatter plot of path length vs. path search complexity. HYB-C (Hybrid Conformal) excelled in generating the shortest paths, solving the zig-zag problem of QUAD representation by selectively using TRI representation. Compared to single representations, HYB-C provided the shortest paths in 50% of scenarios, while HYB-NC reduced path search complexity in 60% of scenarios.

Scalability was tested using the Lmap10_10 environment, consisting of 100 tiles generated using the CA method. HYB-NC significantly reduced path search complexity while maintaining a path length within 10% of the shortest path. These results emphasized the importance of path search complexity over path length in large environments.

Correlation analysis between planned path length and actual traveled distance revealed a strong correlation (0.87) in empty areas but reduced correlation (0.67) in cluttered environments due to the influence of the local planner. Hybrid representations proved practical for safe and efficient navigation, with 70% of scenarios aligning optimal planned length with traveled distance.

Real-world experiments validated the approach in three 10 m × 5 m environments using a Turtlebot 2 and SLAM-generated maps. A total of 60 tests measured traveled distance and traversal time, comparing simulation and real-world results. HYB-C achieved the best average values for traveled distance and traversal time, while TRI had the highest failure rate. QUAD demonstrated fewer failures but longer paths. Discrepancies between simulations and real tests were attributed to environmental factors like terrain slope, dust, and obstacle uncertainties.

3.5 Article

Expert Systems With Applications 248 (2024) 123388

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

An efficient indoor large map global path planning for robot navigation

Ahmadreza Meysami [a,b,*], Sousso Kelouwani [a,b], Jean-Christophe Cuilliere [a,c], Vincent Francois [a,c], Ali Amamou [b], Bilel Allani [b]

[a] Department of Mechanical Engineering, UQTR, des Forges, Trois-Rivieres, G8Z 4M3 Quebec, Canada
[b] Hydrogen Research Institute, Pavilion Tapan-K.-Bose, 3351 Bd des Forges, Trois-Rivi`eres, Quebec G9A 5H7, Canada
[c] ERRICA, UQTR, 3351 Bd des Forges, Trois-Rivi`eres, Quebec G9A 5H7, Canada

ARTICLE INFO

Keywords:
Robotics mapping
Hybrid representations
Convolutional neural network
Mesh conformity

Large indoor cluttered environment representation is still a challenging task when non-uniform triangle cell- based or quadrangle cell-based decomposition is used to build the map. This paper aims at proposing a new method to represent a large indoor environment for efficient and global robot path planning using a trade-off among three criteria: path length, distance to obstacles, and path search complexity. In this regard, three steps are involved: (i) the design of a tiled map that represents several regular sub-regions of the given large environment; (ii) the selection of the best representation between non-uniform triangle cell-based submap and regular quadrangle cell-based submap for each tile using the predefined efficient path planning criteria. Hence, the entire large environment is represented by a hybrid cell-based map; (iii) the path search using the well- known A* algorithm. Moreover, to find the best representation, firstly, a generative method based on cellular automata is used to build a large synthetic database of maps of the same size as the tiles. Each map of the database is associated with the representation which provides the most efficient path planning. Given a tile, the corresponding image is used to find the closed tile image from the database, and the associated representation is selected as the best representation. Extensive simulations, as well as experiments, suggest that for a given large, cluttered environment, the hybrid representation with a mix of triangle-based cells and quadrangles-based cells can provide a more efficient global path compared to the traditional regular quadrangle or triangle representations.

Introduction

Nowadays, robots must operate in large environments with efficient path planning but with limited computational resources. Finding an optimal path may not be practical due to memory limitations when dealing with the increasing search space in a large navigation environment (Santos, Aguiar, Santos, Valente, & Petry, 2020). This problem becomes even more challenging when multiple criteria must be considered for an efficient global path, such as path length, distance to obstacles, and path search complexity[1] in a cluttered environment. This is because path-planning algorithms exhibit varying behaviors in different environments (Xiao et al., 2022). In a large cluttered map, fixed path planning elicits different responses in its various regions. In other words, no single algorithm works efficiently under all conditions. One potential solution involves adjusting parameters (parameter tuning) or altering map representation to better suit a new environment. The first strategy requires training based on the best practices of generated paths and identifying optimal parameters. However, this doesn't guarantee optimal performance in new, unseen large environments (Wang, Zhang, Ma, Li, Ma, Meng, & Meng, 2021). As for the second strategy, it is essential to determine which map representation performs better in path planning for each type of environment.

The study of different map representations for efficient global path planning in various small indoor environments like corridors, ware- houses, and narrow passage areas was presented in Meysami, Cuillie`re, François, and Kelouwani (2022). It was indicated that a non-uniform triangle cell-based or a regular quadrangle cell-based decomposition works well in these scenarios. However, these methods are not

* Corresponding author at: Department of Mechanical Engineering, UQTR, des Forges, Trois-Rivieres, G8Z 4M3 Quebec, Canada.
E-mail addresses: ahmadreza.meysami@uqtr.ca (A. Meysami), sousso.kelouwani@uqtr.ca (S. Kelouwani), jean-christophe.cuilliere@uqtr.ca (J.-C. Cuilliere), vincent.francois@uqtr.ca (V. Francois), ali.amamou@irh.ca (A. Amamou), bilel.allani@uqtr.ca (B. Allani).

[1] Path search complexity can be defined based on the time of finding a path or the number of visited cells by a searching algorithm like A*. In this paper, the second approach (Chagas, Zacarias, de Lima Silva, & de Freitas, 2022) has been selected.

https://doi.org/10.1016/j.eswa.2024.123388
Received 15 September 2023; Received in revised form 8 January 2024; Accepted 30 January 2024
Available online 6 February 2024
0957-4174/© 2024 Elsevier Ltd. All rights reserved.

appropriate for large, cluttered environments due to above mentioned limitations. For example, a regular quadrangle cell-based decomposition (or simply a grid method) using a fixed cell size is not an appropriate choice due to the increasing path search complexity in a large map. A quad-tree map, (Samet, 1984) could be a solution but it presents the implementation and neighbour-finding challenges. Moreover, a non- uniform coarse triangle cell-based decomposition with a limited num- ber of elements may not find a path with a competitive length. This is due to the scarcity of nodes near obstacles, which hinders the ability to achieve optimal turning. While refinement could lead to a shorter path (Aggarwal & Kumar, 2020), it may overlook the advantages of utilizing a low-volume map for rapid path planning, owing to visibility checks required near obstacles with varying shapes and distributions. Thus, it can be inferred that relying solely on a single triangle or quadrangle representation might not provide an efficient solution for path planning in a large, cluttered environment. To a address this challenge, the adoption of a hybrid grid in computational engineering and graphics has emerged as a well-established approach. For instance, Speray and Ken- non (1990) a hybrid grid is defined as the combination of structured and unstructured grids to leverage their computational strengths and their ability to fit various shapes. The implementation of this hybrid approach, for effective adaptation to new regions within a large environment, requires the definition of an environment metric. This metric needs to be monitored to select the optimal representation. Human intervention is not feasible for this task in a large environment, and the robot must adapt itself to accomplish it.

In this work, the concept of a hybrid representation cell-based map is investigated as a promising solution for efficient global path planning in large, cluttered environments. In this regard, a novel method is proposed for representing such environments using a hybrid map for finding an efficient global path and an algorithm for selecting the best representation in each tile or sub-map based on predefined efficient path planning criteria. The process of finding a performance metric for the environment has been also simplified by using a learning model to establish a direct connection between the obstacle distribution in the environment and the efficient global path like using CNN in holistic descriptors and domain adaption (Cebollada, Pay´a, Flores, Peidro´, & Reinoso, 2021). Furthermore, the effect of conformity between the representation tile neighbors, which clarifies the fundamental difference between conformal and non-conformal maps is investigated. The proposed approach aims to improve the performance of existing methods for global path planning in large, cluttered environments by leveraging the strengths of both non-uniform triangle and regular quadrangle cell-based decompositions. The effectiveness of the suggested approach is demonstrated in efficient path planning for a small robot in different cluttered environments. The results show a significant reduction in path search complexity with acceptable path length and safety. Furthermore, the effect of local planning is considered, and the practicality of the proposed method is examined by leveraging a known robotics infrastructure in parallel. Considering the above-discussed matters, the main contributions of this manuscript are as follows:

Unlike other similar methods, this study utilizes a hybrid representation map instead of a hybrid map by combining two successful representations used in the robotics and games industry: namely, a uniform quadrangle (or grid method) and a non-uniform triangle.

This study takes into consideration the influence of different environments on the performance of path planning.

In contrast to existing methods, the suggested approach eliminates the need for defining complex criteria for an environment and directly connects each sub-region (tile) to the most suitable environment representation. This enables the association between the optimal representation for more efficient path planning and the spatial distribution of environmental obstacles. As a result, this approach equips the robot with the capability to find an efficient global path when navigating a new, cluttered environment.

The organization of the remaining sections of this article is as follows. Section 2 describes the proposed approach for generating a hybrid tiled map, including a generative method to train a model for predicting the best representation in each sub-map. Section 3 presents the implementation, simulation, and physical experiments conducted to validate the developed approach. In Section 4, the benefits of the pro- posed method are elaborated upon, and it is compared with similar works that share common objectives. Finally, Section 5 provides a conclusion and discusses future directions for research.

Related works

As mentioned, two primary challenges in path planning for large environments are computational and diversity issues. The primary focus in large-scale navigation work is addressing the computational resource limitation (Ravankar, Ravankar, Emaru, & Kobayashi, 2017), neglecting the efficiency of path planning in diverse environment regions as the second problem. To address computational issues, strategies include improving common algorithms and hybridization. Examples of improved algorithms include JPS(A*) (Duchonˇ et al., 2014), which employs cropping for neighbor selection. Additionally, it introduces various enhancements to A*, dependent on the environment type. Quick-RRT* (Jeong, Lee, & Kim, 2019) utilizes a user-defined depth parameter to leverage the ancestry concept over RRT*, demonstrating distinct performance in maze and clutter environments. KB-RRT* (Wang, Li, & Meng, 2021), which incorporates an additional node rejection condition, exhibits varied path length competency in maze or clutter environments, as indicated by the detailed results. There's competition between various graph-based and sampling-based path planning methods, each aiming to outperform the other. For instance, A* claims superior results compared to RRT, as noted by Zammit and Van Kampen (2022), while in different studies, RRT demonstrates better responsiveness than A* (Xin et al., 2023). However, it's essential to acknowledge that each algorithm's performance varies across different environment types. As highlighted by (Ding et al., 2023), RRT* faces challenges in corridor scenarios, prompting efforts to enhance its per formance. Similarly, Ammar, Bennaceur, Chaˆari, Koubaˆa, and Alajlan (2016) identified problematic behavior of A* in maze environments and proposed a relaxed A* as a solution.

Hybridization, as a tool for addressing these challenges, involves using hybrid path planning algorithms (Zhong, Tian, Hu, & Peng, 2020) or a hybrid map, incorporating two or more different types of map layers to capitalize on their respective advantages for efficient path planning. The use of hybrid maps to address the limitations of computational re- sources or efficient path planning in large environments has also been researched but not as much as grid methods (Liu et al., 2023). The hybrid metric map representation (HYMM) proposed by Nieto, Guivant, and Nebot (2004) and Guivant, Nebot, Nieto, and Masson (2004), for example, combines feature maps and grid maps to address the issue of sparse representation due to computational limitations. HYMM partitions the map into triangular subregions in a global frame, leading to a dense occupancy grid representation. Similarly, Nitsche, de Cristo´foris, Kulich, and Koˇsnar (2011) uses a combination of metric and topological maps, defining two types of nodes, area nodes and gateway nodes, to establish connections between the two maps for hierarchical planning. Liu et al. (2023) divides the map into sub-maps using a combination of metric and feature maps to improve mapping efficiency. Finally, Santos, Santos, Valente, and Costa (2022) proposes a hybrid map approach that reduces the path search space (complexity) by saving a predefined path in local metric maps and conducting a global search in the topological map. However, these hybrid perspectives have overlooked the diversity present in large environments.

Detection of this diversity can be achieved by defining a suitable metric to assess the performance of the path planning in a given environment. By doing so, the path planning parameters or approaches can be adjusted to match the variations in this metric. Previous studies have attempted to define environment difficulty and its effect on path

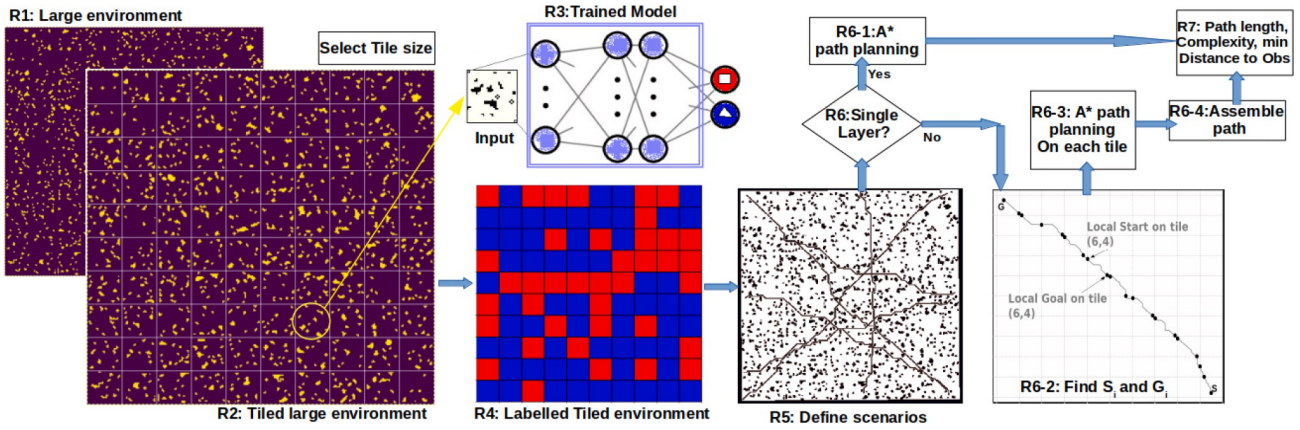


Fig. 1. The approach pipeline for runtime section initiates at R1 and concludes at R7. R1 corresponds to the specific large environment, R2 depicts the tiled environment, R3 involves a trained model predicting the optimal representation map for each tile based on its image. In R4, the symbolic hybrid representation map utilizes red for tiles with uniform quadrangles and blue for those with irregular triangles. Proceeding to R5, scenarios are defined by specifying start and goal points within the large environment. Checkpoint R6 acts as a decision point for employing distinct path planning strategies in single and double-layer maps. R6_1 involves employing A* with hybrid neighboring cells. In contrast, R6_2 and R6_3 pertain to path planning on individual tiles with local start and goal points derived from a rough grid. R6_4 entails the assembly of generated paths within each tile. Concluding the pipeline, R7 calculates essential properties of the generated path to facilitate performance comparison among each representation in path planning. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

planning based on a variety of approaches. For instance, Tsardoulias, Iliakopoulou, Kargakos, and Petrou (2016) used distance transformation on a grid map to define the environment's difficulty, while Ran˜o´ and Minguez (2006) considered obstacle shape and distribution as parame- ters for comparing different environments. In Perille, Truong, Xiao, and Stone (2020), it has been attempted to link environmental difficulty to traversal time and it has combined various environmental metrics in navigation benchmarking. Meanwhile, Anderson and Yang (2007) and Held, Lampe, and Chatila (2006) proposed using entropy calculated on a grid representation of the environment to define a metric of difficulty. More recently, some researchers have utilized convolutional neural networks (CNN) to estimate their defined metric based on the same parameters (Sartori, Ermacora, Pei, Zou, & Yu, 2020). Each paper de- fines its criteria for the performance metric, and there is no consensus on a unified and well-accepted approach. Additionally, according to Wang, Zhang, Ma, Li, Ma, Meng, and Meng (2021), most hybrid methods that combine a learning approach and classical path planning struggle to handle large environments.

The presented work, unlike others, doesn't just insist on any specific path planning algorithm. It attempts to solve two issues: computational and diversity problems, employing an interconnected approach. Its core idea focuses on adapting map representations for each region to enhance path planning efficiency, addressing path search complexity. Although the predominant representation is the grid method, it also considers the benefits of recently used polygonal mesh representation types, like tri- angles, to illustrate this hybrid approach. These representations can even be applied to sampling-based methods like RRT (Mthabela, Withey, & Kuchwa-Dube, 2021).

Approach

The proposed method suggests dividing a given large environment into regular square-shaped sub-regions, referred to as tiles. The best representation is selected for each tile between non-uniform triangle cell-based and regular quadrangle cell-based, using predefined criteria for efficient path planning. The path search is then conducted using the well-known A* algorithm, based on the conformity condition between the tiles. With this approach, the need for an additional stage to define the environment's difficulty is eliminated, and instead, a model is trained based on the environment's image to predict the best representation for efficient path planning. In the remainder of this sec- tion, the detail of the approach's pipeline is explained before providing a problem definition to clarify the concept of a tiled map and explaining how conformity generates different types of hybrid representation maps with their structures. The approaches in path planning for different types of hybrid representation maps are then outlined. Finally, the training stage is discussed in terms of a generative method for producing various environments and an algorithm for labeling each environment. It is worth noting that, similar to previous works, no smoothing method has been applied to the path generated by A* to compare the plain behavior of each representation.

For a given large environment, as illustrated in Fig. 1-R1, and considering a square tile size determined by specific parameters detailed in the subsequent section, a virtual tiled environment is derived, as depicted in Fig. 1-R2. Employing a trained model elaborated in Section 2-4, it becomes feasible to predict the optimal representation—whether uniform quadrangles or irregular triangles —based on the tile's image (Fig. 1-R3). This methodology yields a hybrid representation map for the large environment. Here, red tiles denote uniform quadrangles deemed superior for path planning performance, while blue tiles signify envi- ronments with irregular triangles showcasing enhanced performance compared to uniform quadrangles. With this hybrid representation map in place, scenarios (Fig. 1-R5) can be defined, specifying start and goal points on the global frame of the large environment. The subsequent approach employs a distinct methodology for identifying the optimal path based on the hybrid map's structure (Fig. 1-R6). This structure relies on the assumption of conformity between neighboring tiles, elucidated in Section 2-2. Following the single-layer hybrid map (Fig. 1- R6_1), considering hybrid cell neighbors in the A* algorithm (common edge and common node cells simultaneously defined for triangles and quadrangles), the subsequent step (Fig. 1-R7) computes three parame- ters chosen to showcase the representation type's performance in path planning. An effective representation should yield a path that optimally balances the three parameters: path length, complexity, and minimum distance to obstacles. If the outcome of step R6 is negative, indicating the presence of multiple layers in a hybrid map, steps in Fig. 1-R6_2 through R6_4 are employed to determine the optimal path. The detailed methodology for these steps is outlined in Section 2-3. Finally, step R7, as mentioned earlier, calculates the parameters for the generated path, facilitating comparison with other representations.

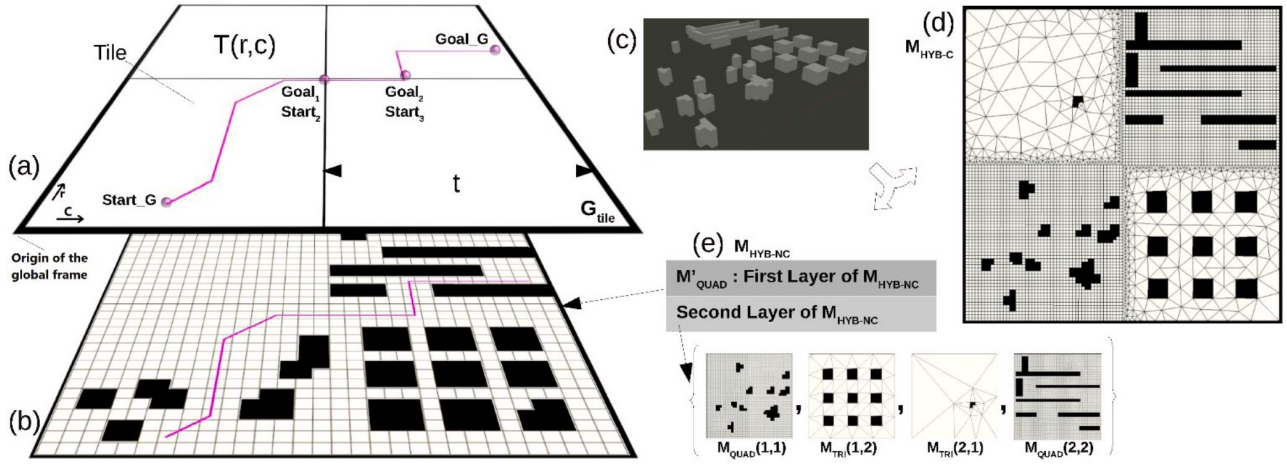


Fig. 2. Definition and structure of two types hybrid representation maps, MHYB-C and MHYB-NC. (a) Definition of grid tile and the methodology to find start and goal points in each tile based on the projected rough path (b) Coarse grid representation map as the first layer of MHYB-NC to find a rough path – denoted by M'QUAD (c) A model of the environment (d) An integrated single-layer MHYB-C (e) Structure of two-layer of MHYB-NC, first layer M'QUAD and second layer the set of individual tile map with the best representation for that tile.

1.1. Problem definition

Consider E as a large and cluttered environment with a size of W*H (width and height) that the robot navigates through. E is primarily represented as a large binary image in which the obstacles (occupied space) are depicted in black, while the free space (non-occupied space) is depicted in white. This image is assumed to be generated by well-known point-cloud-based sensors for mapping. Given any starting point within the free space, and another point within the free space (goal point), the objective is to find an optimized feasible path, if one exists, that can connect the starting and goal points. An optimized path is defined as one that is: (i) collision-free, (ii) the shortest distance between the start and goal points, and (iii) computationally efficient, as determined by algo- rithmic complexity calculations.

The cluttered nature of E indicates that finding a single meshing method capable of providing the optimized path would be highly chal- lenging. To address this challenge, it is assumed that E can be segmented into N squared tiles, each with an edge length of 't'. Therefore, the tiled image of E can be approximated by a matrix of (N_{row} × N_{col}) tiles where N_{row} and N_{col} are the results of rounding up the integer division of H by t, and the the integer division of W by t, respectively. The selection of the tile size depends on various parameters such as the size and spatial distribution of obstacles, the geometric area of the environment, and the robot's specifications. For simplicity, conditions for an environment with rounded values of N_{row} and N_{col} are being considered. However, in cases where there are regions with remaining fractions of tile size, representation types based on the maximum number of preferred rep- resentations among neighboring tiles can be explored. In these cases, N_{row} and N_{col} are not integer values; however, they can be managed similarly to other tiles with a smaller size. Fig. 2 shows parts' definitions and the structure of two types of hybrid representation maps, a single layer M_{HYB-C} and the double layer M_{HYB-NC}. The utilize approach in this study generates these two types of maps based on the conformity concept (explained in Section 2-2). According to the Fig. 2-a, the virtual grid G_{tile} is created with N_{row} * N_{col} tiles, with each tile identified by its specific coordinates in a global frame. This frame originates from the lower-left corner of the environment, where the r-axis and c-axis correspond to the directions of H and W, respectively. In addition, G_{tile} is used to locate the starting and goal points within each tile corresponding to the rough path generated by an M'_{QUAD} with a coarse grid (Fig. 2-b). Each tile T contains both obstacles and free space, with the mesh of free space within T represented by the variable $M_{REP}(r,c)$. Here, "REP" denotes the best representation or element type used for meshing, and r and c represent the coordinates of the tile T in the G_{tile}. For an envi- ronment like Fig. 2-c, the hybrid representation map will be a tiled map in which each tile is generated by two types of elements: non-uniform triangles and regular quadrangles. However, only one type of element (either triangles or quadrangles) is used for mesh generation in each tile. The remaining question of "which element type should be used for each tile?" will be answered in the following sections. Once the element type for each tile is determined, E can be meshed tile by tile, now it is reasonable to assume that the overall map for E will be a finite set of Ms as follows:

$$M_{HYB} = \{M_{REP}(1,1), M_{REP}(1,2), \ldots, M_{REP}(N,M) \mid REP = QUAD \text{ or } TRI\} \quad (1)$$

The subscript HYB indicates that it is a hybrid representation map, which is distinct from other single representations, such as M_{QUAD} and M_{TRI}.

As previously stated, two hybrid representation maps are employed.

In Fig. 2-d, a hybrid conformal map (HYB-C) is depicted, where neighboring tiles are integrated into a single map. Conversely, in Fig. 2-e, a set of individual tile maps or a hybrid non-conformal map (HYB-NC) is displayed.

However, M_{HYB-NC}, which is a finite set of individual tiles generated independently, does not constitute an integrated map. As the end user of the map is the global planner, the map's structure should be designed to allow for the possibility of moving between tiles to find a path. While integrity in an M_{HYB-C} is sufficient for moving between elements in neighbour tiles, non-conformity between tiles in an M_{HYB-NC} is insuffi- cient for moving between neighbour elements in two different tiles. To address this issue, an extra layer is added to the set M_{HYB} in Eq. (1), which is a coarse grid map of E (see Fig. 2-b). As a result, an M_{HYB-NC} will be a two-layer map, with the first layer being a coarse grid map of E (shown by M'_{QUAD}) and the second layer being a set of individual meshed tiles (see Fig. 2-e) similar to search in hierarchical representa- tion like (Brondani, de Lima Silva, Zacarias, & de Freitas, 2019). Although the addition of this extra layer increases the overall path search complexity, the increase is small, as will be explained in the next section. Thus, the definition of M_{HYB-NC} is as follows:

$$M_{HYB-NC} = \{M_{REP}(1,1), M_{REP}(1,2), \ldots, M_{REP}(N,M) \mid REP = \text{QUAD or TRI}\} + M'_{QUAD} \quad (2)$$

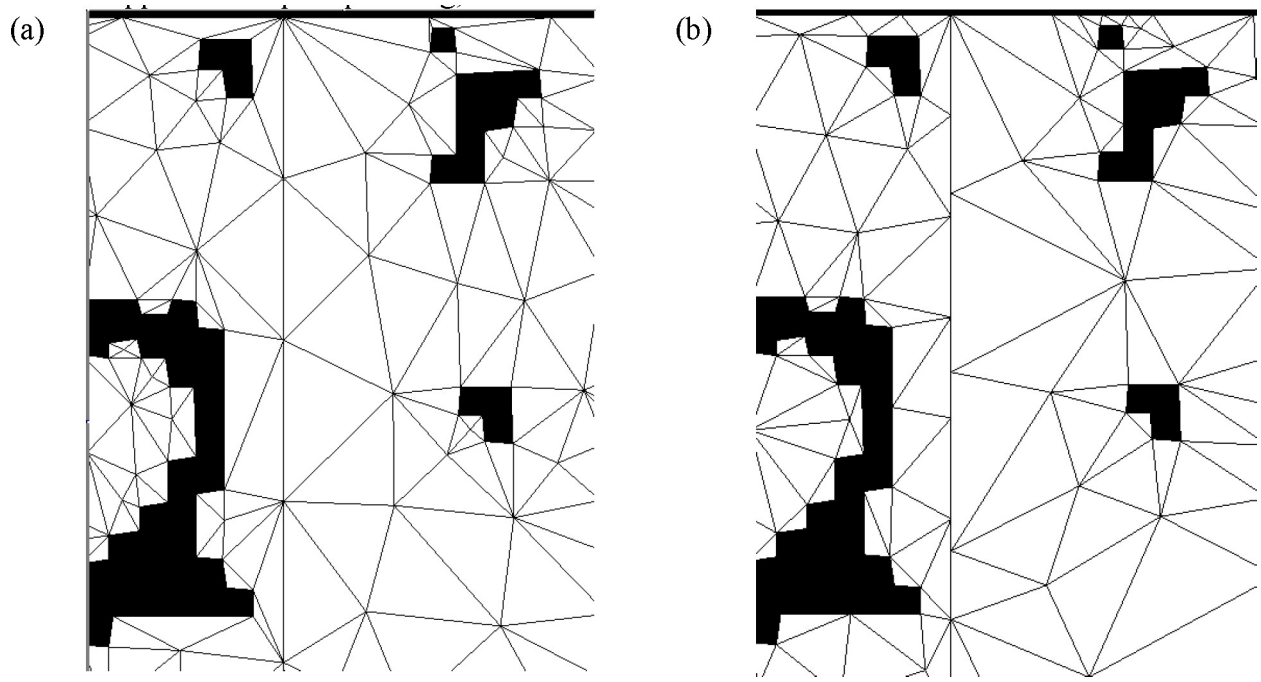


Fig. 3. Conformity as important property in mesh generation difficulty and neighbour definition for moving between elements (a) Conformal mesh is when the intersection of two common-edge elements is a complete edge not only a section of it. (b) Non-conformal mesh.

As mentioned earlier, M'_{QUAD} is a coarse grid map of E.

1.2. Conformity

Mesh generation in a hybrid representation map (M_{HYB}) can be performed in two ways. The first method is by meshing each tile by following the mesh elements of its neighbouring tiles, which results in a conformal mesh throughout the environment. This type of mesh is denoted as $M_{HYB\text{-}C}$ and is illustrated in Fig. 3-a. The second method is by independently meshing each tile, which results in a set of separated meshed tiles, denoted as $M_{HYB\text{-}NC}$ as a hybrid non-conformal represen- tation map. This type of mesh is similar to a non-conformal mesh and is illustrated in Fig. 3-b. Mesh conformity is an important property in computational modelling like finite element method to get an accurate and reliable result. Conformity can influence the complexity of mesh generation when utilizing a polygonal mesh for a robotics map. Furthermore, it modifies the behavior of the search algorithm due to the alteration in the definition of neighboring. As a novel feature of this work, the impact of conformity between meshed neighboring tiles on the performance of global path planning is explored.

Conformity between tiles in an $M_{HYB\text{-}C}$, which is generated during mesh generation, leads to the creation of an integrated single-layer map, and the grid G_{tile} is exclusively employed for hybrid representation generation. Once the map is generated, the concept of tiles is no longer applicable. However, non-conformity prevents $M_{HYB\text{-}NC}$ from becoming an integrated map, and this results in a distinct approach for path planning, as will be discussed in the next section.

1.3. Path planning

As mentioned earlier, the proposed approach attempts to partition a large environment into tiles. The map with optimal representation in each tile, identifiable with a trained model, is a hybrid representation map with two different structures based on whether conformity between tiles is considered. The first structure is a single layer with integrated tiles, referred to as $M_{HYB\text{-}C}$, while the second structure involves two layers with independent tiles, known as $M_{HYB\text{-}NC}$. This hybrid structure is compatible with most path planning algorithms such as Dijkstra, A*, and D*. However, the A* algorithm is utilized in this study for path planning due to its popularity and compatibility with ROS that makes the experimental study more convenient. Path planning using the A* method on a hybrid representation map differs depending on the type of map. In the case of $M_{HYB\text{-}C}$, it is similar to traditional A* on a single representation map, except that neighbour finding between elements works for both triangle and square cell-based elements. However, A* in an $M_{HYB\text{-}NC}$ requires additional stages to manage the impossibility of neighbour finding in a non-conformal mesh between tiles. Since conformity only exists within each tile, path planning by a traditional A* in each tile is possible when the start and goal within that tile are known. To find these starting and ending points within each tile, the extra layer, M'_{QUAD}, has been added to this study. First, a rough path is identified using M'_{QUAD} with the global start and global goal points. This path is then projected onto the G_{tile} to identify their intersections. Each consecutive intersection within each tile serves as the starting and ending points for that tile, and A* can be applied independently to each tile. The final path is constructed by assembling each local path within a global framework. The algorithm for global path planning on an $M_{HYB\text{-}NC}$ is outlined in Algorithm-1.

Algorithm 1: Path planning on an $M_{HYB\text{-}NC}$

1: Inputs: M'_{QUAD}, Start_G, Goal_G
2: Path = FindPath(M'_{QUAD}, Start_G, Goal_G)
3: $Start_0$ = Start_G
4: [$Goal_i$, r_i, _]=Intersection(Path,G_{tile})
5: For i = 1: Number of Intersected Tile do
6: Path_i = FindPath($M_{REP}(r_i,c_i)$, $Start_i$, $Goal_i$)
7: $Start_{i+1}$ = $Goal_i$
8: end for
9: Final_Path = Assemble(Path_i in global coordinates)

M'_{QUAD} represents the coarse grid of a large environment and is included in the input, similar to the inclusion of start and goal points within the global frame. The FindPath function employs a standard A* path plan- ning approach, yielding a Path as its output. $Start_i$ and $Goal_i$ denote the start and goal points within each tile, utilizing local coordinates. The Intersection function takes two inputs: the Path generated by the Find- Path function and G_{tile} – the tiled grid. This function determines the



Fig. 4. Some generated random, regular and irregular environments.

intersection of the Path with the axes of the tiled grid, defining $Goal_i$. Meanwhile, r_i and c_i represent the coordinates of the corresponding tile. Path_i signifies the output of A* path planning. It employs the optimal representation map for the respective tile, utilizing the start and goal points generated through the Intersection function. The Final_Path is calculated using the Assemble function. This function operates on the paths generated within each tile using local coordinates, integrating
them into a unified path within the global frame.

1.4. Training

To address the earlier question, 'Which element type is most suitable for each tile based on path planning efficiency?' it is essential to first quantify the efficiency. The efficiency of path planning can vary depending on the project. As mentioned earlier, three criteria for measuring efficiency have been identified: path length, distance to ob- stacles, and path search complexity. This efficiency can be quantified using a metric represented by a cost value calculated through a cost function, as follows:

$$f_{REP} = w1 \times NL_{REP} + w2 \times NC_{REP} + w3 \times ND_{REP} \quad (3)$$

REP stands for the representations between TRI and QUAD. The term f_{REP} denotes the specific cost value for a given representation. The weights w1, w2, and w3 are defined in accordance with the importance of their respective parameters; for simplicity, they are considered to be equal. NL_{REP} represents the normalized length, NC_{REP} signifies the normalized complexity, and ND_{REP} represents the minimum distance to obstacles.

With this comprehensive definition, the question, can be effectively addressed. To tackle this query, the utilization of an element type that falls between a non-uniform triangle and a regular quadrangle is pro- posed. This approach demonstrates greater efficiency in path planning for the specific tile by selecting either a TRI or QUAD element type that is better suited for it.

In the proposed innovative approach, the raw image of a tile serves as the sole input for predicting the most suitable representation for that specific tile. In the following two sections, the approach (Fig. 5) to generate different environments labeled with triangles or quadrangles as the best representation for predicting the most efficient path planning based on the three criteria is presented.

1.4.1. Generative method

A realistic environment can be modelled as a dynamic system that has defined states and rules for state transitions. Cellular Automata (CA) (Wolfram, 1984) is a useful tool for capturing the evolution of such environments in discrete time with finite states. By leveraging CA, diverse environments can be generated that are more representative of the real world than those generated randomly. To form these environ- ments, several parameters are set, such as the initial fill percentage, filling, clearing, and trimming thresholds of the grid, which are selected based on the environment and robot sizes according to Perille et al. (2020). Robot diameter is an important parameter because it directly affects the search complexity of finding a path among narrow passages. Since structured environments are also within the operational scope of the robot, some regular-shaped random environments were generated to capture atmospheres such as corridors and warehouses. The size of the generated environments is matched to that of the tile size, as the rep- resentation of each tile will be decided upon by the trained model based on a fixed image size, as a routine task in using a fixed input size in a machine learning model. Some irregular and regular random environ- ments are shown in Fig. 4.

Regular patterns often exhibit a consistent geometric shape, and their arrangement according to a specific sequence sets them apart from random or irregular arrangements of elements, which lack

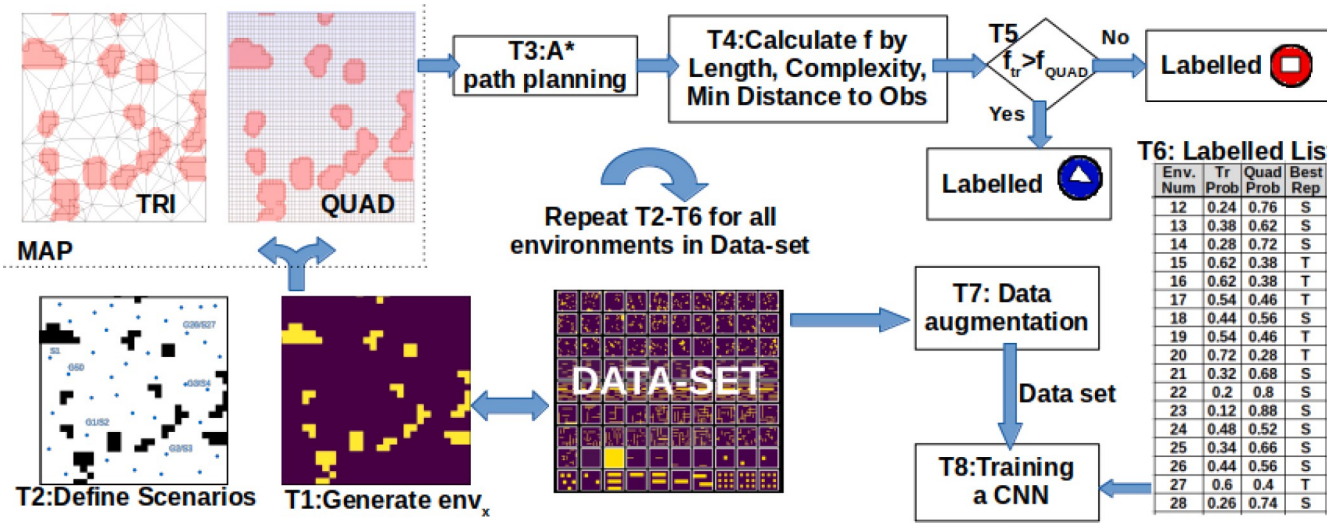


Fig. 5. Approach pipeline for the training section starts from T1 and ends at T8. T1 an image of a sample environment. T2 defining various scenarios in the environment. MAP section shows the inflated obstacles with TRI and QUAD representations. T3 is using the A* path planning with two types of maps and defined scenarios. T4 calculates the cost value for TRI and QUAD for each scenario. T5 is comparison between the f_{tr} and f_{QUAD}. According to this comparison, the envi- ronment will be labelled as QUAD (letter S for square) or TRI (letter T for triangle) and will be logged in list T6. T7 is using data augmentation to increase the number of training samples. T8 is the final step which is using a CNN to train a model with labelled image of environment.

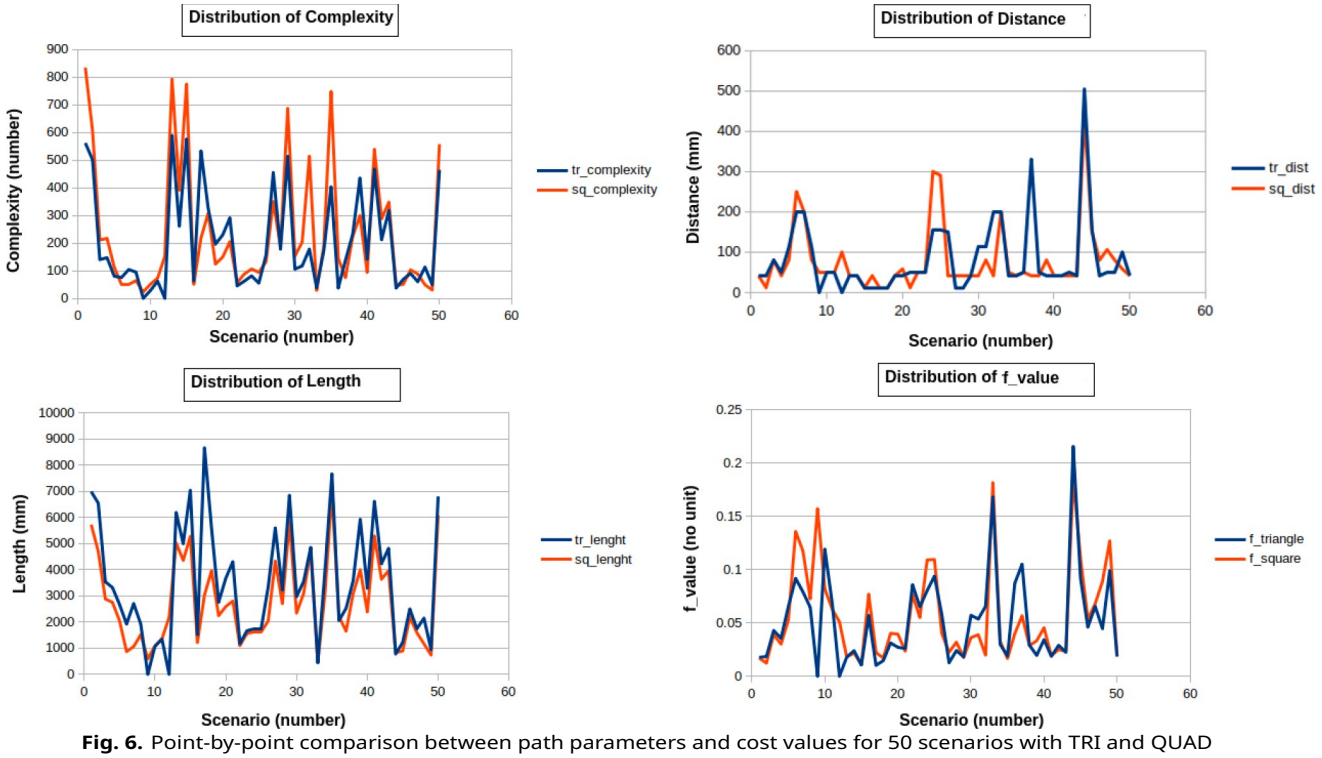


Fig. 6. Point-by-point comparison between path parameters and cost values for 50 scenarios with TRI and QUAD representation maps.

predictability. In certain instances, environments may encompass regular shapes along with a random arrangement. The behavior of repre- sentation within each environment varies in the path planning, especially when navigating around obstacles. To encompass a wide range of scenarios, different types of environments have been considered that span a spectrum of regular, irregular, and hybrid shapes and patterns.

1.4.2. *Labelling the best representation*

Referring to Fig. 1, a trained model employs an image of a specific tile from a tiled environment to determine the optimal representation choice between a uniform quadrangle and an irregular triangle (the former being popular and accurate in robotics, and the latter serving as a successful representation in games, virtual reality and simulation ap- plications, famously known as a navigation mesh). As a result, the trained model requires a substantial number of labeled images to effectively carry out its task. To create a labeled dataset for this purpose, the following approach has been used.

Consider the image I_E of a generated environment using CA or other methods (Fig. 5-T1). Two types of maps, M_{TRI} with non-uniform triangle representation and M_{QUAD} with regular quadrangle representation, are then generated. Using a platform introduced in (Meysami et al., 2022), a navigation scenario can be performed by applying an inflation layer around obstacles to ensure the robot's safety and account for its size (Guimar˜aes, de Oliveira, Fabro, Becker, & Brenner, 2016), as shown in Fig. 5-T2 and Fig. 5-T3. Global path planning on these two maps results in two paths with different values for three criteria: length, path search complexity, and distance to the nearest obstacle. The cost values are defined as f_{TRI} and f_{QUAD}, which are obtained by a weighted summation of the normalized criteria. A path with a higher f value indicates more efficient path planning compared to a path with a lower f value for the same environment and scenario but with a different representation (Fig. 5-T5). However, evaluating a single scenario is insufficient to label the environment. Hence, it is necessary to test multiple random scenarios (Fig. 5-T2). The label assigned to the environment will correspond to the representation with the higher probability of yielding the winning f value. This process repeats for all the environments in dataset and logs the results for all the environments listed in Fig. 5-T6. Fig. 5-T7 is increasing the dataset by data augmentation which is explained in the next section. The labelled dataset including the images and the corre- sponding label is used for training a CNN (Fig. 5-T8). The algorithm's details are presented in Algorithm 2.

Algorithm 2: Labelling the tiles

```
Input: Image of Environment as I_E
M_TRI = STL_Generate(I_E)
M_QUAD = QUAD_Generate(M_TRI)
Generate a random start_point as S_i
if S_i is insdie free-space then
   for i = 1: 50
      Generate a random goal_point as Gi
      if G_i is inside free-space then
         Path_TRI = FindPath(M_TRI,Si,Gi)
         f_TRI = Calculate_cost_value(Path_TRI)
         Path_QUAD = FindPath(M_QUAD,Si,Gi)
         f_QUAD = Calculate_cost_value(Path_QUAD)
      if f_TRI > f_QUAD then Counter_TRI++
      else
      Counter_QUAD++
      Si = Gi
   end for
If Counter_TRI > Counter_QUAD then Label(I_E) = TRI
else
Label(I_E) = QUAD
```

M_{TRI} and M_{QUAD} stand for maps with irregular triangles and uniform quadrangles (squares or QUADs). M_{TRI} is generated using the STL_Generate function with the input being the environment image. Subsequently, the QUAD_Generate function is applied to M_{TRI} to generate a grid-based representation of the environment. S_i and G_i denote randomly generated successive start and goal points. Path_TRI represents the path generated by the triangle representation map, M_{TRI}, along with random start and goal points. The FindPath function involves path planning using the A* algorithm. The Calculate_cost_value function computes the cost value using Eq. (3), and f_TRI represents the calcu- lated cost value for Path_TRI. Similarly, Path_QUAD and f_QUAD correspond to the path generated by the QUAD representation (grid) and

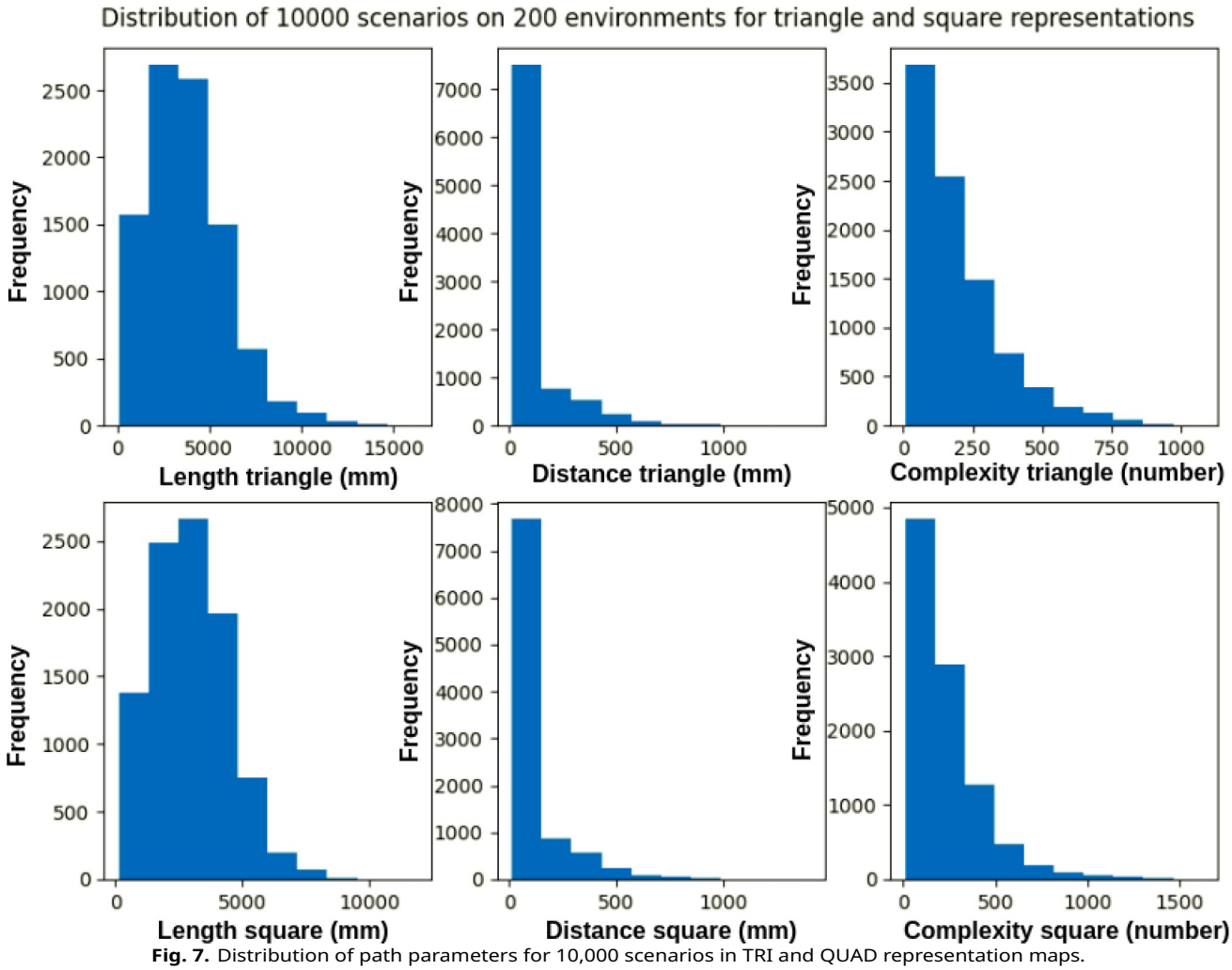


Fig. 7. Distribution of path parameters for 10,000 scenarios in TRI and QUAD representation maps.

its associated cost value. Counter_TRI and Counter_QUAD represent the number of scenarios in which either a TRI or QUAD yields a higher cost value. The Label function is employed to record the corresponding label of the environment, which takes a binary value of TRI or QUAD.

2.Simulation and experimental study

Experimental validation has been conducted to demonstrate that the hybrid representation map using a combination of triangle and quadrangle-based cells leads to a more efficient global path planning compared to the traditional single-only representations (regular quad- rangle or non-uniform triangle). To validate this point, the proposed hybrid representation map has been implemented on a Turtlebot-2 (Clearpath Robotics Inc, 2022) mobile robot platform. The physical environment for the experiment has been designed to comply with our lab constraints.

1.5. Implementation

Although the proposed approach is highly flexible for the selection of various tile sizes, a tile size of 5 m × 5 m has been considered for conducting the experiments owing to space limitations in the laboratory. Using Cellular Automata and random methods, 200 environments have been generated with structured and non-structured distributed obsta- cles, as shown in Fig. 5. Each environment is represented as an image of 369 × 369 pixels, with each pixel representing a 5 m × 5 m training environment or a single tile. The distribution of obstacles in each environment includes wide, narrow, and blocked passages. The images of the free areas in each environment were converted to the STL trian- gular mesh format using OpenCV (Bradski, 2000) (cv2) and PyMesh (Zhou, 2019). Additionally, quadrangle representations were generated using MAGIC libraries for automatic mesh generation (Cuilli`ere & Francois, 2014). The cell grid, or QUAD map, is 50 × 50 cells in size, which was selected based on a previous work (Meysami et al., 2022). To label each environment, 50 scenarios have been selected randomly, as depicted in Fig. 6. The cost value curves of each scenario were compared to determine the winner representation or label for that environment. Finally, Fig. 7 shows the range and distribution of the 10,000 scenarios, which resulted in generating a path of approximately 63 km in an area of about 5 km^2 used for labelling the 200 environments which is a considerable amount of data for training.

After preparing the 200 labelled environments, data augmentation was utilized (Taylor & Nitschke, 2018) to improve the performance of the CNN (Albawi, Mohammed, & Al-Zawi, 2017), which was trained to predict the best representation for a new environment. Data augmen- tation involves applying transformations such as rotation, scaling, and flipping to the existing training data, which helps to address issues such as overfitting and data scarcity. It was ensured that only transformations that did not alter the geometry were applied, as certain operations can modify the obstacle distribution. This approach resulted in a robust model with enhanced performance. The architecture of the CNN pro- vided by MATLAB was utilized (MathWorks, 2023) and the number of classes was adjusted to two, representing the QUAD and TRI represen- tations or 0 and 1, respectively. The CNN takes in black and white

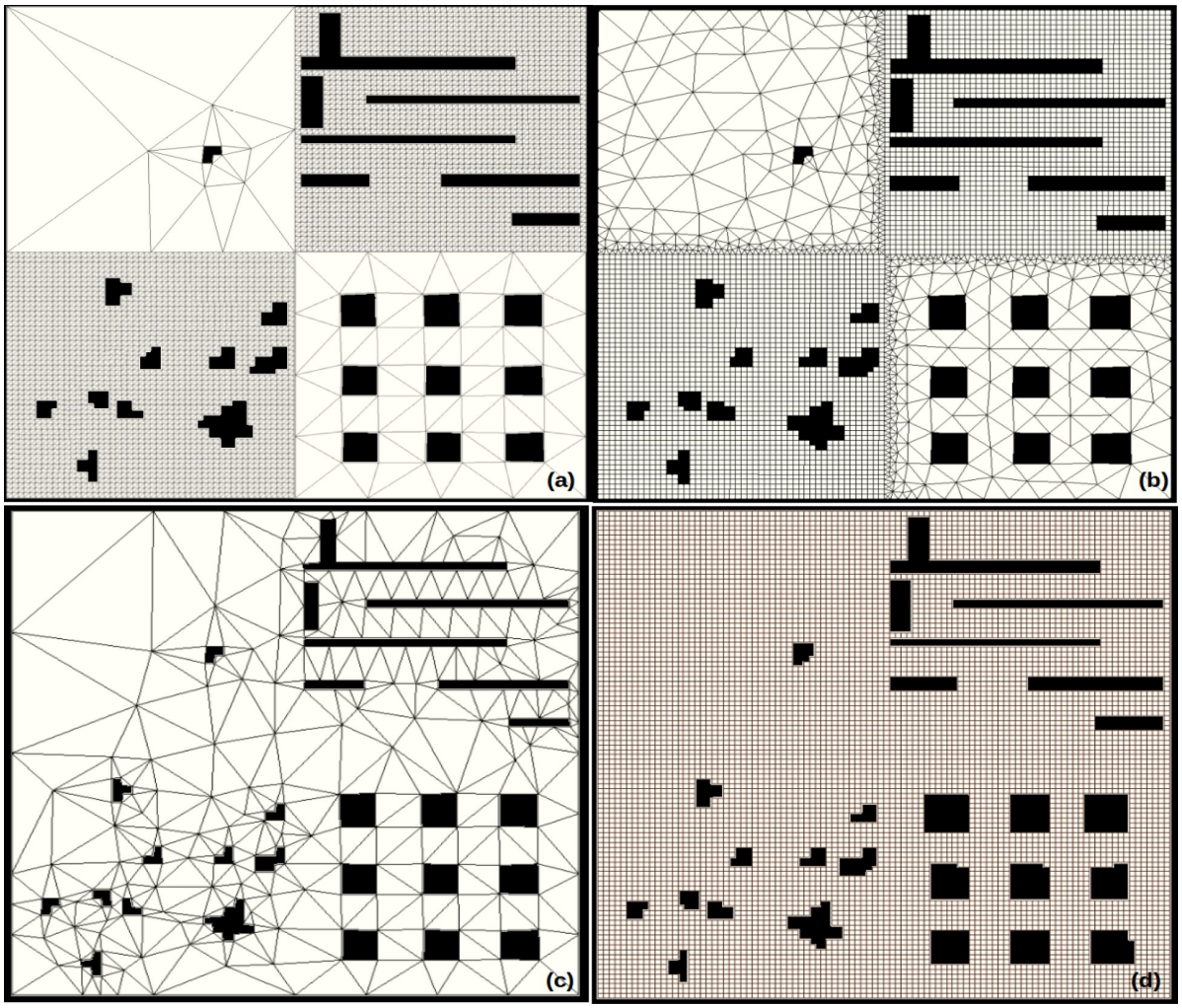


Fig. 8. Four types of representation maps used in this paper. (a) HYB-NC (b) HYB-C (c) TRI (d) QUAD.

images that are 28x28 pixels in size as input.

Using the trained model with 5 m × 5 m tiles, large, cluttered environments were generated by using the same CA and structured dis- tribution methods, in different factors of 5 in length and height. A naming convention is employed to designate various large environ- ments, following a format consisting of a combination of letters and numbers as LmapNx_Ny, where Lmap indicates a large map, and Nx and Ny represent the number of tiles in the x and y directions, respectively. The GMSH library (Geuzaine & Remacle, 2009) API was used to write codes for generating a conformal hybrid representation map. The G_{tile} was used as the template for defining the size map in its nodes based on the preferred representation and then a mesh was applied using the partition concept in the library. Fig. 8 shows all different representation maps for Lmap4_4, including the following hybrid and single representations:

- A hybrid representation map without conformity consideration be- tween meshes, named $M_{HYB\text{-}NC}$. A hybrid representation map with conformity consideration between meshes, named $M_{HYB\text{-}C}$.
- A single representation map with non-uniform triangle cells, named M_{TRI}.
- A single representation map with regular quadrangle cells, named M_{QUAD}.

The ROS platform was chosen to demonstrate the practicality of the generated path in real-life scenarios before conducting physical experiments. ROS, or Robot Operating System, is an open-source framework that simplifies the development of robots. A modular design, communication tools, and visualization aids are offered by ROS. The capability to connect with GAZEBO, a widely used open-source robotics simulator, is provided by ROS. To create a 3D model of the environment in GAZEBO, an STL file was created using FreeCAD. For navigation, the move_base package was utilized. A global path was published in using a plugin for Global_planner to ensure that the robot follows the generated path (Alajlan & Kouba^a, 2016). The Dynamic Window Approach (DWA)

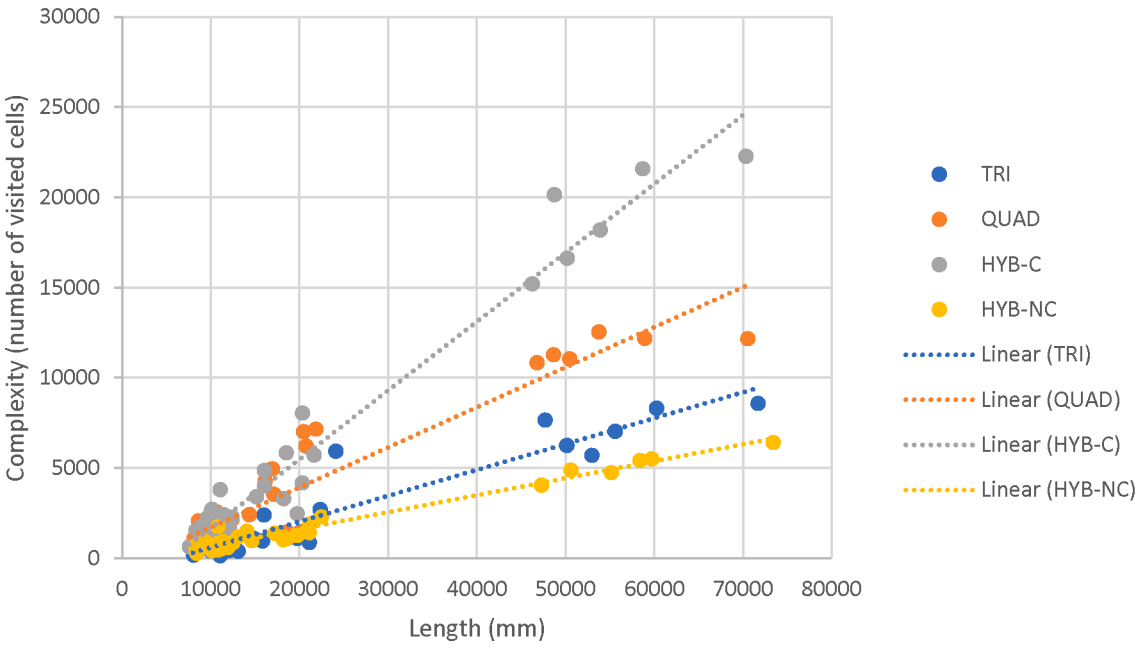


Fig. 9. Distribution of results with trend line for each representation with the slope of trend line as a metric for considering the path length and search complexity.

(Fox, Burgard, & Thrun, 1997) local planner was selected to follow the published path. DWA is a motion planning algorithm used in robotics for navigating based on the calculation of feasible velocities for a robot by considering its kinematic constraints to avoid obstacles. To achieve satisfactory results from the robot in completing the scenario, it was necessary to assign weights to DWA. These weights were chosen from (Zheng, 2021). DWA was opted as the local planner due to its readily available implementation in ROS and a reduced number of configura- tion parameters compared to other algorithms. The primary focus of this work lies on mapping and global planning aspects, and the utilization of the local planner serves primarily to validate and demonstrate the feasibility of the proposed method.

1.6. *Experimental validation*

For training purposes, several environments were generated to be used as large maps, and the best representations (either triangle or quadrangle) were determined tile by tile using the trained model. These environments feature various types of obstacle shapes and distributions, necessitating different best representations throughout the map. A large map can be as simple as two tiles, with one best represented by a TRI and the other by a QUAD (for the sake of implementation in the lab), or as complex as a random large map with 100 tiles. Scenarios were selected based on the inclusion of a path that encompasses at least two different representation tiles or more. The evaluation of the global path generated was based on three criteria: path length, distance to obstacles, and path search complexity. No smoothing method was applied to the generated path. The complete results are presented in the appendix. However, some illustrations demonstrate the difference between each represen- tation based on path length and path search complexity. The minimum distance to obstacles was not presented as all representations yielded nearly the same value. In Fig. 9, the scatter plot displays the distribution of all scenarios with points. The x-coordinate represents the path length, and the y-coordinate represents the path search complexity. The slope of the linear trend line for the points, which indicates an approximate measure of the required path search complexity per unit of path length, is the lowest for the HYB-NC representation.

For each scenario, the best representation map was selected among the four types based on having both the lowest path length and the lowest path search complexity. The criterion for considering the lowest

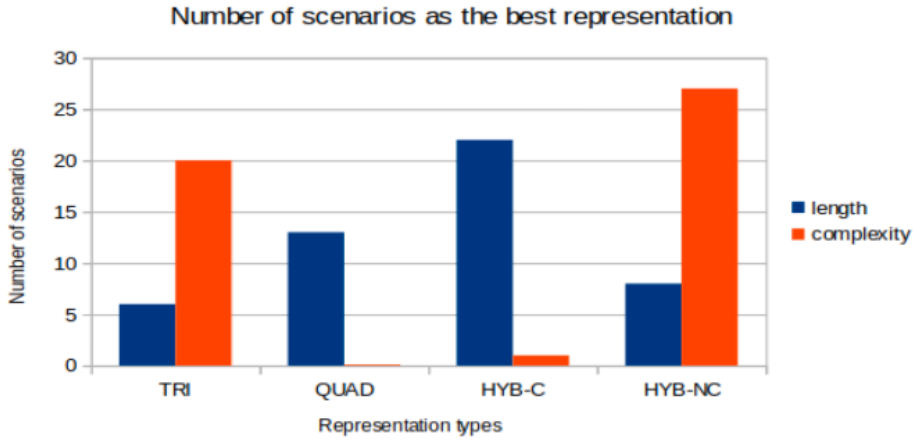


Fig. 10. Distribution of four representation maps as the best in length and path search complexity.

Table 1
Parameter and specification of turtlebot-2.

Representation Type	Average length (mm)	Average Complexity(Number of visited cells)
TRI	19,964	2019
QUAD	19,382	3780
HYB-C	19,353	5235
HYB-NC	20,055	1619

path search complexity was when its length exceeded the best answer for the length by no more than 10 %. The complete results are provided in the appendix, but a summary of the results is presented in Fig. 10. HYB-C couldn't compete with others like QUAD in terms of path search complexity due to the high number of cells in the transition region be- tween two neighboring tiles. However, it performed better in finding the shortest path compared to QUAD. The zig-zag problem in the QUAD representation was solved by HYB-C by using the triangle as the best representation in some environments. HYB-NC emerged as a dominant representation for reducing path search complexity, even though its path length was more competitive with the TRI representation. Overall, compared to other single representation maps (triangle and square-only representations), HYB-C provided the shortest length at around 50 %, while HYB-NC offered the lowest path search complexity for about 60 % of scenarios. It is worth noting that the path search complexity of finding a rough path in the first layer of HYB-NC was added to the sum of path search complexity in the passed tiles for this representation type.

In addition, the average of all scenarios has been compared in Table 1. According to this table the complexity of HYB-NC representa- tion 25 % is lower than the second lowest one (TRI) with only 4 % longer length relative to the shortest length among all representations (HYB-C).To demonstrate the scalability of the approach, results for the Lmap10_10 environment are presented in Fig. 11. This environment comprises 100 tiles generated using the CA method, with blue and red colors indicating the best representation as triangles and quadrangles, respectively. Due to implementation limitations in generating large- sized conformal meshes, a comparison is made only between the hybrid representation HYB-NC and other single representations. The selected scenarios traverse different tiles to illustrate the impact of switching representations. Additionally, the feasibility of the generated path was validated using ROS simulation to ensure a safe and successful scenario. The simulation results include path length and path search complexity, as depicted in the curve representing the number of scenarios in Fig. 12. The results demonstrate that the HYB-NC representation exhibits a significant reduction in path search complexity while maintaining a path length similar to that of single representations (as

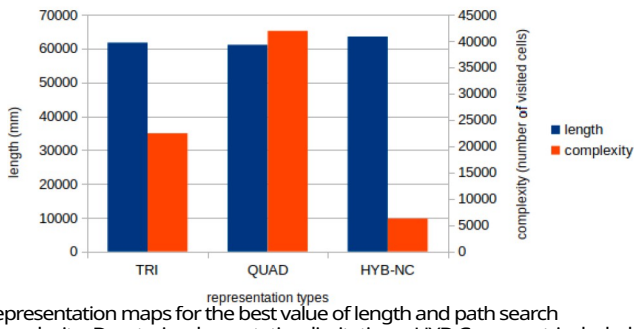


Fig. 12. Distribution of HYB-NC representation map as compared to single representation maps for the best value of length and path search complexity. Due to implementation limitations, HYB-C was not included in Lmap10_10.

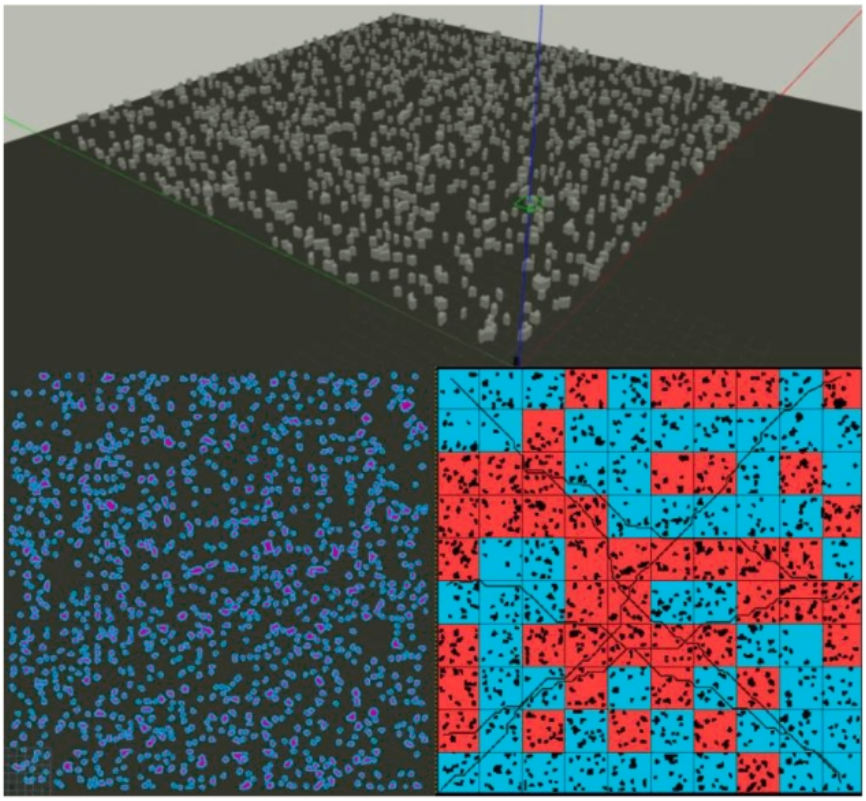


Fig. 11. Some scenarios on tiled labelled Lmap10_10 with GAZEBO and RVIZ view. The blue tiles show the environment which triangle representation totally gives higher cost value relative to the grid representation. The condition reverses for red tiles with grid representation as more efficient representation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

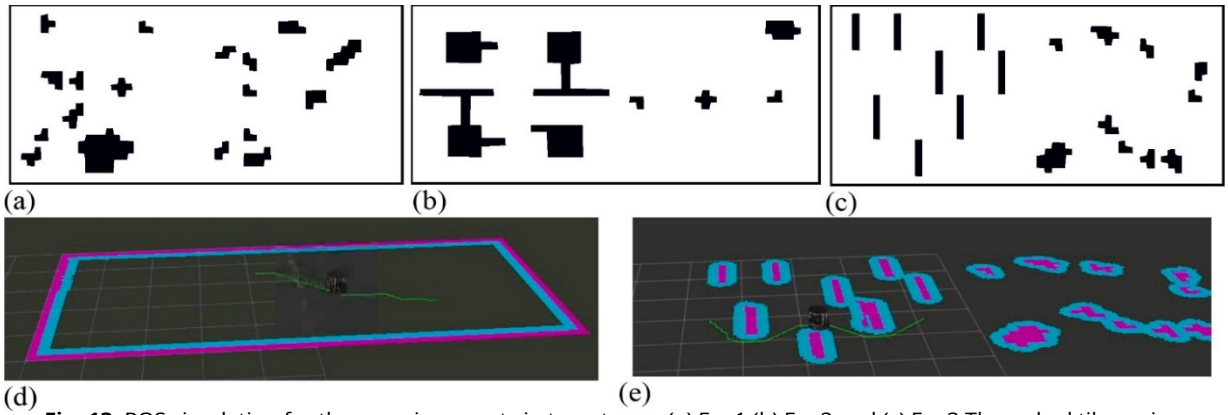


Fig. 13. ROS simulation for three environments in two stages. (a) Env1,(b) Env2 and (c) Env3 Three dual tile environments with both TRI and QUAD as best representations. (d) ROS simulation with the empty environment with the following published path – stage 1. (e) ROS simulation with obstacles according to the map and following the same published path – stage 2.

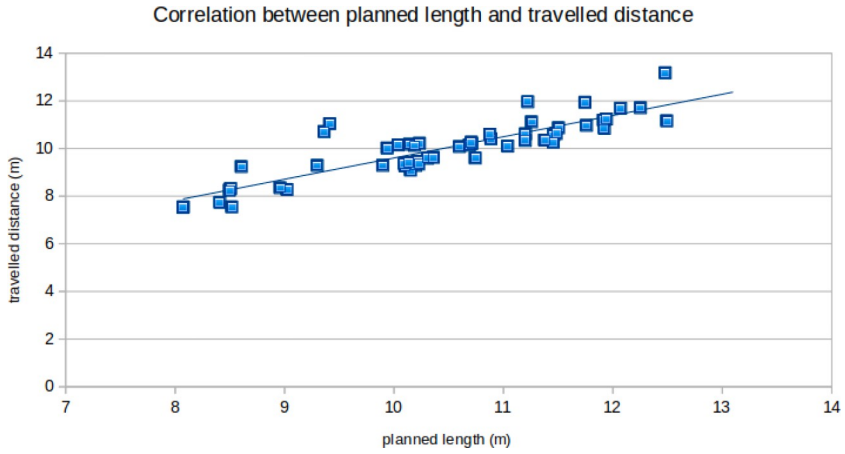


Fig. 14. Correlation between travelled distance and planned length for three environments.

previously mentioned, with no more than a 10 % increase in length). This suggests that path search complexity has a greater impact in large, cluttered environments than path length.

The correlation between planned path length and actual traveled distance, which is an important parameter in navigation due to its impact on energy consumption, was also analyzed. For testing, Imap1_2 environments were used, with one tile in TRI and the other in QUAD, featuring random and structured obstacle distributions simulating in- dustrial environments. Scenarios that traversed both tiles in different directions were selected, considering four types of map representations. Tests were conducted in two conditions: one in an empty area to mini- mize the local planner's influence, and another with obstacles consistent with the maps (Fig. 13). The results, presented in the appendix, reveal a correlation of 0.87 between planned length and traveled distance when the robot follows the planned path in an empty area (Fig. 14). However, in environments with obstacles, the correlation decreased to 0.67 due to the local planner's influence. Nevertheless, the results demonstrate the practicality of using hybrid representations for successful navigation in scenarios with safety and minimal failures. Additionally, Table 25 in the supplementary material shows that in 70 % of scenarios, the best rep- resentations for planned length also proved to be the best for traveled distance.

Figs. 15–17 display some of the environments and scenarios with various map representations.

1.7. Tests in real environments

The objective of the experiment is to show the practicability of the proposed approach in real environments and validating the result of the performed simulations. To do so, three environments (Fig. 18) have been implemented similar to the simulation environments in GAZEBO. These environments are 10 m × 5 m area according to the lab's size constraint and combination of two maps with triangle and square as the best representation.

The selected robot is a Turtlebot 2 (Fig. 19) with specifications ac- cording to Table 2. When converting a 400 mm inflation layer in global path generation into an inflation radius cost map, it is set to 0.2 m. The only difference between real experiments and simulation is in using the map. In simulation, a ground-truth map has been employed, while in the real experiment, a real map generated by the SLAM Gmapping method, which incorporates intrinsic errors and uncertainties, has been utilized. The reason for this strategy is for increasing the localization accuracy.

As in the simulation, the benefit of hybrid maps in global planning and the correlation between the global path and traveled distance were demonstrated. In practice, an attempt was made to measure the traveled distance to validate the simulation. A total of 60 tests were conducted, consisting of 5 scenarios for each environment with four different rep- resentations based on the generated global paths. The reported values for each test include traveled distance and traversal time. The error of

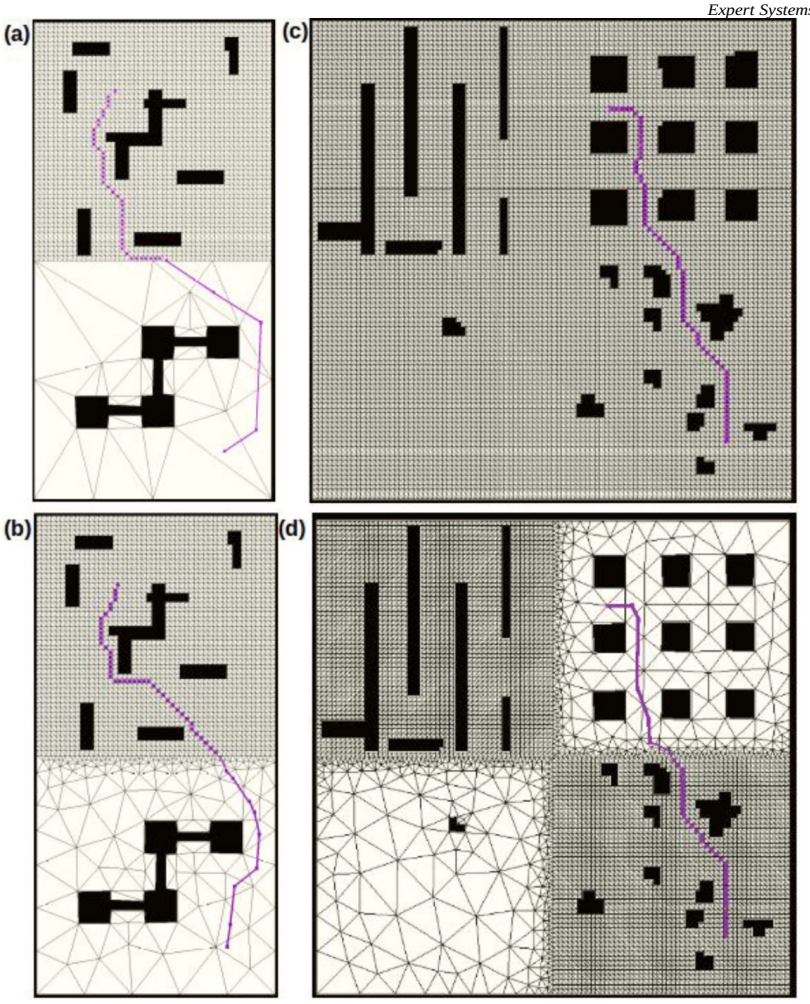


Fig. 15. Some large maps and scenarios (a) Lmap1_2_HYB-NC (b) Lmap1_2_HYB-C (c) Lmap2_2_QUAD (d) Lmap2_2_HYB-C.

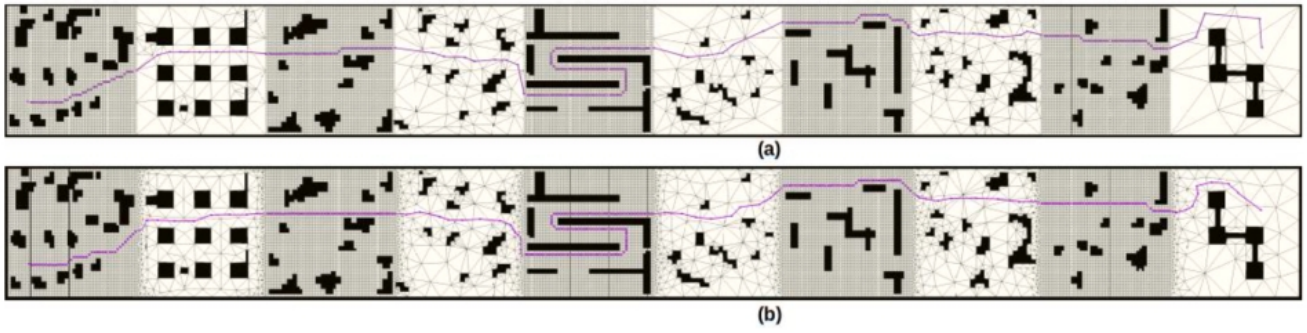


Fig. 16. Some large maps and scenarios. (a) Lmap1_10_HYB-NC (b)Lmap1_10_HYB-C

the traveled distance in Table 3 is calculated for each scenario as the percentage of the absolute value of the traveled distance in the simu- lation minus the traveled distance in the experiment, divided by the traveled distance in the experiment. In the table, "N/A" indicates scenarios that were not completed successfully due to issues such as colli- sions with obstacles or getting stuck.

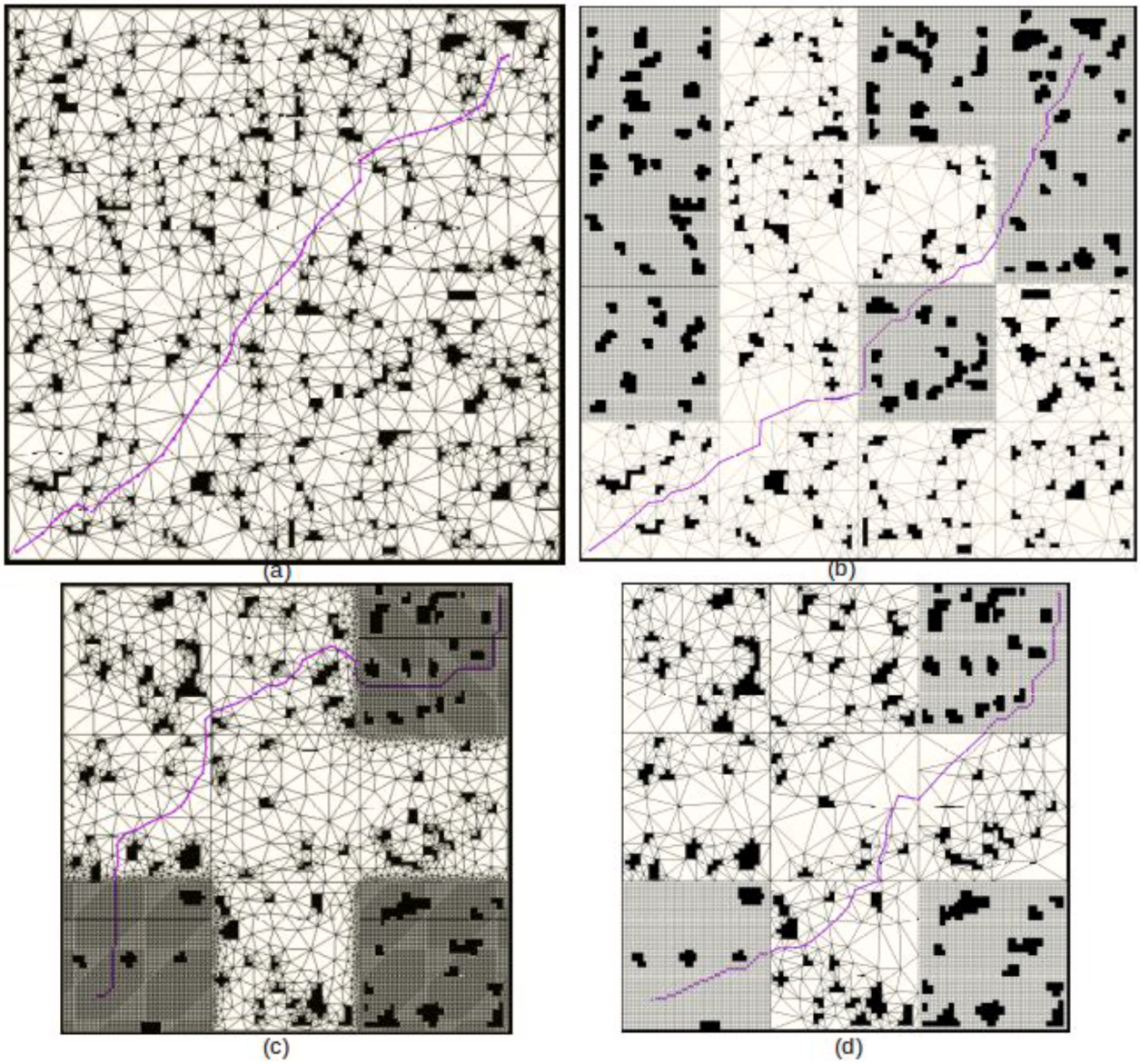


Fig. 17. Some large maps and scenarios. (a) Lmap4_4_TRI (b) Lmap4_4_HYB-NC (c) Lmap3_3_HYB-C (d) Lmap3_3_HYB-NC.

According to the table in Environment 1, TRI as a single representation could not successfully complete the scenarios due to an inconsistency between the globally generated path and the local planner implementation, which was not observed with other representations. The presence of a high traveled distance error (last column) in certain scenarios, such as Scenario 5 in Environment 3, is attributed to the aforementioned problem caused by the recovery behavior[2] to escape from getting stuck. Overall, there is a good consistency between the results (as shown in the Fig. 20), with an average traveled distance error below 5 %. The additional metric of traversal time was introduced to capture scenarios where the traveled distance is the same but exhibits a considerable difference in time. This difference indicates challenges with obstacle navigation.

In addition to the effect of the global path generated by each representation, which directly influences the behavior of the local planner, there are other factors contributing to the discrepancies between sim- ulations and real conditions. These factors include the terrain slope, presence of dust in the environment, and inherent measurement un- certainties in constructing obstacles, all of which contribute to the disparity between simulations and real-world tests.

Table 4 shows that TRI exhibits the highest number of failures among all representations, while QUAD has the lowest. On the other hand, HYB- C demonstrates the best average values for travelled distance and traversal time compared to other representations. The failure of the TRI representation is linked to a weakly planned path in turning obstacles, which intensifies due to imperfections in real conditions. The improved results in terms of traveled distance and traversal time with the HYB-C representation stem from the shorter length of the planned path using this representation.

As mentioned earlier, some noticeable disparities between experi- ments and simulations can often be attributed to the greater un- certainties and imperfections present in real-world environments, in contrast to the idealized conditions that simulated environments provide.

Discussion

Given the potential variety in the spatial distribution of obstacles

[2] - As an option provided in navigation stack in ROS to handle the problem like getting stuck.

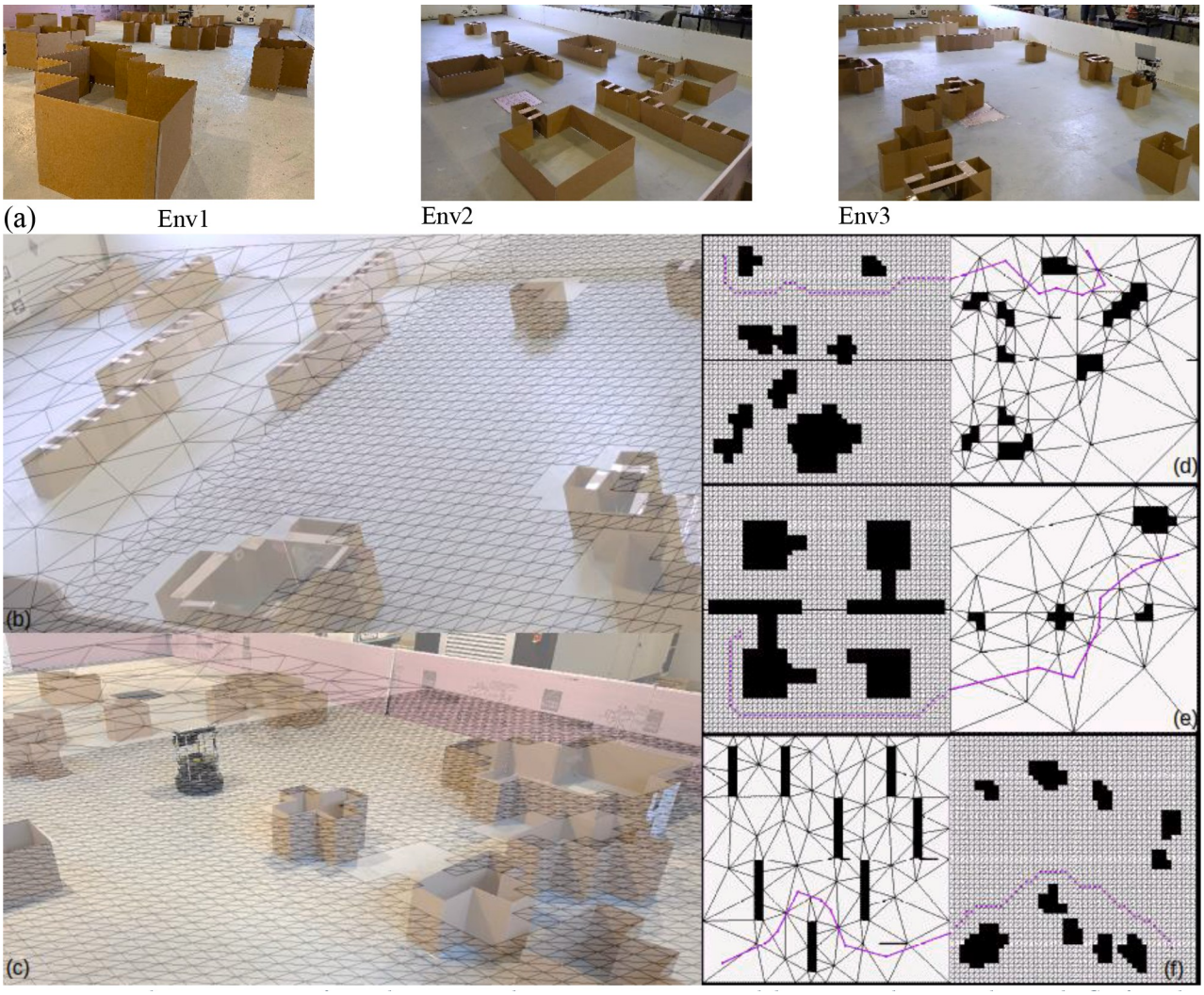


Fig. 18. (a) Three environments for implementing real test scenarios in IRH test lab center similar to simulation. (b) Conformal mesh of Env3. (c) Non-conformal mesh of Env1. (d)–(f) Sample scenarios with non-conformal mesh for three environments.



Fig. 19. The Turtlebot-2 robot with installed lidar used for tests.

Table 2
Parameter and specification of Turtlebot-2.

Parameter/Specification	Value/Definition
Radius	354 mm
Height	420 mm
Max speed	0.65 m/s
Sensors	Lidar
Software Driver	ROS Differential Kobuki Driver

within any large, cluttered environment, and considering that single representations like TRI or QUAD are only suitable for specific distributions, this hybrid novel approach emphasizes the benefits of merging these individual representations. This ensures that each subregion is effectively modeled using the most suitable meshing method which prepares two types of the hybrid representation maps were found to provide more efficient global paths compared to traditional single rep- resentations for both HYB-C and HYB-NC maps (as shown in the numerical results and Fig. 10). HYB-C enables the identification of shorter paths than the uniform quadrangle representation without requiring additional smoothing methods, thereby reducing post-processing costs. This representation type effectively addresses the zig- zag behavior of uniform quadrangles in diagonal directions within certain environments. On the other hand, HYB-NC significantly reduces path search complexity, with path lengths limited to no more than ten percent of the shortest possible path as the main assumption has been accepted for considering the lowest path search complexity among all representations. This criterion was particularly important for large, cluttered maps. This outcome was observed by analyzing the lowest slope concept of HYB-NC, which represents the ratio of the number of visited cells to the unit length of the optimal map. In addition to the main advantage of hybrid representation map, this approach combines

Table 3
Comparison between simulation and experiment results for three environments, five scenarios and four representations.

Env	Sc	Rep	Simulation		Experiment		Travelled distance error (%)[1]
			Travelled distance (m)	Traversal time (s)	Travelled distance (m)	Traversal time (s)	
Env1	Sc1	TRI	N/A	N/A	N/A	N/A	N/A
		QUAD	11.25	24.28	11.88	25.17	5.3
		HYB-C	11.22	28.29	11.78	26.7	4.75
		HYB-NC	11.08	23.01	11.78	26.7	5.94
	Sc2	TRI	N/A	N/A	N/A	N/A	N/A
		QUAD	11.38	23.96	12	24.33	5.17
		HYB-C	11.22	50.45	12.51	61.66	10.31
		HYB-NC	11.41	23.97	12.13	29.16	5.94
	Sc3	TRI	N/A	N/A	N/A	N/A	N/A
		QUAD	9.88	22.34	10.47	23.78	5.64
		HYB-C	9.94	23.9	10.26	24.62	3.12
		HYB-NC	9.91	24.25	10.72	27	7.56
	Sc4	TRI	N/A	N/A	N/A	N/A	N/A
		QUAD	7.07	14.75	7.15	19.97	1.12
		HYB-C	7.16	14.52	7.28	16.9	1.65
		HYB-NC	N/A	N/A	N/A	N/A	N/A
	Sc5	TRI	N/A	N/A	N/A	N/A	N/A
		QUAD	8.58	18.8	8.78	24.74	2.28
		HYB-C	N/A	N/A	N/A	N/A	N/A
		HYB-NC	N/A	N/A	N/A	N/A	N/A
Env2	Sc1	TRI	10.6	21.76	10.91	31.75	2.84
		QUAD	10.57	34.68	10.934	28.2	3.33
		HYB-C	11.07	36.04	10.59	25.62	4.53
		HYB-NC	10.56	22.69	10.829	27.67	2.48
	Sc2	TRI	11.1	28.46	11.199	27.09	0.88
		QUAD	10.89	22.04	N/A	N/A	N/A
		HYB-C	11.22	90.75	11.04	31.49	1.63
		HYB-NC	11.09	23.07	11.41	36.31	2.8
	Sc3	TRI	11.68	23.79	11.87	29.37	1.6
		QUAD	11.8	22.8	11.98	52.81	1.5
		HYB-C	11.58	24.08	N/A	N/A	N/A
		HYB-NC	11.79	24.11	11.85	45.47	0.51
	Sc4	TRI	10.74	22.32	9.88	82.77	8.7
		QUAD	9.69	19.41	10.18	22.06	4.81
		HYB-C	11.19	20.77	9.85	23.11	13.6
		HYB-NC	11.43	23.73	9.84	21.92	16.16
	Sc5	TRI	N/A	N/A	8.39	19.66	N/A
		QUAD	8.25	17.81	8.29	22.49	0.48
		HYB-C	8.04	16.76	8.3	17.63	3.13
		HYB-NC	8.25	16.15	8.25	18.13	0
Env3	Sc1	TRI	7.57	17.18	7.64	17.54	0.92
		QUAD	7.39	15.72	7.82	16.82	5.5
		HYB-C	7.7	18.49	7.84	16.19	1.79
		HYB-NC	7.71	16.66	8.13	18.14	5.17
	Sc2	TRI	10.34	21.62	10.64	22.05	2.82
		QUAD	10.18	21.81	10.72	25.59	5.04
		HYB-C	10.23	22.39	10.89	32.75	6.06
		HYB-NC	10.31	20.9	10.96	27.46	5.93
	Sc3	TRI	9.31	19.85	9.84	36.8	5.39
		QUAD	9.3	21.57	9.53	27.67	2.41
		HYB-C	9.25	21.44	9.5	26.45	2.63
		HYB-NC	9.37	21.71	9.61	23.18	2.5
	Sc4	TRI	9.39	30.6	10.1	27	7.03
		QUAD	9.39	38.07	9.59	27.38	2.09
		HYB-C	9.44	37.05	9.55	26.69	1.15
		HYB-NC	9.53	20.04	11.01	45.46	13.44
	Sc5	TRI	10.29	37.15	15.42	126	33.27
		QUAD	10.32	79.9	9.93	25.18	3.93
		HYB-C	9.8	26.57	9.89	27.77	0.91
		HYB-NC	10.56	28.37	9.92	25.18	6.45

[1] Error = ABS(($Tr_{Simulation}$ — $Tr_{Experiment}$) * 100/$Tr_{Experiment}$) which Tr stands for the travelled distance and ABS() as the absolute value function.

learning methods and traditional path planning techniques, such as using a CNN with A*, to achieve more efficient path planning. This approach aligns with recent research trends and found results are comparable to (Ferreira, Júnior, Galva˜o, Fernandes, & Barros, 2020), which proposed a combination of CNN with A* to reduce the number of iterations for finding optimal paths. However, this work goes beyond (Ferreira, Júnior, Galv˜ao, Fernandes, & Barros, 2020), by not only considering multi-criteria for path planning but also evaluating the practicability of the approach in large cluttered environments, in contrast to the limited scope of Ferreira et al. (2020) that focuses on a small area. Performed simulation takes into account the size of the robot through the use of an inflation layer, a consideration that was not

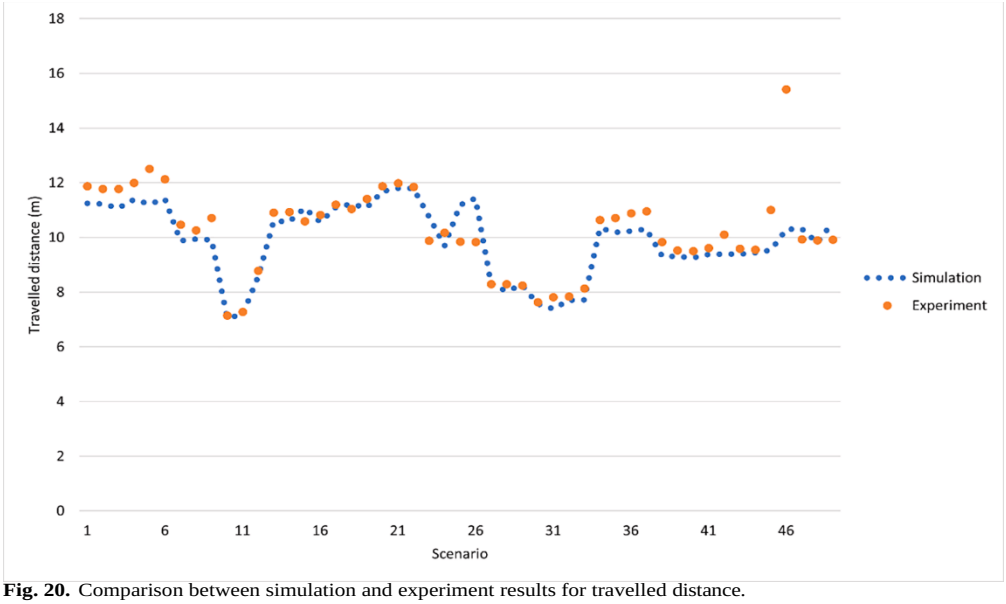


Fig. 20. Comparison between simulation and experiment results for travelled distance.

Table 4
Average behaviour of representations in all scenarios.

Representation in	Average travelled distance experiment(m)	Average traversal time (s)	Num of Failure
TRI	10.05	32.67	5
QUAD	9.88	27.87	1
HYB-C	9.69	24.99	2
HYB-NC	10.21	29.3	2

addressed in previous multi-objective path planning studies such as Hidalgo-Paniagua, Vega-Rodríguez, and Ferruz (2016). Additionally, this work eliminates the need to determine a metric for environmental difficulty, unlike (Perille et al., 2020) and establishes a direct connec- tion between the environment image and the best representation for efficient global paths. This proposed approach for indoor industrial changeable environments differs from Santos et al. (2022), which relies on predefined paths and is more suitable for outdoor agriculture trains with relatively fixed pathways.

Another advantage of this work lies in its generality, allowing for adjustment of tile sizes based on the environment and robot dimensions. Furthermore, it is not limited to the three mentioned criteria and can be applied while considering other cost functions.

The effect of conformity between meshed tiles, also was explored and found that, unlike the FEM solver, which typically requires conformity in hybrid meshes, conformality is not necessary for hybrid representa- tion maps. In fact, deviating from this conformity can even reduce path search complexity compared to conformal hybrid representation maps. It is important to note that after generating an HYB-C map, there is no distinction between tiles, and this map behaves in global path planning as a single representation map with integration between its elements. While the bottleneck of this approach lies in the mesh generation stage rather than path planning, an HYB-NC map requires a simple map generation process where each tile is treated as an individual M_{REP}. Path planning then requires an additional stage of finding the start and goal in each tile without any difficulty. The independence between tiles enables the use of a parallel approach for mesh generation, resulting in faster generation of large maps. Furthermore, an HYB-NC is a double-layer map that allows for the individual indexing of each tile, which proves useful in semantic mapping or managing dynamic changes within each tile without affecting other parts of the map.
Additionally, observing the behavior of global paths generated by the hybrid maps in real scenarios demonstrated an acceptable level of performance relative to the current representation in robotics, QUAD, and even outperformed the TRI representation, which experienced more failures in completing scenarios. However, it is important to note that the goal of the experiment was not an exhaustive comparison of the representations' behaviors relative to the local planner. Instead, it aimed to validate the feasibility of the proposed approach for real-world implementation.

In the study, the assumption is made for environments where ob- stacles don't obstruct divisions on a large map in a conformal hybrid representation. This simplifies the process without affecting outcomes. However, opting for a conformal hybrid representation would demand a specific mesh generation platform, achievable with extra effort.

In a previous work (Meysami et al., 2022) our team investigated the application of variable-size triangular meshes, offering a competitive option to uniform quadrangle cell-based representations. Indeed, triangular meshes easily allow using smaller or bigger elements where needed and this for various reasons (narrow passages, high curvature boundaries, etc.). This variable size approach could also be explored either as an alternative to hybrid representation maps or inside hybrid representation maps, representing a potential avenue for future research.

Conclusion

Constructing a map for large, cluttered indoor environments, using a single representation, such as small uniform quadrangles or non- uniform triangles, poses a challenging task. This challenge arises from the increasing complexity of path searching due to the map's size and obstacle distribution. To address this issue, this manuscript puts forward a novel hybrid representation map, which generates efficient global paths compared to traditional regular quadrangle or triangle representations in these environments. The need for explicitly defining the environment's difficulty using fixed metrics is eliminated by this approach, which has been a subject of debate among researchers. The large map is divided into tiles, and a trained model is utilized to determine the most

suitable representation for each tile. Global paths for this hybrid map are generated, both with and without consideration of the conformation between neighboring tiles, and the results are compared with those obtained from single-type maps. The non-conformal hybrid represen- tation significantly reduces path search complexity while maintaining a path length increase of no more than 10 %. On average, a 25 % reduction in complexity compared to the successful and fast TRI representation for pathfinding was achieved, while only resulting in a 4 % longer path length than the widely used QUAD representation in robotics. Further- more, practical scenarios have been implemented in real environments using a mobile robot to verify the practicality of this approach. The hybrid representations demonstrated their feasibility for robot applica- tions in real scenarios, in contrast to the single triangle representation, which had the highest failure rate.

In future research, exploration of the impact of local changes in the environment on the selection of the best representation will be under- taken. This ability will enhance the effectiveness of the two-layer non- conformal hybrid representation map in managing large dynamic en- vironments. Another direction for future research could involve inves- tigating the influence of tile and mesh size variations on the performance of the hybrid map. Additionally, the objective function will be refined based on insights gained from practical experimentation to achieve better alignment with the local planner.

CRediT authorship contribution statement

A**hmadreza Meysami:** Conceptualization, Methodology, Software, Writing, Visualization, Investigation, Writing – original draft. **Sousso Kelouwani:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Jean- Christophe Cuilliere:** Conceptualization, Methodology, Writing – re- view & editing, Supervision. **Vincent Francois:** Conceptualization, Methodology, Software, Supervision. **Ali Amamou:** Investigation, Re- sources, Data curation, Project administration. **Bilel Allani:** Software, Validation, Formal analysis, Investigation, Writing – review & editing, Visualization.

Declaration of competing interest

The authors declare the following financial interests/personal re-lationships which may be considered as potential competing interests: Sousso Kelouwani reports financial support was provided by Natural Sciences and Engineering Research Council of Canada.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work was supported by Natural Sciences and Engineering Research Council of Canada, grant Number CRSNG RDCPJ 518029-18.
Funding. The authors declare that no funds, grants, or any financial assistance, were received while preparing this manuscript.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi. org/10.1016/j.eswa.2024.123388.

References

Aggarwal, R., & Kumar, M. (2020). Chance-Constrained Approach to Optimal Path *Planning for Urban UAS. Paper presented at the AIAA Scitech 2020 Forum*.

Alajlan, M., & Koubˆaa, A. (2016). Writing global path planners plugins in ROS: A tutorial.

Robot Operating System (ROS) The Complete Reference (Vol. 1,, 73–97.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional *neural network. Paper presented at the 2017 international conference on engineering and technology (ICET)*.

Ammar, A., Bennaceur, H., Chaˆari, I., Koubˆaa, A., & Alajlan, M. (2016). Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments. *Soft Computing*, *20*, 4149–4171.

Anderson, G. T., & Yang, G. (2007). A proposed measure of environmental complexity for *robotic applications. Paper presented at the 2007 IEEE International Conference on Systems, Man and Cybernetics*.

Bradski, G. (2000). The openCV library. Dr. Dobb's Journal: Software Tools for the Professional Programmer, *25*(11), 120–123.

Brondani, J. R., de Lima Silva, L. A., Zacarias, E., & de Freitas, E. P. (2019). Pathfinding in hierarchical representation of large realistic virtual terrains for simulation systems. *Expert Systems with Applications*, *138*, Article 112812.

Cebollada, S., Pay´a, L., Flores, M., Peidro´, A., & Reinoso, O. (2021). A state-of-the-art *review on mobile robotics tasks using artificial intelligence and visual data. Expert Systems with Applications*, *167*, Article 114195.

Chagas, C., Zacarias, E., de Lima Silva, L. A., & de Freitas, E. P. (2022). Hierarchical and smoothed topographic path planning for large-scale virtual simulation environments. *Expert Systems with Applications*, *189*, Article 116061.

Clearpath Robotics Inc. Turtlebot-2-Open-Source-Robot. Available online: https://clear pathrobotics.com/turtlebot-2-open-source-robot/ (accessed on 9 April 2022).

Cuilli`ere, J.-C., & Francois, V. (2014). Integration of CAD, FEA and topology optimization through a unified topological model. *Computer-Aided Design and Applications*, *11*(5), 493–508.

Ding, J., Zhou, Y., Huang, X., Song, K., Lu, S., & Wang, L. (2023). An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling. *Journal of Computational Science*, *67*, Article 101937.

Duchonˇ, F., Babinec, A., Kajan, M., Benˇo, P., Florek, M., Fico, T., & Juriˇsica, L. (2014). *Path planning with modified a star algorithm for a mobile robot. Procedia Engineering*, *96*, 59–69.

Ferreira, J., Júnior, A. A., Galva˜o, Y. M., Fernandes, B. J., & Barros, P. (2020). Cnn encoder to reduce the dimensionality of data image for motion planning. *arXiv preprint arXiv:2004.05077*.

Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, *4*(1), 23–33.

Geuzaine, C., & Remacle, J. F. (2009). Gmsh: A 3-D finite element mesh generator with *built-in pre-and post-processing facilities. International journal for numerical methods in engineering*, *79*(11), 1309–1331.

Guimara˜es, R. L., de Oliveira, A. S., Fabro, J. A., Becker, T., & Brenner, V. A. (2016). ROS *navigation: Concepts and tutorial. Robot Operating System (ROS) The Complete Reference (Volume, 1)*, 121–160.

Guivant, J., Nebot, E., Nieto, J., & Masson, F. (2004). Navigation and mapping in large unstructured environments. *The International Journal of Robotics Research*, *23*(4–5), 449–472.

Held, J., Lampe, A., & Chatila, R. (2006). Linking mobile robot performances with the *environment using system maps. Paper presented at the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Hidalgo-Paniagua, A., Vega-Rodríguez, M. A., & Ferruz, J. (2016). Applying the MOVNS (multi-objective variable neighborhood search) algorithm to solve the path planning problem in mobile robotics. *Expert Systems with Applications*, *58*, 20–35.

Jeong, I. B., Lee, S. J., & Kim, J. H. (2019). Quick-RRT*: Triangular inequality-based *implementation of RRT* with improved initial solution and convergence rate. Expert Systems with Applications*, *123*, 82–90.

Liu, L., Wang, X., Yang, X., Liu, H., Li, J., & Wang, P. (2023). Path planning techniques *for mobile robots: Review and prospect. Expert Systems with Applications*, *120254*.

Liu, C., Zhang, G., Rong, Y., Shao, W., Meng, J., Li, G., & Huang, Y. (2023). Hybrid *metric-feature mapping based on camera and Lidar sensor fusion. Measurement*, *112411*.

MathWorks, (2023). Deep Learning Toolbox: User's Guide. Retrieved June 14, 2022, from https://www.mathworks.com/help/deeplearning/gs/create-simple-image -classification-network-using-deep-network-designer.html.

Meysami, A., Cuilli`ere, J.-C., François, V., & Kelouwani, S. (2022). Investigating the impact of triangle and quadrangle mesh representations on AGV path planning for various indoor environments: With or without inflation. *Robotics*, *11*(2), 50.

Mthabela, C., Withey, D., & Kuchwa-Dube, C. (2021). RRT based path planning for mobile robots on a 3D surface mesh. In 2021 Southern African universities power engineering conference/robotics and mechatronics/pattern recognition association of South Africa (SAUPEC/RobMech/PRASA) (pp. 1-6). IEEE.

Nieto, J. I., Guivant, J. E., & Nebot, E. M. (2004). *The hybrid metric maps (HYMMs): A novel map representation for DenseSLAM*. Paper presented at the IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004.

Nitsche, M., de Cristo´foris, P., Kulich, M., & Koˇsnar, K. (2011). Hybrid mapping for *autonomous mobile robot exploration. Paper presented at the Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems*.

Perille, D., Truong, A., Xiao, X., & Stone, P. (2020). Benchmarking metric ground *navigation. Paper presented at the 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*.

Ran˜o´, I., & Minguez, J. (2006). Steps towards the automatic evaluation of robot obstacle *avoidance algorithms. Paper presented at the Proc. of workshop of benchmarking in robotics, in the IEEE/RSJ int. conf. on intelligent robots and systems (IROS)*.

Ravankar, A. A., Ravankar, A., Emaru, T., & Kobayashi, Y. (2017, September). A hybrid topological mapping and navigation method for large area robot mapping. In *2017 56th annual conference of the Society of Instrument and Control Engineers of Japan (SICE)* (pp. 1104-1107). IEEE.

Samet, H. (1984). The quadtree and related hierarchical data structures. ACM Computing Surveys (CSUR), 16(2), 187–260.

Santos, L. C., Aguiar, A. S., Santos, F. N., Valente, A., & Petry, M. (2020). Occupancy grid and topological maps extraction from satellite images for path planning in agricultural robots. *Robotics*, *9*(4), 77.

Santos, L. C., Santos, F. N., Valente, A., & Costa, P. (2022). Path Planning with Hybrid *Maps for processing and memory usage optimisation. Paper presented at the 2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*.

Sartori, D., Ermacora, G., Pei, L., Zou, D., & Yu, W. (2020). A CNN approach to assess *environment complexity for robotics autonomous navigation. Paper presented at the 2020 3rd International Conference on Mechatronics, Robotics and Automation (ICMRA)*.

Speray, D., & Kennon, S. (1990). Volume probes: Interactive data exploration on arbitrary grids. *ACM SIGGRAPH Computer Graphics*, *24*(5), 5–12.

Taylor, L., & Nitschke, G. (2018). Improving deep learning with generic data *augmentation. Paper presented at the 2018 IEEE symposium series on computational intelligence (SSCI)*.

Tsardoulias, E., Iliakopoulou, A., Kargakos, A., & Petrou, L. (2016). A review of global path planning methods for occupancy grid maps regardless of obstacle density.
Journal of Intelligent & Robotic Systems, 84(1), 829–858.

Wang, J., Li, B., & Meng, M. Q. H. (2021). Kinematic Constrained Bi-directional RRT with *Efficient Branch Pruning for robot path planning. Expert Systems with Applications*, *170*, Article 114541.

Wang, J., Zhang, T., Ma, N., Li, Z., Ma, H., Meng, F., & Meng, M. Q. H. (2021). A survey of learning-based robot motion planning. *IET Cyber-Systems and Robotics*, *3*(4), 302–314.

Wolfram, S. (1984). Cellular automata as models of complexity. *Nature*, *311*(5985), 419–424.

Xiao, X., Wang, Z., Xu, Z., Liu, B., Warnell, G., Dhamankar, G., & Stone, P. (2022). Appl: Adaptive planner parameter learning. *Robotics and Autonomous Systems*, *154*, Article 104132.

Xin, P., Wang, X., Liu, X., Wang, Y., Zhai, Z., & Ma, X. (2023). Improved bidirectional RRT* algorithm for robot path planning. *Sensors*, *23*(2), 1041.

Zammit, C., & Van Kampen, E. J. (2022). Comparison between A* and RRT algorithms for 3D UAV path planning. *Unmanned Systems*, *10*(02), 129–146.

Zheng, K. (2021). Ros navigation tuning guide. Robot Operating System (ROS). The Complete Reference (Vol. 6,, 197–226.

Zhong, X., Tian, J., Hu, H., & Peng, X. (2020). Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment. *Journal of Intelligent & Robotic Systems*, *99*, 65–77.

Zhou, Q. (2019). PyMesh—Geometry processing library for Python. *Software available for download at* https://github. com/PyMesh/PyMesh.

3.6 Conclusion:

Mapping complex indoor environments using a single representation, like quadrangles or triangles, is difficult because of size and obstacle challenges. It is hypothesized that a hybrid representation map can exhibit higher efficiency by striking a balance between parameters such as length, time, and safety, as compared to a single representation like QUAD or TRI, for navigation in large cluttered environments. This hypothesis was confirmed across several environments and was introduced a tiled hybrid map that adapts representations for different tiles through a trained model, removing the need for fixed difficulty metrics. The hybrid approach reduces pathfinding complexity by 25% compared to TRI, with a minor 4% increase in path length over QUAD. Practical tests with a mobile robot confirm the hybrid method's effectiveness, surpassing single representation maps, especially TRI, which had the most failures.

Chapter 4. A Tiled Large Map

4.1 Introduction

The proposed tiled hybrid map, as presented in previous work, was limited to a fixed tile size of 5m × 5m. Like other mapping methods, it is crucial to evaluate the effect of tile size—the fundamental building blocks of the tiled hybrid map—on path planning performance. Additionally, the methodology previously proposed for path planning relied on multiple platforms, requiring manual data transfer between them, which limited efficiency and scalability. To validate the effectiveness of the tiled hybrid representation for path planning across various large environments and to account for the impact of tile size on performance, a systematic and comprehensive approach is essential.

Moreover, since the core idea of the tiled hybrid map is its adaptability to different mapping methods based on project requirements, it was critical to explore its application in diverse environments, including cluttered, structured, and mixed configurations. Extending the original concept to accommodate multiple tile sizes and environment types enables a more thorough analysis of large-scale maps using specific metrics. These metrics provide a foundation for making informed decisions regarding the optimal mapping template and tile size before initiating navigation tasks or conducting extensive simulations. Overall, a tiled hybrid representation map that incorporates an appropriate tile size selection ensures a more efficient and effective path-planning process.

This chapter directly addresses sub-objective 3 of the thesis, which is the analysis of environment partitioning and tile-size selection to enable local adaptivity in the tiled hybrid representation map.

4.2 Article 3: AI-Driven Adaptive Tiled Hybrid Mapping for Efficient Autonomous Path Planning

Journal: Robotics and Autonomous Systems

Authors: Ahmadreza Meysami, Sousso Kelouwani, Jean-Christophe Cuilliere,

Vincent Francois

Initial date submitted: May 4, 2026 (under-review)

4.3 Methodology

The proposed approach builds on previous work by dividing a large environment into uniform, square-shaped sub-regions, referred to as tiles. For each tile, an optimal representation is selected between a non-uniform triangular cell-based structure and a regular quadrilateral cell-based structure using a pre-trained model from the prior study. If the model identifies a quadrilateral representation as optimal, the tile is treated as a grid graph. Conversely, if a triangular representation is preferred, the tile is represented using a triangular mesh in STL format. To convert the triangular mesh into a graph, its dual graph is utilized.

Once the graphs for all tiles are generated, they are connected by linking the boundary nodes of adjacent tiles, creating a unified graph that integrates the individual tile graphs. Path planning is then performed on this connected graph using the A* algorithm, enhanced with distance transforms and visibility checks to ensure nodes avoid obstacles.

For identical navigation scenarios, the results from various connected graphs are compared to identify the tile division size that provides the most efficient solution. To aid this analysis, a metric called the Participation Index has been introduced. This metric quantifies the balance between the use of triangular and quadrilateral representations in the tile configuration. A perfectly balanced configuration, where both representations are equally utilized, corresponds to a Participation Index value of 0. Positive values indicate a predominance of triangular representations, while negative values reflect a predominance of quadrilateral representations. This metric helps evaluate different tile divisions by assessing the contribution of each representation type.

As in previous work, three key metrics—path length, computational complexity, and minimum distance to obstacles—are used to evaluate the performance of the A* algorithm on unified graphs derived from different tile configurations. Comparing these metrics across various tile sizes highlights the most efficient tile size for a tiled hybrid map and examines whether a high balance between triangular and quadrilateral representations leads to improved performance for large-scale environments.

4.4 Summary of Analysis

Four distinct environments were analyzed:

- Lmap1: A cluttered, random environment.
- Lmap2: A mixed environment combining random and structured areas.
- Lmap3: A fully structured environment.
- Lmap4: A room-by-room environment with random and structured areas within the rooms.

Each environment is represented as a 50 m × 50 m square, equivalent to 25,000 times the robot's circular area (354 mm diameter), classifying these as large maps relative to the robot. Tile divisions (n) ranged from 2 to 20, with tile sizes varying from 25 m × 25 m (at n=2) to 2.5 m × 2.5 m (at n=20).
The Result for each environment is as follow:

• Lmap1: Divisions 9 and 10 provided the optimal balance between time and path length, aligning well with the participation index. Lower divisions generally favored shorter path lengths but required higher computation times, while higher divisions improved time performance and clustered in the mid-low region, except for divisions 12 and 18.

• Lmap2: Division 3 achieved the best balance between time and length, corresponding closely to the participation index's balanced value. A cluster of results in the light green region indicated shorter times, consistent with the predominance of triangular representations at higher divisions.

• Lmap3: Division 4 yielded the most efficient results, fully aligning with the participation index. Group behavior was also observed for higher division numbers.

• Lmap4: Analysis was constrained to lower divisions due to implementation limitations. Among these, Division 6, representing a single room's size, produced the best results, while larger divisions covered multiple rooms.

The statistical analysis shows that division size affects path-planning performance in a map-dependent manner. LMAP1 displays the lowest variability in both path length and CPU time, while LMAP2 and LMAP3 are far more sensitive—especially in computation time. Normalized ranges and CV values confirm that division influences CPU time more strongly than path optimality. Overall, mid-range divisions provide the best balance between path quality and computational cost in large, randomly structured environments, whereas structured or hybrid maps achieve optimal performance only at specific division levels. These results highlight tile-size selection as a key factor for enabling local adaptivity in large-scale navigation.

4.5 Article

AI-Driven Adaptive Tiled Hybrid Mapping for Efficient Autonomous Path Planning

Ahmadreza Meysami*
Department of Mechanical Engineering, UQTR, des Forges, Trois-Rivieres, Quebec, Canada
E-mail: ahmadreza.meysami@uqtr.ca
ORCID iD: https://orcid.org/0009-0004-4008-0196
*Corresponding Author

Sousso Kelouwani
Hydrogen Research Institute, Pavilion Tapan-K.-Bose, 3351 Bd des Forges, Trois-Rivières, Quebec, Canada.
E-mail: sousso.kelouwani@uqtr.ca

Jean-Christophe Cuilliere
Department of Mechanical Engineering, UQTR, des Forges, Trois-Rivieres, Quebec, Canada.
E-mail: jean-christophe.cuilliere@uqtr.ca

Vincent Francois
Department of Mechanical Engineering, UQTR, des Forges, Trois-Rivieres, Quebec, Canada.
E-mail: vincent.francois@uqtr.ca

Abstract: Efficient path planning in large indoor environments requires balancing path length, computational cost, and obstacle avoidance. Conventional mapping methods often lose efficiency as environment size and complexity increase, motivating the need for adaptive mapping strategies. This study introduces an adaptive tiling strategy that partitions the environment into square regions and aligns each region's map representation with its local structure. Within every tile, a trained classifier selects the more efficient representation—triangular mesh or quadrangular grid—based on spatial features through an AI-driven, environment-aware adaptive process. Connected graphs are constructed by linking local grids and triangular meshes, enabling global path planning across the entire space. To assess the influence of tile size, statistical measures including normalized range and coefficient of variation (CV) quantify division-induced variability in path length and computation time across different maps. A participation index is introduced to estimate the efficiency of different tile sizes without requiring full simulations. Simulations show that balanced use of both representations improves planning efficiency in random, structured, and mixed environments, while room-by-room layouts benefit from tiles sized to match individual room dimensions. Results also indicate that division sensitivity is map-dependent, with CPU time more affected by division changes than path length. The proposed hybrid approach improves large-scale navigation by adapting local representations to environmental structure, providing a scalable framework for autonomous systems operating in complex indoor spaces.

Index Terms: Large environment, Hybrid tiled map, graph representation, participation index.

1 Introduction

Large indoor environments remain challenging for global path planning because computation must scale with both map size and environmental complexity. In addition to improving search algorithms, the spatial representation of the environment strongly affects planning efficiency, yet the impact of tile size on hybrid map performance remains poorly understood. As map size grows, the search space expands rapidly, making optimal path computation increasingly costly [1]. Cluttered environments

further complicate planning, where path length, obstacle clearance, and computation time must all be balanced. The primary challenge in path planning for large environments has often focused on addressing computational resource limitations [2]. However, this focus frequently overlooks the efficiency of path planning across diverse environmental regions [3], introducing another significant challenge. A further limitation is that most existing mapping approaches lack AI-driven adaptation, applying a single representation across regions with very different spatial structures. This gap becomes increasingly significant as large environments exhibit diverse spatial structures that cannot be efficiently represented by a single mapping strategy.

To address computational issues, strategies such as algorithm improvements and hybridization have been explored. Algorithmic improvements aim to reduce computational load in classic planners such as A* [4], RRT [5], and PRM [6] to improve performance in large environments. Examples include Jump Point Search (JPS) [7] and [8], which accelerates A* by selectively expanding nodes; RRT* [9], which grows the tree more intelligently toward the goal; and Lazy PRM [10], which reduces collision checks. Further refinements include cropped neighbor selection in JPS(A*) [11], the depth-biased Quick-RRT* [12], and an adaptive-bias RRT* [13], though the latter remains untested in diverse large-scale settings. These methods illustrate the trade-offs between graph-based and sampling-based planners—A* often outperforms RRT in static maps, while RRT can excel in dynamic ones [14]. Even an approach like [15], which employs different sampling densities for open areas and narrow passages to show the effect of diversity on algorithm settings, relies on an offline process to ensure safety in cluttered environments and generates efficient paths. However, this limits the approach as the calculation of safety after small changes is more costly compared to methods that incorporate safety as part of the online search.

Hybridization techniques are applied in two main approaches. The first involves combining two different algorithms to leverage their respective advantages [16], while the second uses two different map types based on their effectiveness in path planning [17]. For example, in the first approach, [18] combines Voronoi diagrams for safe travel with fast marching methods for smooth paths in cluttered environments. However, this method has not been validated for use in large, cluttered environments. Work such as [19], which employs hybrid approaches by integrating algorithms or adding improvements to existing ones for efficient pathfinding in cluttered environments, often relies on parameter tuning to achieve optimal results. For instance, in [19], a parameter that balances exploration and exploitation requires fine-tuning based on the environment. However, in large, cluttered environments with varying densities of obstacles, such reliance on parameter tuning makes this approach less feasible as a general solution.

Hybridization techniques, such as the use of hybrid maps that combine different types of map layers, have also been explored [20]. One of the earliest works on hybrid mapping is [21] or [22], which integrate two types of maps while leveraging their respective advantages and addressing their limitations. This approach laid the foundation for many subsequent studies in hybrid mapping. Although AutoInspect [23] does not explicitly define itself as a hybrid mapping approach, it effectively integrates a topological map for autonomous navigation and a point cloud for localization, linking them through a pose graph. This structure combines the strengths of both representations—using the metric accuracy of point clouds for localization while leveraging the efficiency of topological graphs for path planning.

The paper [24] introduces a novel hybrid map representation that integrates 2D and 2.5D map features, enhancing path planning efficiency and safety in complex and unstructured terrains. This approach leverages the computational simplicity of 2D maps while incorporating the detailed terrain analysis provided by 2.5D maps, making it a practical solution for navigating outdoor environments. The study demonstrates that hybrid mapping techniques, traditionally associated with indoor navigation, can also effectively address challenges in outdoor robot navigation, highlighting their versatility and broader applicability.

These methods leverage the strengths of various map representations, like grid and feature maps, to

improve path planning efficiency [25]. The paper [26] introduces a novel hybrid mapping method with three layers: the scene level, the feature level, and the metric level. This approach relies on defining turning points near features to connect segment lines, which introduces additional work and requires human intervention for point definition. Furthermore, it does not address the safety of a path in proximity to obstacles. Recent work [27] introduced a hybrid map that combines feature-based representations with topological map structures and defines two types of nodes for localization and path planning. However, as noted by the authors, this hybrid methodology requires manual configuration of hyperparameters, which makes it less feasible for large maps with varying types of clutter and complexity. The use of hybrid map representations, such as the Hybrid Metric Map (HYMM) or a combination of metric and topological maps, has proven effective in addressing computational limitations while improving mapping efficiency [28].

However, most of these studies do not fully address the diversity present in large environments, which affects path planning performance in complex, variable settings. Unlike other works, [29] is designed to address the complexities of marine environments by integrating multiple representations to effectively manage both static and dynamic elements within the surroundings. These representations include Cubic Splines for dynamic objects, Point Features for stationary objects, and Occupancy Grids for landmasses, leading to improved mapping accuracy and computational efficiency.

Building on this body of work, we have developed a tiled mapping system that partitions large environments into smaller, manageable tiles. Each tile can be represented using one of two methods: a rectangular grid (similar to the occupancy grid method) or a triangular discretization, often referred to as an Irregular Triangular Mesh [30]. Earlier experiments comparing quadrangular and triangular grids have shown that performance varies with obstacle distribution. For instance, narrow passages benefit from quadrilateral representations for enhanced safety, obstacle avoidance, and time efficiency. Conversely, environments with large open areas or widely spaced obstacles often achieve better results with triangular representations, achieving a balance between time efficiency and path length. Despite evidence that representation influences planning efficiency, the effect of tile size on hybrid maps has not been systematically explored. To further examine how division influences planning performance, we also incorporate a statistical analysis using range and coefficient-of-variation (CV) metrics to quantify division-induced variability in both path length and computation time. Unlike prior hybrid mapping studies, this work provides a quantitative assessment of division sensitivity using normalized range and CV metrics, offering new insight into how tile size affects hybrid map performance. Furthermore, unlike existing static hybrid maps, our method adaptively selects QUAD or TRI per tile based on local structure. This paper analyzes how varying tile sizes affect navigation performance when tiles are adaptively assigned triangular or quadrangular representations.

The main contributions of this work are threefold. First, we extend adaptive tiled hybrid mapping by analyzing how tile size influences performance when QUAD or TRI representations are selected per tile based on local structure. Second, we provide a quantitative assessment of division sensitivity using normalized range and CV metrics, offering new insight into how tile size affects hybrid map performance. Third, we introduce a participation index that predicts efficient tile sizes without requiring full navigation simulations.

The remainder of this article is organized as follows. Section 2 describes the methodology for generating hybrid maps with varying tile sizes. Section 3 presents the simulation analysis, including implementation details, results, and evaluation across multiple large environments. Section 4 concludes the study and outlines directions for future research.

2 Methodology

The concept of tiles is widely used across various fields, including Geographic Information Systems (GIS) [31], Google Maps [32], video games [33], and mapping of a data center [34] due to

their scalability in managing large datasets. Dividing a large environment into uniform tiles results in either a high number of small tiles or a low number of large tiles, depending on the desired resolution.

Our mapping approach uses a tiled hybrid representation in which each tile is automatically assigned the most efficient representation for path planning (Fig. 1). The process has two stages: (i) training a model to classify tiles, and (ii) generating a connected hybrid map using this model.

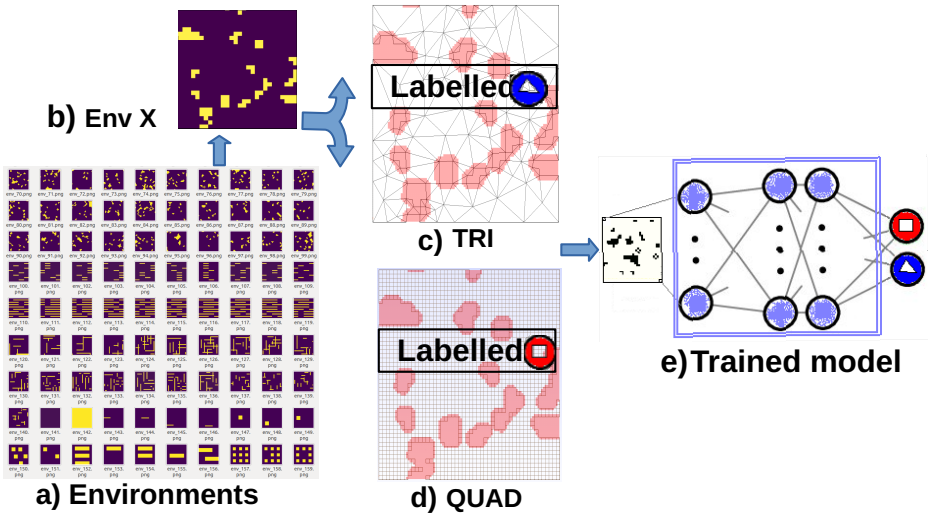


Fig.1. Methodology for training a model: a) Different styles of environments are selected, featuring varying obstacle sizes and distributions. b) A sample environment corresponding to a specific tile size is used to generate two different map representations: triangular and quadrangular. c) A triangular representation map is created with an inflation layer. d) A uniform quadrangle (QUAD) representation map is generated from the same environment. e) A Convolutional Neural Network (CNN) is trained with labeled environments, where TRI (blue) or QUAD (red) is determined as more efficient representation based on random path planning tests conducted on both maps (not shown).

The second section describes the tiling method, which applies this trained model to generate a tiled map for path planning in a large environment (Fig. 2).

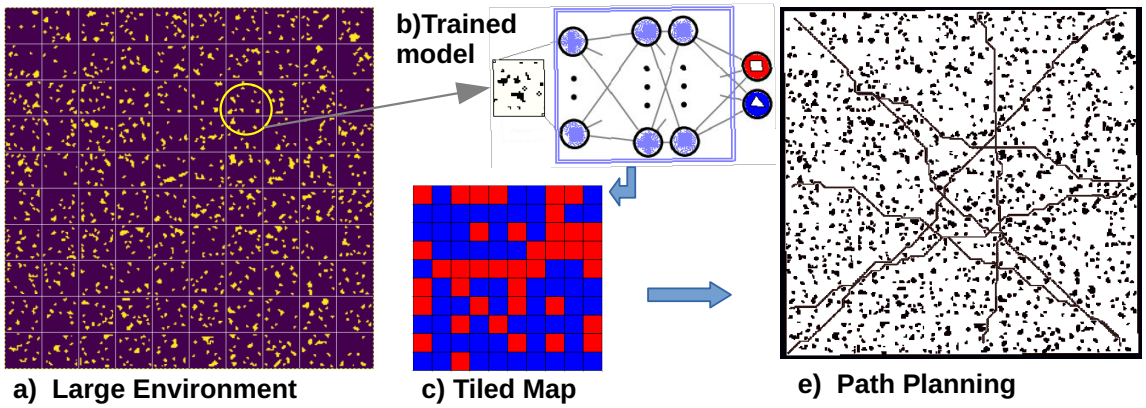


Fig.2. Methodology for generating a tiled map: a) A large environment is divided into uniform square tiles, and each tile's image is compared against the trained model to determine whether its representation should be TRI or QUAD (blue or red). b) A Convolutional Neural Network (CNN) is trained with labeled environments, where TRI (blue) or QUAD (red) is determined as the more efficient representation based on random path planning tests conducted on both maps (not shown) c) A virtual tiled map template is created for a specific tile size. d) The template map is then used for path planning as a unified map while leveraging the advantages of dual representations for optimal path efficiency.

A large dataset of black-and-white environment images with random and structured obstacles was generated using Cellular Automata, random methods, and manual design. Each image of size 28 × 28 pixels was labeled 0 (quadrangular more efficient) or 1 (triangular more efficient) by comparing A* planning outcomes on both representations. Efficiency was evaluated by a weighted combination of path length, computation time, and minimum distance to obstacles [35]. Because obstacle inflation produced comparable clearance for both maps, only path length and computation time were used in this work.

For each tile, if the quadrangular representation is selected as an optimal choice, a graph-based map in a specific grid structure is generated. This grid representation, resembling an occupancy grid map, connects each cell to its neighbors through either 4 or 8 neighboring cells. The center of each cell serves as a node in the corresponding graph for that tile.

If the triangular representation yields higher efficiency for path planning, based on the results from the trained model, the tile is represented as a triangle mesh stored in STL format. For path planning, similar to A* graph search method used in the grid representation, a graph is constructed where the centroids of the triangles act as nodes. This graph is defined as the dual graph of triangular mesh, with edges connecting neighboring triangles.

With this approach, each tile in a hybrid map representation is represented either as a grid graph or a dual graph of the triangular mesh (Fig. 3), depending on the result obtained from applying the trained model to the environment.

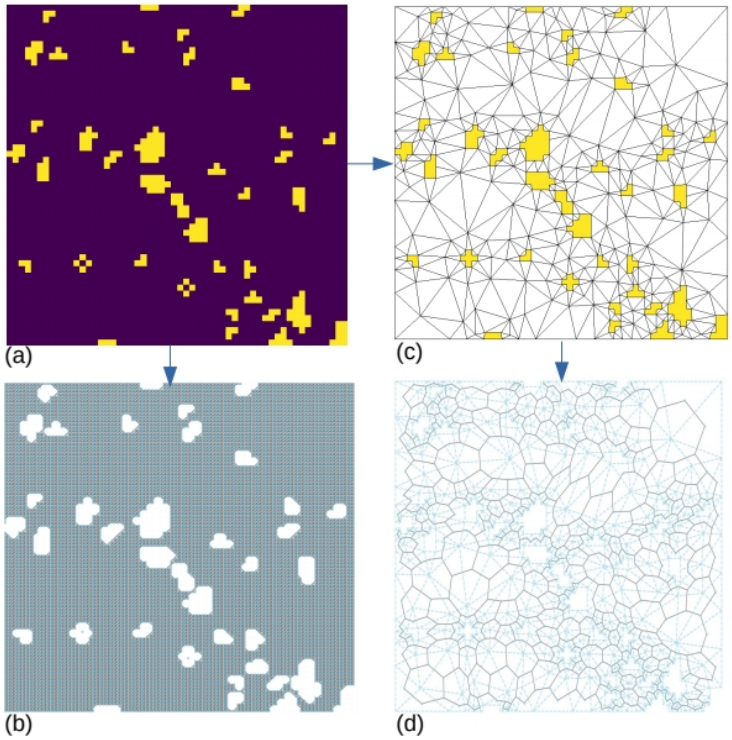


Fig.3. Generation of two types of graphs from an environment: (a) A sample environment with random obstacles (yellow regions) within an open area, representing a single tile in a tiled map. (b) A grid-based graph illustrating the quadrangle (QUAD) representation. (c) A triangular (TRI) representation created by meshing the free area into triangles (STL format). (d) The dual graph of the triangular representation, enabling path planning on the TRI structure.

A tiled map containing separate graphs cannot be used directly for path planning due to the disconnection between its tiles. To address this, an approach is applied to connect neighboring graphs through their boundary nodes. For instance, as shown in Fig. 4-a, a grid graph on the left is connected to the dual graph of an STL-represented tile on the right via newly defined edges. In this approach, a connected graph is created (Fig. 4-b) with dark grey representing one tile and light grey representing its neighboring tile, connected through their boundary nodes.

Boundary nodes for the grid graph are located precisely on the boundary edge of the tile, while for the triangular representation, they are the nearest nodes to the edges. By sequentially connecting all tiles through their boundary nodes, a single graph is constructed, representing the entire environment.

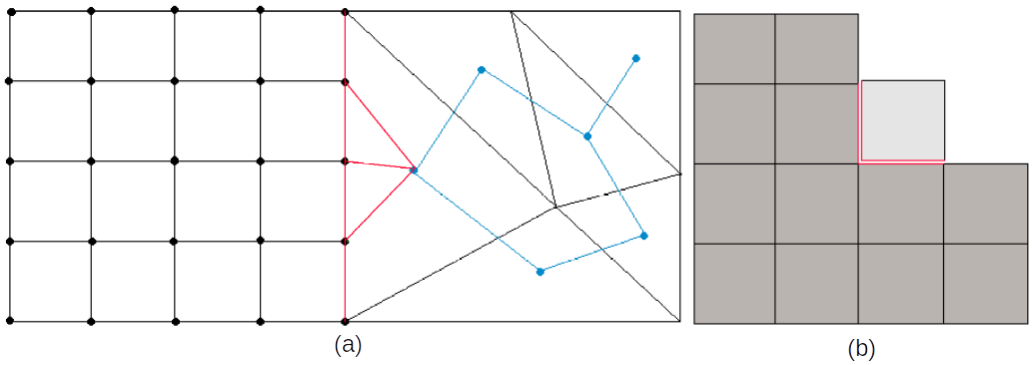


Fig.4. Approach to create a connected graph. a) A connected graph illustrating the integration of a grid graph and a dual graph of a triangular mesh via boundary nodes. b) Step-by-step approach for connecting tiles to form a unified graph.

This methodology, which divides an environment into tiles, applies a trained model to determine the optimal representation for each tile and ultimately merges these into a unified connected graph, is illustrated in Fig. 5. The resulting graph is ready for use with the A* graph search algorithm. Now, for identical scenarios, the results of navigation across various connected graphs can be compared to evaluate which configuration, based on the chosen tile division size, provides the most efficient solution.

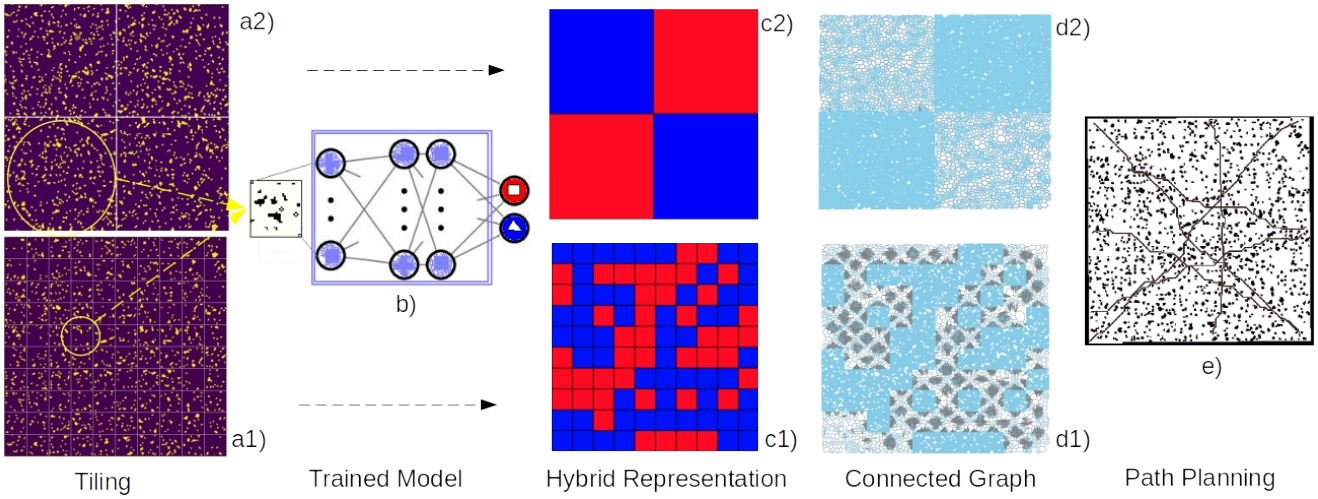


Fig.5. Pipeline of checking navigation scenarios on a Tiled Hybrid Representation Map. The first step, tiling, involves dividing an environment into tiles of varying sizes, shown by circles. a1) High division (small tile size). a2) Low division (large tile size). b) The trained model determines the best representation using a black-and-white input, with the output

being a label—square (QUAD) in red or triangle (TRI) in blue—for that tile representation. c1) and c2) Red-blue templates of the environment based on all results. d1) and d2) Connected graphs generated from the templates, shown by blue and grey elements. Red tiles, corresponding to a dense cell-grid representation, appear as condensed blue regions, while lower-density regions represent the dual graph of TRI representation. Grey edges connect separate graphs based on their proximity. e) Same scenarios are tested on the graphs generated from different templates, reflecting the results of identifying the best representation for each tile.

To analyze the results and identify an optimal range for tile division, a metric called the Participation Index has been introduced. This metric quantifies the balance between the usage of triangular and quadrilateral representations for a given tile division:

$$ParticipationIndex = \frac{N_{TR} - N_{QUAD}}{N_{TR} + N_{QUAD}}$$

Here, N_{TR} represents the number of triangular tiles, and N_{QUAD} represents the number of quadrilateral tiles. The total number of tiles in a tiled map is the sum of these two values, as indicated in the denominator.

A value of zero for the Participation Index indicates perfect balance, meaning equal usage of both types of tiles. A positive value approaching 1 indicates the dominance of the triangular (TRI) representation, while a negative value near -1 indicates that the quadrilateral (QUAD) representation is predominant.

This metric allows for an insightful analysis of the behavior of a tiled map in response to varying tile sizes. A near zero participation index reflects a greater balance between the two representations, while a higher absolute value indicates a larger disparity in their usage.

Defining this metric could help evaluate the hypothesis that a tiled hybrid representation map, incorporating both TRI and QUAD representations, enhances global path planning in large, cluttered environments. By comparing metric values across different tile divisions in fixed-scenarios path planning, it may be possible to determine whether a high balance between TRI and QUAD representations corresponds to the highest efficiency for path planning in such environments.

In addition, range and coefficient-of-variation (CV) metrics were computed to quantify the sensitivity of path length and CPU time to changes in the division parameter.

3 Simulation Analysis

3.1 Implementation

Each environment is represented as a 50 m × 50 m square, with an area equivalent to 25,000 times the robot's circular area (diameter 354 mm). Given this ratio, combined with obstacle distributions such as wide passages, narrow pathways, and blocked sections, the environment can be classified as a large map relative to the robot's size. Using OpenCV [36], the large image of environment is divided into smaller tiles. Each tile is represented either as a quadrangular grid or as a triangular mesh. For the triangular representation, free areas are converted into an STL triangular mesh format (Fig. 3-c) using libraries provided by OpenCV [36] and PyMesh [37]. Quadrangle representations, structured as grids, are often referred to as QUAD maps with a uniform cell size of 10 cm × 10 cm (Fig. 3-b). For example, an environment measuring 50 m × 50 m would result in a grid of 500 × 500 cells.

To determine the best representation for each tile, a pre-trained model based on a Convolutional Neural Network (CNN) [38] is used (Fig. 5-b). The CNN architecture, provided by MATLAB [39], was adapted to classify tiles into two categories: QUAD (0) and TRI (1). The input images for the CNN are resized to 28 × 28 pixels, and the model was trained on tiles representing areas of 5 m × 5 m as part of the previous work [40].

3.2 Simulation setup

Four large maps (Fig. 6), were analyzed to study the impact of varying tile sizes on navigation performance. These maps featured obstacles of diverse shapes and distributions, including small, medium, and large obstacles arranged randomly, systematically, or in mixed patterns. Each environment is divided into N tiles, where N=n×m, with n and m representing the number of divisions along the environment's width and height. For simplicity, the environments are square-shaped, so n=m, resulting in N=n × n tiles. The number of divisions (n) ranged from 2 to 20, with the largest tiles generated at n=2, creating four 25 m × 25 m tiles, and the smallest tiles at n=20, producing 400 tiles of 2.5 m × 2.5 m each.

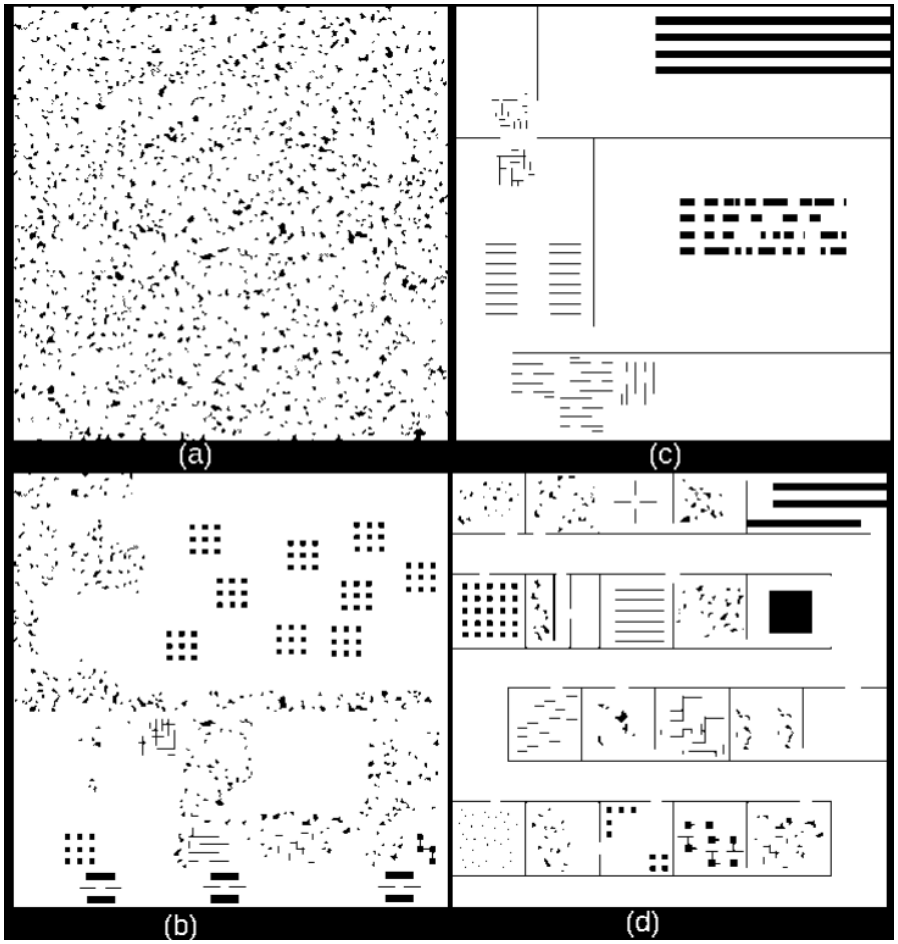
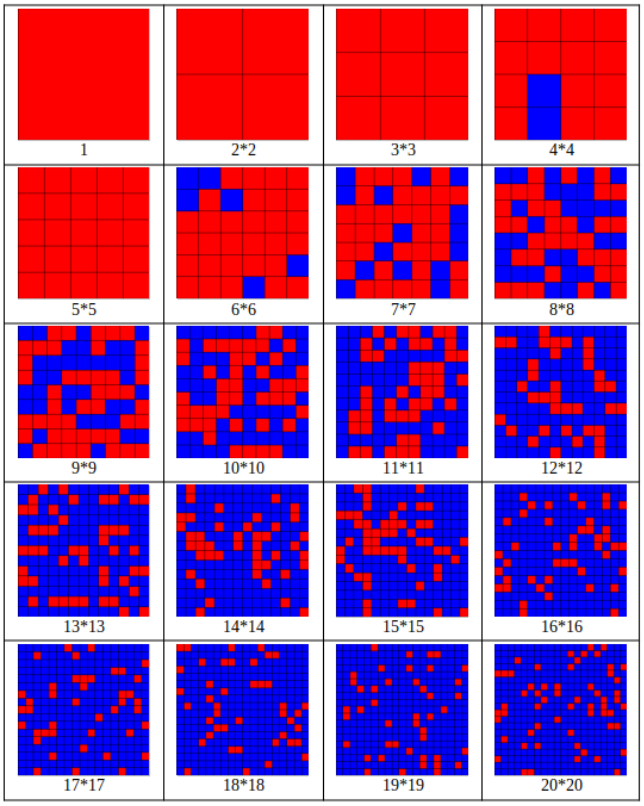


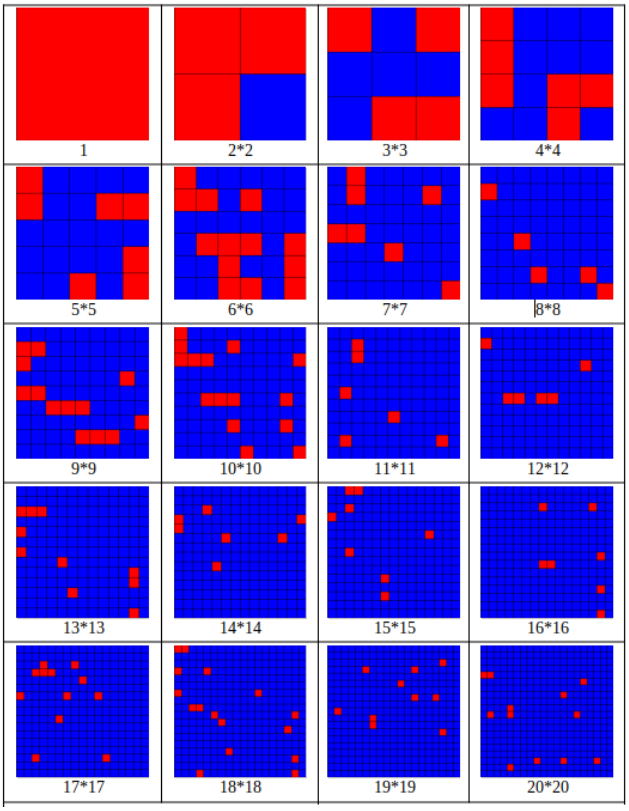
Fig.6. Four environments: a) Lmap1 - a cluttered, random environment. b) Lmap2 - a mixed environment combining random and structured areas. c) Lmap3 - a fully structured environment. d) Lmap4 - a room-by-room environment with both random and structured areas inside the rooms.

The pre-trained model determines the optimal representation—square or triangular—for the tiles in

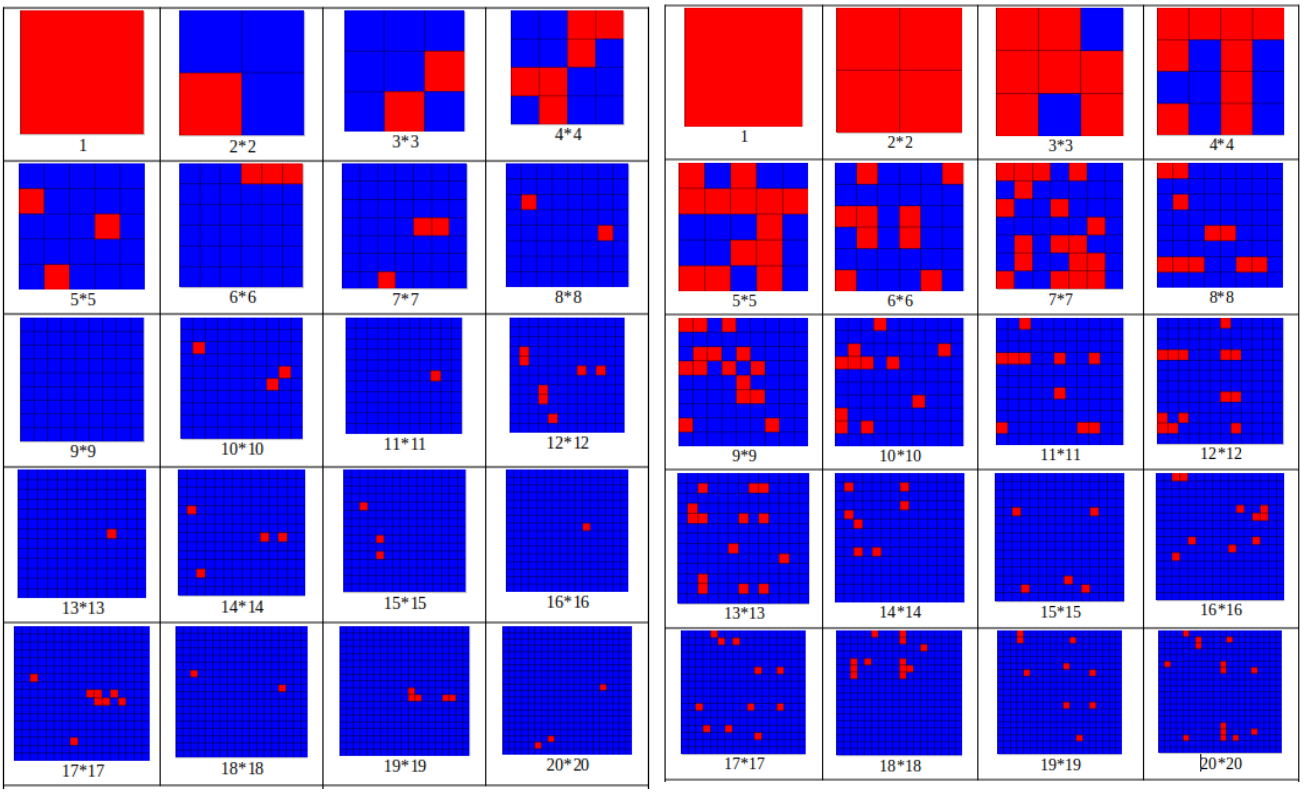
each configuration. Each tile's image is sent to the model for comparison and labeled as a quadrangle representation (red) or a triangular representation (blue). The distribution of TRI and QUAD tiles for all division levels is shown for the four environments (Fig. 7). These divisions serve as a template for generating a tiled hybrid map, helping to identify which configurations can provide more efficient paths by balancing path length and computation time in A* search. To evaluate navigation on each large map, several scenarios were defined, considering obstacle distributions, and ensuring full accessibility of the environment for testing. Navigation performance was evaluated using path length, computation time, and minimum distance to obstacles.



a) Lmap1



b) Lmap2

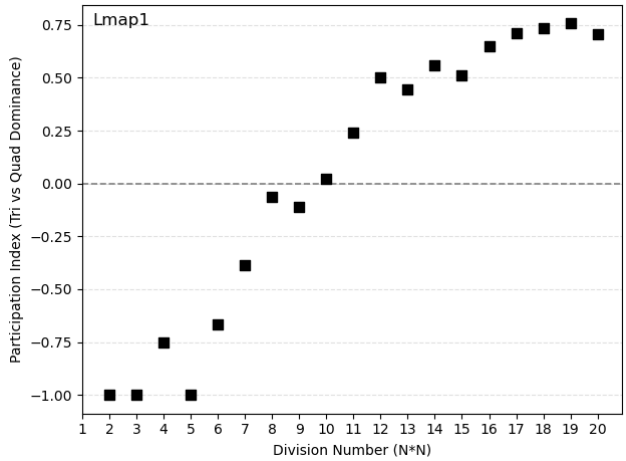


c) Lmap3 d) Lmap4

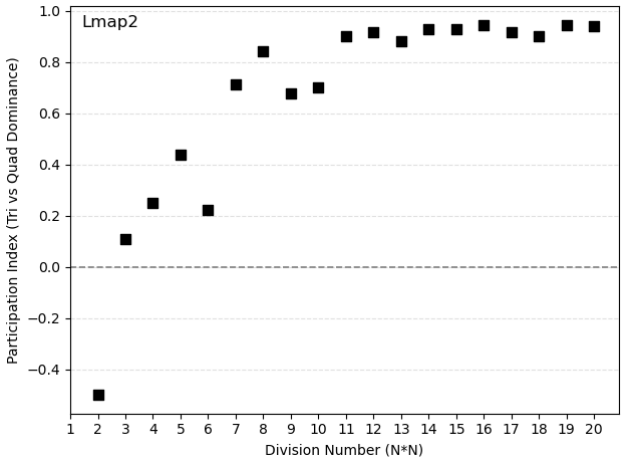
Fig. 7. Distribution of tiled hybrid maps, which are a combination of two representations: quadrangle (QUAD) and triangle (TRI), for different division levels (n) represented by colors. Each red square indicates a tile where QUAD is the optimal representation, while each blue square represents a tile where TRI is the preferred representation (n from 1 to 20). a) result for lmap1. b) result for lmap2. c) result for lmap3. d) result for lmap4.

3.3 Participation Index Analysis

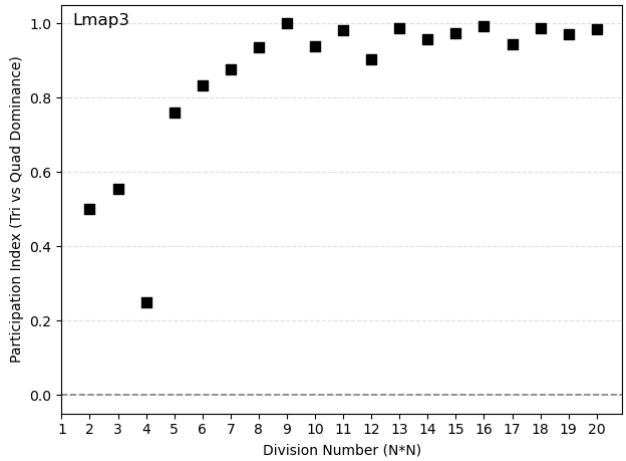
According to the defined metric participation index, Fig. 8 illustrates scatter plots that indicate the range with the highest diversity usage for rectangular and triangular representations. Each point shows the participation index for a division number in the map templates. While some points may appear as outliers in the scatter plot, they are valid data points representing natural variations in the dataset.



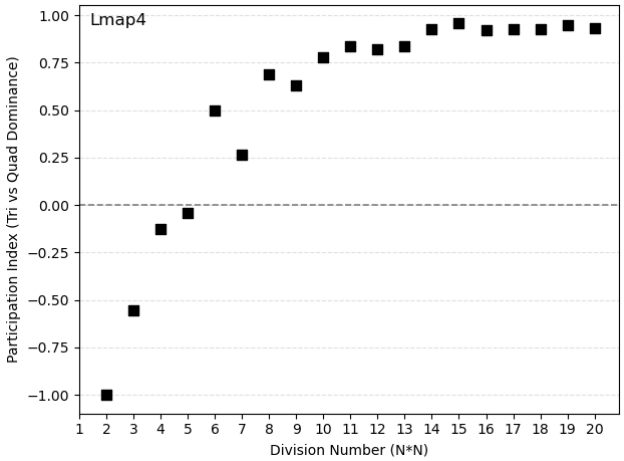
(a) For Lmap1, a random environment, lower division numbers demonstrate the predominance of QUAD representation, while higher division numbers show an increase in tiles where TRI representation is optimal. Divisions in the range of 8 to 10 exhibit the highest balance between QUAD and TRI, corresponding to a participation index near zero.



(b) For Lmap2, a mixed environment with both random and structured obstacles, the majority of the points lie in the TRI representation region. The balanced condition occurs between divisions 2 and 3, with division 3 having the lowest participation index.



(c) For the structured environment, Lmap3, the entire curve is located in the TRI section, indicating the predominance of TRI representation (or blue tiles in the template) for this type of environment. Division 4 exhibits the lowest participation index, representing the most balanced condition between QUAD and TRI. As mentioned, division 4 is a valid data point representing natural variations in the result.



(d) For Lmap4, a room-by-room environment, a similar behavior to the other environments is observed. Divisions 4 and 5 exhibit the highest balance between QUAD and TRI representations.

Fig. 8. Curves illustrating the participation index as a function of division number for the four Lmaps.

For instance, in a completely random environment (Lmap1), an 8 × 8 grid achieves the highest balance between quadrilateral and triangular representations. In contrast, a room-by-room environment (Lmap4) shows the highest participation index with a 5 × 5 grid, indicating a stronger preference for

one representation over the other. The participation indexes for Lmap2 through Lmap4 indicate a predominantly single representation, as evidenced by a line at the top of the curves. In contrast, Lmap1 does not exhibit this condition. This metric serves as an analysis tool for large environments without the need for conducting navigation.

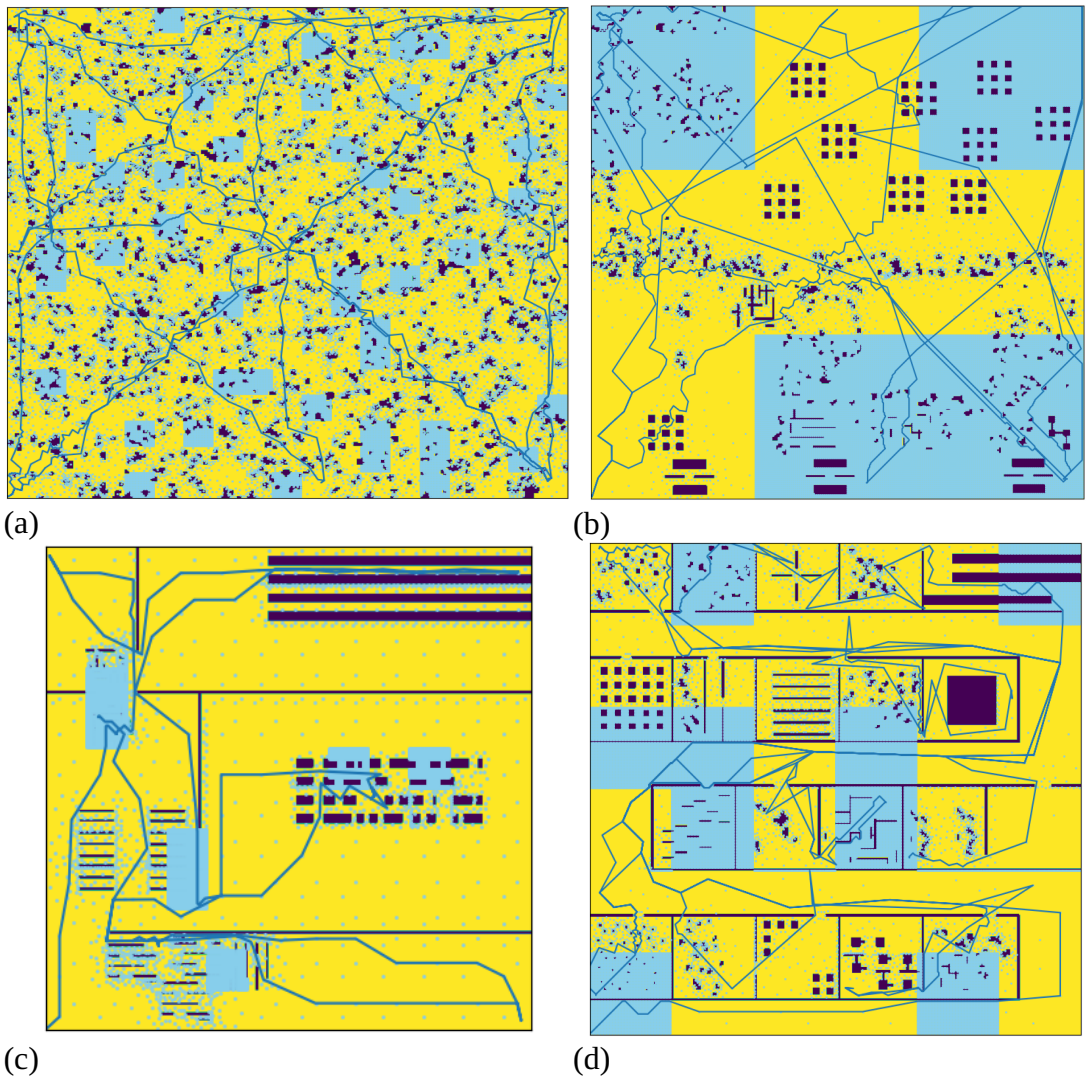


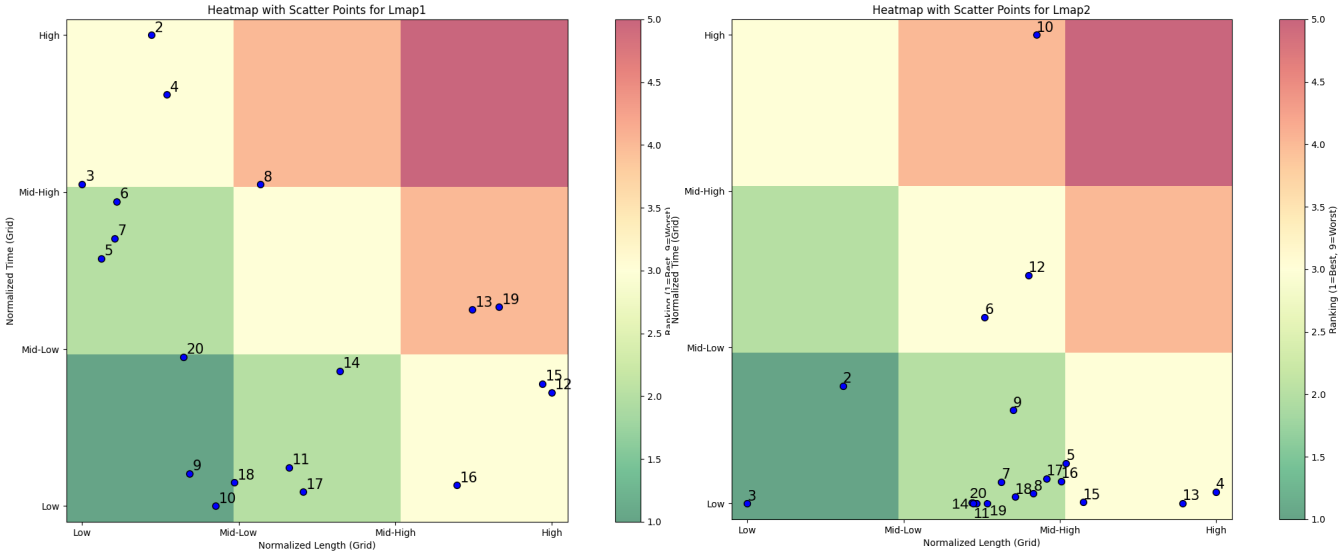
Fig. 9. Navigation simulations on the Lmaps for specific divisions. The blue points in the figures represent the graph nodes used in the corresponding area. Blue points on a yellow background indicate regions with TRI representation, while blue areas with densely packed nodes correspond to regions with QUAD representation.

3.4 Navigation Performance

For all Lmaps, fixed navigation scenarios have been defined for each environment based on the locations and distribution of obstacles to evaluate accessibility across different regions. The Fig. 9 illustrates samples of representative navigation results, where areas with a yellow background and blue points represent triangular mesh regions, with the blue points denoting the nodes of the corresponding dual graph. Areas with a blue background represent grid or quadrangle (grid-based) representations. Planned paths are shown as solid blue lines. Each figure illustrates a specific tile division of the environment;

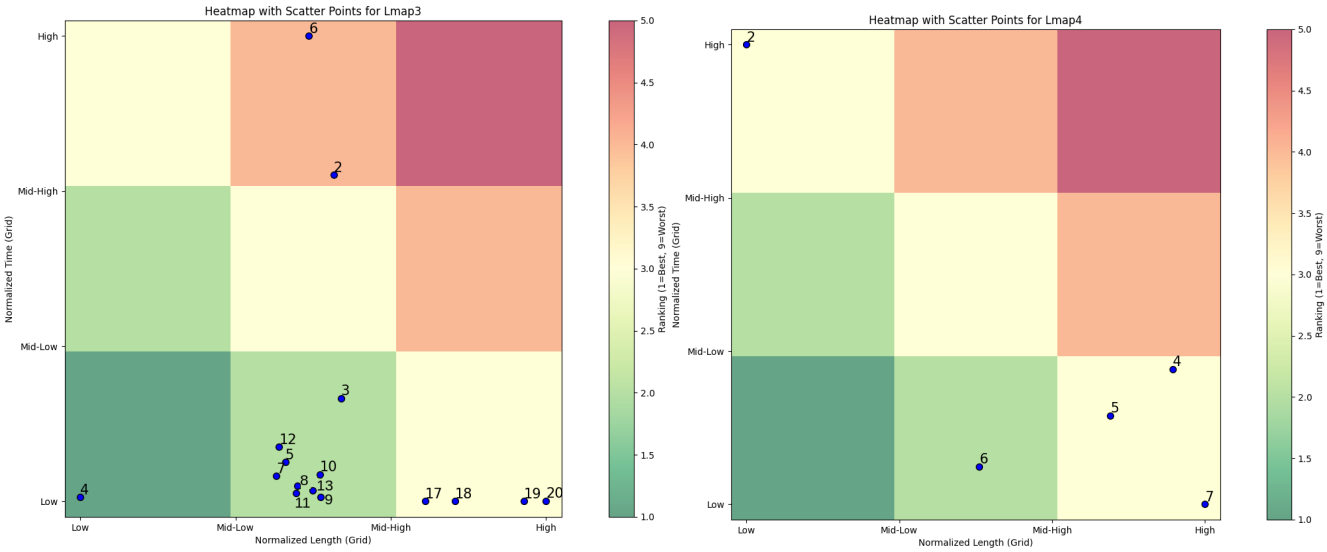
for example, (a) shows a 19×19 division of Lmap1, while (b) presents a 3×3 division of Lmap2, and so on.

After simulating a series of identical navigation scenarios for each environment with tile divisions ranging from n = 2 to n = 20, the normalized path length and computation time were plotted on a performance grid (Fig. 10). The grid highlights solution quality based on the combined trade-off between path length and computation time. Dark green regions indicate the most balanced and efficient solutions, while red regions correspond to less efficient solutions with high values for both metrics. Each blue point is labeled with its corresponding division number.



(a) Lmap1: In the dark green region, divisions 9 and 10 provide the best balance between time and length, aligning with the participation index. Lower division numbers generally yield better length results but with higher time values. Higher divisions typically offer better time results and are mostly situated in the mid-low region, except for divisions 12 and 18.

b) Lmap2: Division 3, located in the dark green region, demonstrates the best balance between time and length, closely aligning with the balanced line value in the participation index. A cluster of points in the lower section of the light green region indicates the shortest time results, aligning with the participation index prediction, where the triangular representation predominates in higher division numbers.



(c) Lmap3: Division 4 provides efficient results that align completely with the participation index. Additionally, group behavior is observed for higher divisions.

(d) Lmap4: Lmap4 is a special case with a limited number of runnable results. Due to implementation constraints, higher divisions could not be tested, and the analysis was limited to lower divisions covering individual room sizes and larger divisions representing multiple rooms together. Among these, Division 6, which is close to the size of a single room, provides the best results.

Fig. 10. Simulation results depicting the normalized path length and time for fixed scenarios across different divisions. Blue points, labeled with their corresponding division numbers, are plotted on a grid where the vertical axis represents normalized time from the analysis for each division, and the horizontal axis represents the normalized path length from the performed scenario. The quality of the results is categorized into four levels: Low, Mod-Low, Mid-High, and High. Dark green indicates the region with the shortest time and path length, transitioning to light green, yellow, orange, and red as the quality of the results decreases.

For Lmap1, divisions 9 and 10, located in the dark green region, offer the best balance between computation time and path length. Lower division numbers generally result in shorter paths at the cost of higher computation times, whereas higher division numbers tend to reduce computation time and mainly occupy the mid-to-low regions of the grid, with the exception of divisions 12 and 18. The alignment of divisions 9 and 10 with a participation index near zero indicates that a balanced combination of triangular and quadrangle representations yields the most efficient solution for this environment. This result confirms the effectiveness of the proposed tiled hybrid representation for Lmap1 and is consistent with the findings reported for division 10 in [40].

For Lmap2, division 3, situated in the dark green region, demonstrates the most balanced trade-off between computation time and path length, closely matching the balanced condition predicated by the participation index. A cluster of points in lower section of the light green region indicates the fastest computation times, which aligns with the participation index prediction that triangular representations dominate as higher division numbers in this environment.

In Lmap3, division 4 delivers highly efficient results that align closely with the participation index. Beyond this division, group behavior is observed, with triangular representations predominantly at higher divisions. Although division 4 appears as a local outlier, its performance confirms that even limited participation of quadrangular tiles can improve efficiency in otherwise highly structured environments.

Lmap4 represents a special case due to a limited number of available results. This limitation arises from difficulties in generating accurate triangular meshes when obstacle edges align exactly with tile boundaries. When wall and obstacle thicknesses are comparable, the triangular representation is generated reliably; however, when obstacle thickness exceeds wall thickness, smaller tile sizes lead to inaccurate triangular meshes. Consequently, the analysis is restricted to lower divisions representing individual rooms and larger divisions encompassing multiple rooms. To compensate for this limitation, a trend analysis is included.

Among these configurations, division 6, which corresponds approximately to the size of a single room, provides the best overall performance. Despite the limited dataset, the validity of the findings remains intact, as multiple data points around the balanced region are analyzed. As shown in Fig. 11, computation time exhibits a significant reduction with increasing division number, while a local minimum in path length is observed at division 6, where the computed path is only 4% longer than the shortest observed path. These results demonstrate the effectiveness of the tiled hybrid map in achieving a balanced solution and suggest that aligning tile size with room-scale structural features plays a key role in performance improvement.

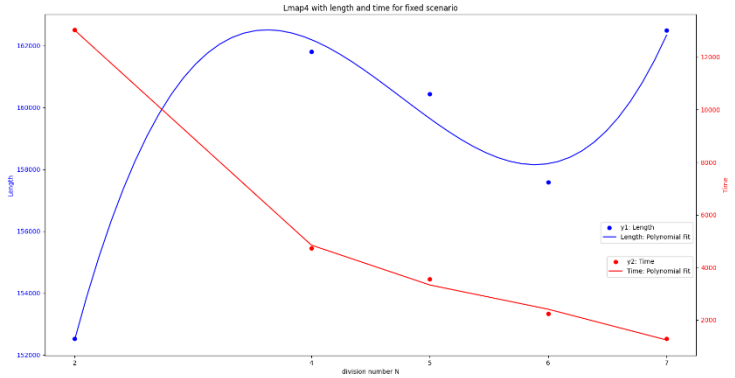
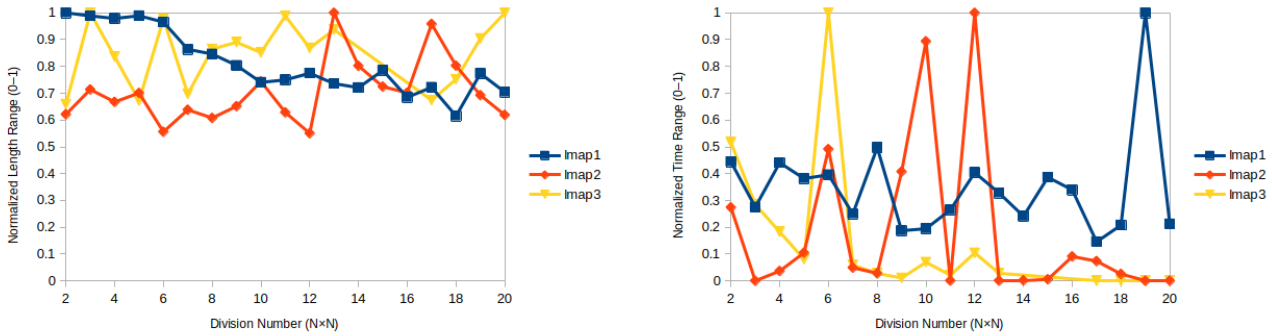


Fig. 11. The blue curve represents a trend of the average path length (measured in millimeters), while the red curve shows a trend for the time taken to find the solution (measured in CPU time). A noticeable reduction in time is observed as the number of divisions increases, with a local reduction in path length occurring at division 6.

3.5. Statistical Analysis of Division Effects

We apply statistical analysis to quantify how the division parameter influences path-planning performance. Two descriptive metrics—range and coefficient of variation (CV)—are used for both path length and CPU time. To compare sensitivity across LMAP1–LMAP3, the path-length range and CPU-time range were normalized per map, enabling direct comparison of variability patterns independent of absolute scale. For a given division, the length range is defined as the difference between the longest and shortest paths obtained across all scenarios on that map. Similarly, the time range is defined as the difference between the slowest and fastest computation times across scenarios.



(a) (b)

Fig. 12. Normalized range of path length and CPU time across scenarios for LMAP1–LMAP3 as the division size increases. Lower normalized values indicate more stable divisions, while higher values reflect greater sensitivity to scenario differences. (a) Normalized range of path length. (b) Normalized range of time.

These normalized curves illustrate how each division responds to scenario differences (Figure 12). To further quantify the influence of division on planning performance, we compute the coefficient of variation (CV) across all division levels (2×2 to 20×20) for each scenario, separately for path length and computation time. Because each scenario is fixed, the scenario-wise CV captures the relative variation induced solely by changes in division. The scenario-level CV values are then averaged to obtain a single map-level metric, termed Mean Division Sensitivity, which reflects the overall sensitivity of the planning results to the division parameter. The results are summarized in Table 1.

Table 1. Map-Level Average CV Values for Path Length and CPU Time Across Divisions

Metric	Lmap1	Lmap2	Lmap3
Average CV of Path Length	0.042	0.079	0.142
Average CV of CPU Time	0.377	2.79	2.338

The results indicate that lmap1 has the lowest variability (CV = 0.042 for length and 0.377 for time), suggesting minimal sensitivity to division changes. In contrast, lmap2 and lmap3 exhibit higher CV values, particularly for CPU time (2.79 and 2.338, respectively), indicating that their performance is more strongly affected by the choice of division. Overall, these results demonstrate that the impact of division is map-dependent. In addition, the significantly higher CV values for CPU time compared to path length indicate that division has a more pronounced effect on computational performance than on path optimality. This observation is consistent with existing literature, where improvements in navigation methods are often more reflected in computational efficiency than in path length optimization.

In addition to exhibiting the lowest variability in both path length and CPU time (Table 1), LMAP1 also shows smoother and smaller range fluctuations across divisions, as illustrated in Fig. 12. This consistent and gradual behavior indicates that the influence of division on LMAP1 is more structured and less affected by irregular scenario-specific responses compared to the other maps. For this reason, LMAP1 is selected for detailed examination, as it provides a clearer and more reliable representation of how division impacts path-planning performance, as shown in Fig. 13.

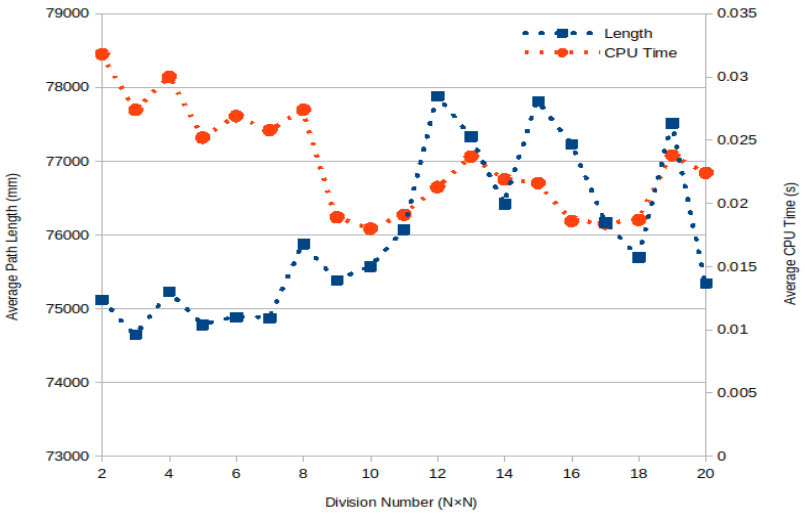


Fig. 13. Average path length and CPU time versus division number (N×N) for lmap1, illustrating the impact of division on performance.

Fig. 13 shows how the average path length and average CPU time vary with division number for LMAP1. The path-length curve does not follow a monotonic trend; instead, it begins to fluctuate noticeably beyond the mid-range divisions, suggesting that low-to-mid division levels tend to produce more efficient path length. In contrast, CPU time generally decreases as the division becomes finer, although a slight increase appears again after the mid-range. Together, these behaviors indicate the presence of a potential mid-range division level that offers a practical trade-off between path length and computation time.

The results demonstrate that the tiled hybrid representation map achieves efficient performance in both path length and CPU time. The proposed participation index effectively captures the balance between TRI and QUAD representations and enables identification of the tile size and template that yield maximum efficiency. The analysis also shows that performance is map-dependent, with division exerting a more pronounced influence on CPU time than on path length across all maps. Moreover, in larger maps with relatively uniform random structures—where scenario dependency is lower—mid-range tile sizes provide a favorable balance between path quality and computational cost. In contrast, for more structured or hybrid environments, optimal performance occurs only at specific division levels rather than across a broad range, and these points can be reliably identified using the participation index.

4 Conclusion

Efficient path planning in large-scale environments presents a critical challenge for autonomous navigation systems, where balancing path length, computational efficiency, and obstacle clearance remains essential. This study proposes a tiled hybrid map integrating triangular and quadrangular grid representations to address these challenges. Dividing large environments into tiles, each assigned the most efficient representation based on a trained model, enhances navigation performance while maintaining computational efficiency.

Findings demonstrate that the size of tiles plays crucial role in achieving optimal results. Incorporating statistical analysis using normalized range and coefficient-of-variation (CV) metrics shows that the

influence of division is map-dependent, with CPU time being more sensitive to division changes than path length. Three out of four tested environments confirm that a high participation between triangular and quadrangular representations provides the most efficient paths. This aligns with the principle that leveraging both representations allows adaptability to varied environmental features, such as narrow passages and open spaces. However, in one environment, characterized by room-by-room navigation, optimal tile size corresponds to individual room dimensions. This suggests that tailoring tile sizes to local environmental features yields better results than solely relying on high participation across entire map.

The proposed strategy works with different tile sizes but remains limited to fixed tile size division throughout the map. This restriction makes effectiveness somewhat dependent on environment's characteristics. Additionally, to clearly demonstrate the behavior of various cell shapes in response to A* search graph, any path smoothing is not applied intentionally, resulting in non-optimal paths in terms of energy consumption.

Future work includes enabling dynamic tile-size selection based on local environmental structure to further improve the adaptability of the tiled hybrid map. Extending the framework to continuously refine or combine TRI and QUAD representations would further enhance how the method adapts to different regions of the environment. In addition, expanding the training dataset to cover a broader range of tile sizes would improve the model’s accuracy.

References

[1] L. C. Santos, A. S. Aguiar, F. N. Santos, A. Valente, and M. Petry, “Occupancy grid and topological maps extraction from satellite images for path planning in agricultural robots,” Robotics, vol. 9, no. 4, p. 77, 2020.

[2] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, “Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges,” Sensors, vol. 18, no. 9, p. 3170, 2018.

[3] E. Tsardoulias, A. Iliakopoulou, A. Kargakos, and L. Petrou, “A review of global path planning methods for occupancy grid maps regardless of obstacle density,” Journal of Intelligent & Robotic Systems, vol. 84, no. 1, pp. 829–858, 2016.

[4] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, 1968.

[5] S. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” Iowa State University, 9811, 1998.

[6] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566–580, 1996.

[7] D. Harabor and A. Grastien, “Online graph pruning for pathfinding on grid maps,” in AAAI Conference on Artificial Intelligence, 2011.

[8] D. Harabor and A. Grastien, “Improving jump point search,” in International Conference on Automated Planning and Scheduling, 2014.

[9] S. Karaman and E. Frazzoli, “Incremental sampling-based algorithms for optimal motion planning,” in Robotics Science and Systems VI, 2010, pp. 267–274.

[10] R. Bohlin and L. E. Kavraki, “Path planning using lazy PRM,” in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation, 2000.

[11] F. Duchoň et al., “Path planning with modified a star algorithm for a mobile robot,” Procedia Engineering, vol. 96, pp. 59–69, 2014.

[12] I.-B. Jeong, S.-J. Lee, and J.-H. Kim, “Quick-RRT*: Triangular inequality-based implementation of RRT* with improved initial solution and convergence rate,” Expert Systems with Applications, vol. 123, pp. 82–90, 2019.

[13] I. Noreen, A. Khan, H. Ryu, N. L. Doh, and Z. Habib, “Optimal path planning in cluttered environment using RRT*-AB,” Intelligent Service Robotics, vol. 11, pp. 41–52, 2018.

[14] C. Zammit and E.-J. Van Kampen, “Comparison between A* and RRT algorithms for 3D UAV path planning,” Unmanned Systems, vol. 10, no. 2, pp. 129–146, 2022.

[15] T. Taha, J. V. Miro, and D. Liu, “An efficient path planner for large mobile platforms in cluttered environments,” in IEEE Conference on Robotics, Automation and Mechatronics, 2006.

[16] M. Shankar and G. Sushnigdha, “A Hybrid Path planning approach combining Artificial Potential Field and Particle Swarm Optimization for Mobile Robot,” IFAC-PapersOnLine, vol. 55, no. 22, pp. 242–247, 2022.

[17] N. Tomatis, “Hybrid, metric-topological representation for localization and mapping,” in Robotics and Cognitive Approaches to Spatial Mapping, Springer, 2008, pp. 43–63.

[18] S. Garrido, L. Moreno, and D. Blanco, “Exploration of a cluttered environment using voronoi transform and fast marching,” Robotics and Autonomous Systems, vol. 56, no. 12, pp. 1069–1081, 2008.

[19] Z. Tahir, A. H. Qureshi, Y. Ayaz, and R. Nawaz, “Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments,” Robotics and Autonomous Systems, vol. 108, pp. 13–27, 2018.

[20] P. Buschka, “An investigation of hybrid maps for mobile robots,” PhD Thesis, Örebro universitetsbibliotek, 2005.

[21] S. Thrun and A. Bücken, “Integrating grid-based and topological maps for mobile robot navigation,” in Proceedings of the National Conference on Artificial Intelligence, 1996.

[22] S. Thrun, “Learning metric-topological maps for indoor mobile robot navigation,” Artificial Intelligence, vol. 99, no. 1, pp. 21–71, 1998.

[23] M. Staniaszek et al., “AutoInspect: Towards Long-Term Autonomous Industrial Inspection,” arXiv preprint arXiv:2404.12785, 2024.

[24] J. Liu, X. Chen, J. Xiao, S. Lin, Z. Zheng, and H. Lu, “Hybrid Map-Based Path Planning for Robot Navigation in Unstructured Environments,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023.

[25] X. Zhong, J. Tian, H. Hu, and X. Peng, “Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment,” Journal of Intelligent & Robotic Systems, vol. 99, no. 1, pp. 65–77, 2020.

[26] J. Wang and Z. Chen, “A novel hybrid map based global path planning method,” in 3rd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), 2018.

[27] K. Song, W. Liu, G. Chen, X. Xu, and Z. Xiong, “FHT-Map: Feature-based Hybrid Topological Map for Relocalization and Path Planning,” IEEE Robotics and Automation Letters, 2024.

[28] J. I. Nieto, J. E. Guivant, and E. M. Nebot, “The hybrid metric maps (HYMMs): A novel map representation for DenseSLAM,” in IEEE International Conference on Robotics and Automation, 2004.

[29] C. Bibby and I. Reid, “A hybrid SLAM representation for dynamic marine environments,” in 2010 IEEE International Conference on Robotics and Automation, 2010.

[30] J.-C. Cuillière and V. Francois, “Integration of CAD, FEA and topology optimization through a unified topological model,” Computer-Aided Design and Applications, vol. 11, no. 5, pp. 493–508, 2014.

[31] J. T. Sample and E. Ioup, Tile-based geospatial information systems: principles and practices. Springer, 2010.

[32] Google, “Google Maps Tile API Overview.” [Online]. Available: https://developers.google.com/maps/documentation/tile/overview

[33] M. White, “Real-Time Optimally Adapting Meshes: Terrain Visualization in Games,” International Journal of Computer Games Technology, vol. 2008, no. 1, p. 753584, 2008.

[34] C. Mansley et al., “Robotic mapping and monitoring of data centers,” in IEEE International Conference on Robotics and Automation, 2011.

[35] A. Meysami, J.-C. Cuillière, V. François, and S. Kelouwani, “Investigating the impact of triangle and quadrangle mesh representations on AGV path planning for various indoor environments: with or without inflation,” Robotics, vol. 11, no. 2, p. 50, 2022.

[36] G. Bradski, The Opencv Library, vol. 25. 2000.

[37] Zhou Q., “PyMesh API Reference — PyMesh 0.2.1 documentation.” Accessed: Oct. 27, 2025. [Online]. Available: https://pymesh.readthedocs.io/en/latest/api.html

[38] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in 2017 International Conference on Engineering and Technology (ICET), Aug. 2017, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.

[39] The MathWorks, Inc., “MATLAB Version: R2023a.” [Online]. Available: https://www.mathworks.com

[40] A. Meysami, S. Kelouwani, J.-C. Cuilliere, V. Francois, A. Amamou, and B. Allani, “An efficient indoor large map global path planning for robot navigation,” Expert Systems with Applications, vol. 248, p. 123388, 2024.

4.6 Conclusion

This study addresses the challenge of efficient path planning in large-scale environments for autonomous navigation systems, focusing on balancing path length, computational efficiency, and obstacle clearance. The proposed solution is a tiled hybrid map that combines triangular and quadrangular grid representations. By dividing large environments into tiles and assigning the most appropriate representation to each tile based on a trained model, the method enhances navigation performance while maintaining computational efficiency.

The results demonstrate that tile size is crucial for achieving optimal performance. In three out of four tested environments, a combination that maximizes the use of both triangular and quadrangular representations produced the most efficient paths. This supports the notion that utilizing both representations provides greater adaptability to varying environmental features, such as narrow passages and open spaces. However, in one environment with room-by-room navigation, the best results were obtained by adjusting tile sizes to match the dimensions of individual rooms. Statistical analysis shows that division effects are map-dependent and influence CPU time much more than path length, reinforcing the need for adaptive tile-size selection. The trend toward adaptivity indicates that tailoring tile sizes to local environmental features is more effective than applying a uniform division across the entire map. A defined participation index further helps identify the optimal tile size for large maps without the need for simulation.

Chapter 5. Conclusion and Recommendations

5.1 Conclusion

Navigation in large, cluttered environments presents significant challenges for robotics due to limited computational resources and difficulties in achieving feasible efficiency. No single environmental representation offers an ideal solution for path planning. This thesis addresses these challenges by proposing a hybrid representation approach, combining two well-known methods: QUAD, commonly used in robotics, and TRI, widely adopted in FEM and the game industry. This approach involves developing a tiled map where each tile's representation is selected based on the most efficient option for path planning. To support this objective, the thesis removes the need for predefined environment-specific metrics and instead enables automated, scalable representation selection. For large maps, where manual processing is impractical, this method enables a systematic selection process based on environment analysis without requiring simulations.

The first step paid in chapter 2 involved bench-marking the two representations to determine which one demonstrates higher efficiency in path planning for specific environments. The methodology involves creating diverse indoor environments with varying obstacle distributions and constructing QUAD, TRI, and VITM mesh representations for each. Path planning is performed using the A* algorithm, incorporating metrics like path length, safety margins, and robot size (via an inflation layer) to evaluate the efficiency of each representation. According to analysis, in the scenarios without inflation, a coarse ITM was effective for most environments, with VITM outperforming in hybrid environments. However, with an inflation layer, robot size considerations shifted results significantly, making VITM and QUAD representations essential for narrower passages and complex multi-obstacle environments. This study demonstrates the importance of selecting the right representation for path planning, showing how efficiency varies with the environment. These findings establish the foundation for developing a hybrid map that incorporates inflation layers and performance-driven representation selection.

Chapter 3, as the core of the thesis, proposes a novel tiled hybrid representation map that demonstrates higher efficiency in path planning for large, cluttered environments. The study introduces a non-metric approach using a trained Convolutional Neural Network (CNN) to predict the best representation based on environment images. The results from both simulation and real-world experiments confirm and validate the effectiveness of this approach, showing significant reduction in path search complexity with only a minor overlength in the path compared to the optimal result. However, the main limitation in this work was selecting a specific tile size, which is addressed in the next chapter.

Chapter 4 introduces a tiled map with the capability to select tile size as a fundamental building block - similar to how resolution functions in an ordinary map. This approach optimizes the balance between computation time and path length by selecting tile sizes that maximize the contributions of both representations. While simulations confirmed these findings, the introduction of a contribution index eliminates the need for simulations in real-world environments. With this method, it becomes possible to determine the most efficient tiled map for a large environment using only its image. Additionally, by considering a connected graph of all tile types, this approach removes the need for the two-layer map previously defined in the second paper. To summarize the key elements of this thesis, Table 2 provides an overview of the three papers and their main characteristics.

Table 3: Summary of paper comparisons

Category	Paper 1	Paper 2	Paper 3
Core Idea	Relation between environment type and map representation	Hybrid representation based on environmental features	Tiled hybrid representation for large, diverse environments
Environment Type	Small environments with specific obstacles	Random and structured environments	Large, cluttered, structured, and hybrid environments
Representation Types	QUAD, IJM (TRI), and VXM	QUAD and IJM	QUAD and IJM
Methodology	Comparison of uniform quadrangle, constant irregular triangle, and variable IJM. Includes inflation layer and custom efficiency metric.	Use of CNN to select mesh type based on image features; hybrid two-layer map for path planning	Tile-based mapping of large environments, assigning optimal representation (QUAD or IJM) per tile
Input of Environment	Geometry	Image	Image
Implementation Tools	MAGIC, AutoCAD, C++	MAGIC, CV2, PYMESH, MATLAB, PYTHON, ROS	CV2, PYMESH, NETWORKX, PYTHON
Type of Analysis	Simulation	Simulation and physical experiments (ROS)	Simulation
Key Findings	Path planning efficiency varies with representation type	Hybrid maps improve path planning by balancing length, time, and safety	Hybrid tile-based representation enhances path planning efficiency in large, complex environments
Conclusion	IJM is better for open	Hybrid non-	Combining IJM and

	spaces; QUAD suits narrow passages	conformal maps reduce computation with only minor path length increase	QUAD in tiles leads to optimal results across varied map regions
Evaluation Metrics	Path length, computation time, safety	Path length, computation time, safety	Path length, computation time, safety
Contribution	Benchmarking of QUAD, IJM, and VXM; path planning metric definition; analysis of inflation layer	CNN-based model to choose best representation; study of mesh conformity; physical validation	Adaptive tile-based mapping framework; factor definition for optimal template selection per environment
Path Planning Method	A*	A* on dual-layer hybrid map	A* on integrated hybrid graph
Limitation	Limited to small environments	Evaluation based on a single tile size	Uniform tile size applied across the map

Together, the three papers demonstrate a clear progression from benchmarking individual representations, to developing a hybrid mapping strategy, to optimizing tile size for large-scale environments, fully addressing the thesis objectives. Across this progression, the thesis establishes that no single representation is universally efficient; instead, adaptive selection between TRI and QUAD yields consistently superior performance in large, cluttered indoor spaces. The benchmarking results highlight the critical role of the inflation layer and show how representation efficiency depends on obstacle distribution and robot size. The hybrid tiled map, supported by a CNN-based selector, reduces search complexity while maintaining near-optimal path quality, confirming the practicality of automated representation assignment. The final stage introduces adaptive tile-size selection and the contribution index, demonstrating that tile size strongly influences computational cost and that maximizing participation between TRI and QUAD provides the best balance between path length and CPU time. Statistical analysis further shows that division sensitivity is map-dependent, with CPU time being more affected than path length, reinforcing the need for adaptive tiling rather than uniform partitioning. Collectively, these findings establish a scalable, and representation-aware framework for efficient global path planning in large indoor environments.

5.2 Contributions

The primary contribution of this thesis is the development of an adaptive tiled hybrid mapping framework that improves global path-planning efficiency in large, static, and cluttered indoor

environments by dynamically selecting between triangular and quadrangular representations based on local environmental characteristics.

Building on this main contribution, the thesis offers three key sub-contributions:

1. Benchmarking of Basic Representations:

   A comprehensive evaluation of triangular and quadrangular map structures, including the effect of obstacle inflation, revealing their environment-dependent behaviour and demonstrating the sensitivity and failure of triangular meshes near obstacles.

   This directly supports Objective 1 by establishing how different representations behave across varying indoor environments.

2. Hybrid Tiled Representation

   A validated tiled hybrid mapping method that assigns TRI or QUAD to each region using automated tile analysis, enabling adaptive representation across diverse spatial patterns.
   This addresses Objective 2 by developing and validating the adaptive hybrid mapping strategy.

3. Impact of Tile Size

   A systematic investigation of tile partitioning size and its influence on planning efficiency, safety, and computational cost, resulting in practical guidelines for selecting optimal tiling strategies.

   This fulfills Objective 3 by quantifying how tile size affects performance and by defining optimal tiling parameters.

Beyond these core contributions, this thesis also introduces several developments that enable the hybrid framework. These include an automated tile-classification mechanism that evaluates local obstacle patterns without manual intervention; an analysis of mesh conformity to understand how boundary alignment affects computational performance; and extensive simulation and ROS-based experimental validation demonstrating real-time feasibility, improved path quality, and scalability for industrial navigation systems.

Despite its contributions, this research has certain limitations:

- Incorporating uncertainty into this approach requires additional mechanisms for handling uncertainty in triangular meshes, which are typically used as explicit representations.

- Path smoothing was intentionally not applied, which leads to non-optimal paths in terms of energy consumption.
- Dynamic obstacles were not modelled, meaning the best TRI–QUAD representation can change in real environments where obstacle motion alters local geometry.
- The model was trained on a specific tile size and used a scaling method for other sizes, which may not provide precise results for all cases.
- The study assumed a circular robot, which may reduce efficiency when applied to non-circular robots.
- All training maps and large-scale environments used in this work are synthetic, and therefore differ from real indoor environments; this gap — arising from sensing noise, surface irregularities, and construction tolerances — may affect real-world performance.

However, these limitations do not undermine the results but may restrict their applicability to broader navigation scenarios. In the next section, future recommendations are discussed to address these limitations and explore potential improvements.

5.3 Recommendations for future directions

The step-by-step progression of results across the three papers highlights an increasing level of efficiency, suggesting that even more flexible and localized applications of hybrid representations could further enhance path planning. This presents a promising direction for future research.

5.3.1 Triangle navigation approach

Using the center of a triangle as a representative for the element area may not be a reliable approach in navigation meshes or triangle discretization. When moving from the center of a triangle to a shared edge or common vertex with neighboring triangles, a precise check is needed to avoid collisions with surrounding obstacles. This can increase the computational time required to find a safe path, though it can be beneficial for shorter distances. Alternative approaches, such as considering the midpoints of triangle edges and moving within each triangle, offer a safer solution but result in longer paths. Other methods that use shared edge neighbors to identify a safe route and then employ a tunnel algorithm to find the shortest path through optimal triangle channels can also be considered. However, these methods may heavily depend on triangle size for producing close-to-optimal solutions. It is recommended to compare these different approaches in order to identify the best compromise between path length, time, and safety.

5.3.2 Incorporating Path Smoothing

In this work, no path smoothing approach was considered after finding a path using A*. While this approach offers several benefits, such as allowing comparison of A*'s raw behavior on each representation, simplifying post-processing of the result, and reducing the time required to find a path, it may not be energy efficient due to the sharp turns in the path. As a future recommendation, implementing a post-processing approach for path smoothing in both representations would result in a more energy-efficient path, which is crucial when multiple robots are deployed in a factory setting.

5.3.3 Adaptive Tile Sizing

Although our strategy works with different tile sizes, it is limited to a fixed, uniform tile division without flexibility for local adjustments. This is due to the simplicity and popularity of square divisions in tiling the environment, which makes the working algorithms simpler. This restriction makes its effectiveness somewhat dependent on the environment's characteristics. For a uniform random or uniformly structured environment, it has minimal impact. However, for a non-uniform environment composed of different areas with varying obstacle densities, empty spaces, and structured regions, it reduces the benefit of the hybrid representation, as the environment is divided into uniform tiles rather than more controlled ones. As a future development, hierarchical or adaptive subdivision techniques, such as those inspired by quad-tree decomposition, which dynamically selecting flexible tile sizes based on the characteristics of specific sub-regions within the environment could be explored (Figure 31). Although this would introduce more complexity compared to uniform tile sizes, it would enhance the overall efficiency of the tiled hybrid map. In contrast to a strict quad-tree hierarchy, the proposed adaptive subdivision should adjust tile sizes in proportion to spatial variation, while avoiding the full recursive structure that increases computational overhead.

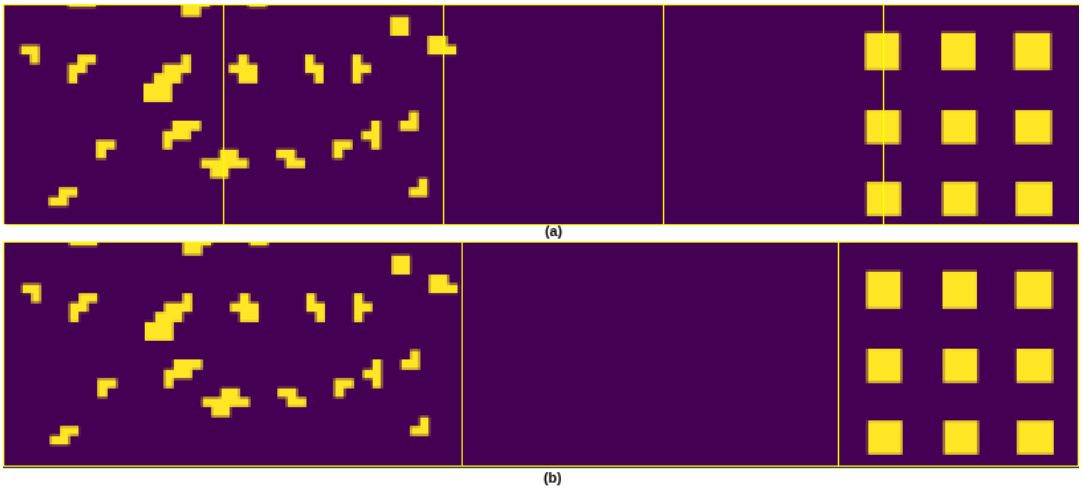


Figure 31: Comparison between hierarchical uniform tile division (a) and adaptive tile subdivision (b) a non-uniform tiling similar in concept to a quad-tree subdivision, where tile size varies according to obstacle density, though without full recursive hierarchy.

5.3.4 Comparing a variable irregular triangular mesh with the proposed hybrid approach

In the first paper, we used a Variable Irregular Triangular Mesh (VITM) alongside quadrilateral and nearly uniform Irregular Triangular Mesh (ITM) representations. For the later stages of the work, coarser triangle meshes resembling Constrained Delaunay Triangulation (CDT) were used, as supported by mesh generation tools like Pymesh. Although refinement strategies based on spatial characteristics—such as distance transform maps—were considered, implementing them within the MAGIC platform was not feasible due to its requirement for geometric input files, while our pipeline relied on image-based environment generation to produce thousands of training samples efficiently.

Even though some refinement capabilities were available through other tools, relying on finer triangulation alone in narrow regions does not guarantee safe or efficient paths and significantly increases computational cost. This insight led to the adoption of a tiled hybrid map, which combines the benefits of both triangular and quadrilateral representations, offering a more scalable and adaptable solution for large and diverse environments.

Nonetheless, a VITM-based strategy remains a valuable competitor to compare against the hybrid approach in future work. Although it requires substantial preprocessing, including mesh refinement and geometry preparation, these steps could also enable high-quality representations that make VITM a very promising direction for large-scale or learning-based applications.

5.3.5 Online re-classification of tiles under environmental change

The present tile-classification mechanism assigns TRI or QUAD based on static spatial features. However, this assumption does not fully reflect real-world conditions, where dynamic obstacles frequently alter the local structure of the environment. As obstacles move, appear, or disappear, the spatial configuration within a tile may change sufficiently to need switching to a different representation type. A promising direction for future work is to develop an online tile re-classification process that updates representation types whenever dynamic obstacles modify local geometry. Such a mechanism would allow the map to adapt continuously as regions become more or less cluttered, thereby improving robustness and enabling efficient planning in dynamic environments such as warehouses or hospitals.

5.3.6 Incorporating uncertainty and SLAM into the hybrid framework

A promising direction for future work is to extend the hybrid mapping framework so it can handle environmental uncertainty and work together with SLAM systems. The current approach assumes a known, static map and does not include uncertainty caused by sensor noise, partial observability, or the SLAM process. SLAM uncertainty refers to the accumulated errors in robot pose and map estimation caused by sensor noise, motion drift, data-association mistakes, and dynamic changes in the environment, all of which can distort the map. Introducing uncertainty into triangular meshes is difficult because they are explicit geometric representations and do not naturally support probabilistic updates. Future research could explore probabilistic mesh formulations. Probabilistic mesh formulations extend triangular meshes by attaching uncertainty models to vertices, edges, and faces, allowing the representation to update and stay consistent under sensor noise, SLAM drift, and partial observability. Such extensions would help the framework work reliably in unknown or partially known environments and improve robustness in real-world settings where mapping and localization are constantly changing.

Chapter 6. References

Aggarwal, R., & Kumar, M. (2020). Chance-Constrained Approach to Optimal Path Planning for Urban UAS. Paper presented at the AIAA Scitech 2020 Forum.

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. Paper presented at the 2017 international conference on engineering and technology (ICET).

Alajlan, M., & Koubˆaa, A. (2016). Writing global path planners plugins in ROS: A tutorial. Robot Operating System (ROS) The Complete Reference (Vol. 1,, 73–97.

Ammar, A., Bennaceur, H., Chaˆari, I., Koubˆaa, A., & Alajlan, M. (2016). Relaxed Dijkstra and A* with linear complexity for robot path planning problems in large-scale grid environments. Soft Computing, 20, 4149–4171.

Anderson, G. T., & Yang, G. (2007). A proposed measure of environmental complexity for robotic applications. Paper presented at the 2007 IEEE International Conference on Systems, Man and Cybernetics.

Bradski, G. (2000). The openCV library. Dr. Dobb's Journal: Software Tools for the Professional Programmer, 25(11), 120–123.

Brondani, J. R., de Lima Silva, L. A., Zacarias, E., & de Freitas, E. P. (2019). Pathfinding in hierarchical representation of large realistic virtual terrains for simulation systems. Expert Systems with Applications, 138, Article 112812.

Botsch, M.; Kobbelt, L.; Pauly, M.; Alliez, P.; Lévy, B. Polygon Mesh Processing; CRC Press: Boca Raton, FL, USA, 2010.

Cebollada, S., Pay´a, L., Flores, M., Peidro´, A., & Reinoso, O. (2021). A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. Expert Systems with Applications, 167, Article 114195.

Chagas, C., Zacarias, E., de Lima Silva, L. A., & de Freitas, E. P. (2022). Hierarchical and smoothed topographic path planning for large-scale virtual simulation environments. Expert Systems with Applications, 189, Article 116061.

Clearpath Robotics Inc. Turtlebot-2-Open-Source-Robot. Available online: https://clearpathrobotics.com/turtlebot-2-open- source-robot/ (accessed on 9 April 2022).

Cuillière, J.-C. (1998). An adaptive method for the automatic triangulation of 3D parametric surfaces. Comput.-Aided Des. 1998, 30, 139–149.

Cuillière, J.-C., & Francois, V. (2014). Integration of CAD, FEA and topology optimization through a unified topological model. Comput.-Aided Des. Appl. 2014, 11, 493–508.

Cuillière, J.-C., François, V., & Lacroix, R. (2016). A new approach to automatic and a priori mesh adaptation around circular holes for finite element analysis. Comput.-Aided Des. 2016, 77, 18–45.

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.

Ding, J., Zhou, Y., Huang, X., Song, K., Lu, S., & Wang, L. (2023). An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling. Journal of Computational Science, 67, Article 101937.

Do, D. T., & Lee, J. (2019). An automatically connected graph representation based on B-splines for structural topology optimization. Structural and Multidisciplinary Optimization, 59(6), 2023-2040.

Duchonˇ, F., Babinec, A., Kajan, M., Benˇo, P., Florek, M., Fico, T., & Juriˇsica, L. (2014). Path planning with modified a star algorithm for a mobile robot. Procedia Engineering, 96, 59–69.

Eykhoff, P. System Identification; Wiley: New York, NY, USA, 1974; Volume 14.

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.

Farin, G. (2002). *Curves and Surfaces for CAGD: A Practical Guide*. Morgan Kaufmann.

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.

Ferreira, J., Júnior, A. A., Galvaˇo, Y. M., Fernandes, B. J., & Barros, P. (2020). Cnn encoder to reduce the dimensionality of data image for motion planning. arXiv preprint arXiv:2004.05077.

Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, 4(1), 23–33.

Fragapane, G.; Ivanov, D.; Peron, M.; Sgarbossa, F.; Strandhagen, J.O. Increasing flexibility and productivity in Industry 4.0 pro-duction networks with autonomous mobile robots and smart intralogistics. Ann. Oper. Res. 2020, 1–19.

Francois, V.; Cuilliere, J.-C. 3D automatic remeshing applied to model modification. Comput.-Aided Des. 2000, 32, 433–444.

François, V.; Cuillière, J.-C. Automatic mesh pre-optimization based on the geometric discretization error. Adv. Eng. Softw. 2000, 31, 763–774.

Fujimura, K. Path planning with multiple objectives. IEEE Robot. Autom. Mag. 1996, 3, 33–38.

Garrido, S., Moreno, L., & Blanco, D. (2008). Exploration of a cluttered environment using voronoi transform and fast marching. Robotics and Autonomous Systems, 56(12), 1069-1081.

Gemme, S., Bakambu, J. N., & Rekleitis, I. (2005, May). 3D reconstruction of environments for planetary exploration. In The 2nd Canadian Conference on Computer and Robot Vision (CRV'05) (pp. 594-601). IEEE.

Geuzaine, C., & Remacle, J. F. (2009). Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. International journal for numerical methods in engineering, 79(11), 1309–1331.

Guimara˜es, R. L., de Oliveira, A. S., Fabro, J. A., Becker, T., & Brenner, V. A. (2016). ROS navigation: Concepts and tutorial. Robot Operating System (ROS) The Complete Reference (Volume, 1), 121–160.

Guivant, J., Nebot, E., Nieto, J., & Masson, F. (2004). Navigation and mapping in large unstructured environments. The International Journal of Robotics Research, 23(4–5), 449–472.

Harabor, D., & Grastien, A. (2011). Online graph pruning for pathfinding on grid maps. Paper presented at the Proceedings of the AAAI conference on artificial intelligence.

Harabor, D., & Grastien, A. (2014). Improving jump point search. Paper presented at the Proceedings of the International Conference on Automated Planning and Scheduling.

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.

Hwang, Y. K., & Ahuja, N. (1992). A potential field approach to path planning. IEEE transactions on robotics and automation, 8(1), 23-32.

Held, J., Lampe, A., & Chatila, R. (2006). Linking mobile robot performances with the environment using system maps. Paper presented at the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Hershberger, J., & Snoeyink, J. (1994). Computing minimum length paths of a given homotopy class. Computational geometry, 4(2), 63-97.

Hidalgo-Paniagua, A., Vega-Rodríguez, M. A., & Ferruz, J. (2016). Applying the MOVNS (multi-objective variable neighborhood search) algorithm to solve the path planning problem in mobile robotics. Expert Systems with Applications, 58, 20–35.

Jeong, I. B., Lee, S. J., & Kim, J. H. (2019). Quick-RRT*: Triangular inequality-based implementation of RRT* with improved initial solution and convergence rate. Expert Systems with Applications, 123, 82–90.

Kallmann, M. (2005). Path planning in triangulations. Paper presented at the Proceedings of the IJCAI workshop on reasoning, representation, and learning in computer games.

Kallmann, M. Shortest Paths with Arbitrary Clearance from Navigation Meshes. In Symposium on Computer Animation; Euro- graphics Association: Madrid, Spain, 2010; pp. 159–168.

Karaman, S., & Frazzoli, E. (2010). Incremental sampling-based algorithms for optimal motion planning. Robotics Science and Systems VI, 104(2), 267-274.

Kavraki, L. E., Svestka, P., Latombe, J.-C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE transactions on robotics and automation, 12(4), 566-580.

Kobbelt, L. P., Bischoff, S., Botsch, M., Kähler, K., Rössl, C., Schneider, R., & Vorsatz, J. (2000). Geometric modeling based on polygonal meshes (Vol. 1): Max-Planck-Institut für Informatik Saarbrücken, Germany.

Koefoed-Hansen, A., & Brodal, G. S. (2012). Representations for path finding in planar environments. Citeseer,

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.

Lamini, C.; Benhlima, S.; Elbekri, A. Genetic algorithm based approach for autonomous mobile robot path planning. Procedia Comput. Sci. 2018, 127, 180–189.

Latombe, J.-C. Robot Motion Planning; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 124.

Lau, B.; Sprunk, C.; Burgard, W. Efficient grid-based spatial representations for robot navigation in dynamic environ-ments. Robot. Auton. Syst. 2013, 61, 1116–1130.

LaValle, S. (1998). Rapidly-exploring random trees: A new tool for path planning. Research Report 9811.

LaValle, S. M. (2006). Planning algorithms. Cambridge university press.

Liu, C., Zhang, G., Rong, Y., Shao, W., Meng, J., Li, G., & Huang, Y. (2023). Hybrid metric-feature mapping based on camera and Lidar sensor fusion. Measurement, 112411.

Liu, G. R., & Quek, S. S. (2013). The finite element method: a practical course. Butterworth-Heinemann.

Liu, J., Chen, X., Xiao, J., Lin, S., Zheng, Z., & Lu, H. (2023). Hybrid Map-Based Path Planning for Robot Navigation in Unstructured Environments. Paper presented at the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

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.

Liu, Y.; Jiang, Y. Robotic Path Planning Based on a Triangular Mesh Map. Int. J. Control. Autom. Syst. 2020, 18, 2658–2666.

Liu, L., Wang, X., Yang, X., Liu, H., Li, J., & Wang, P. (2023). Path planning techniques for mobile robots: Review and prospect. Expert Systems with Applications, 120254.

Lozano-Perez, T. (1983). Spatial planning: A configuration space approach. IEEE transactions on computers, 32(02), 108-120.

Malkin, P. K. (1990). LOGnets: A Hybrid Graph Spatial Representation for Robot Navigation.

Mansley, C., Connell, J., Isci, C., Lenchner, J., Kephart, J. O., McIntosh, S., & Schappert, M. (2011). Robotic mapping and monitoring of data centers. Paper presented at the 2011 IEEE

International Conference on Robotics and Automation.

Mark, d.B.; Otfried, C.; Marc, v.K.; Mark, O. Computational Geometry Algorithms and Applications; Spinger: Ber-lin/Heidelberg, Germany, 2008.

Martínez-Barberá, H.; Herrero-Pérez, D. Autonomous navigation of an automated guided vehicle in industrial environments. Robot. Comput.-Integr. Manuf. 2010, 26, 296–311.

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.

MathWorks, (2023). Deep Learning Example, “Training a Model from Scratch”, https://www.mathworks.com/solutions/deep-learning/examples/training-a-model-from-scratch.html

MathWorks, (2023). Deep Learning Toolbox: User’s Guide. Retrieved June 14, 2022, from https://www.mathworks.com/help/deeplearning/gs/create-simple-image -classification-network-using-deep-network-designer.html.

Meng, F., & Meng, M. Q. H. (2021). A survey of learning-based robot motion planning. IET Cyber-Systems and Robotics, 3(4), 302–314.

Meysami, A., Cuilli`ere, J.-C., François, V., & Kelouwani, S. (2022). Investigating the impact of triangle and quadrangle mesh representations on AGV path planning for various indoor environments: With or without inflation. Robotics, 11(2), 50.

Meysami, A., Kelouwani, S., Cuilliere, J.-C., Francois, V., Amamou, A., & Allani, B. (2024). An efficient indoor large map global path planning for robot navigation. Expert Systems with Applications, 248, 123388.

Moravec, H., & Elfes, A. (1985). High resolution maps from wide angle sonar. Paper presented at the Proceedings. 1985 IEEE international conference on robotics and automation.

Nieto, J. I., Guivant, J. E., & Nebot, E. M. (2004). The hybrid metric maps (HYMMs): A novel map representation for DenseSLAM. Paper presented at the IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004.

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.

Nitsche, M., de Cristo´foris, P., Kulich, M., & Ko˘snar, K. (2011). Hybrid mapping for autonomous mobile robot exploration. Paper presented at the Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems.

Noreen, I., Khan, A., Ryu, H., Doh, N. L., & Habib, Z. (2018). Optimal path planning in cluttered environment using RRT*-AB. Intelligent Service Robotics, 11, 41-52.

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.

Otfried, C.; Marc, v.K.; Mark, O. Computational Geometry Algorithms and Applications; Spinger: Ber-lin/Heidelberg, Germany, 2008.

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.

Pauly, M.; Alliez, P.; Lévy, B. Polygon Mesh Processing; CRC Press: Boca Raton, FL, USA, 2010.

Pelechano, N.; Pettré, J.; Geraerts, R. Comparing navigation meshes: Theoretical analysis and practical metrics. Comput. Graph. 2020, 91, 52–82.

Peng, C.-C. Path smoothing techniques in robot navigation: State-of- the-art, current and future challenges. Sensors 2018, 18, 3170.

Perille, D., Truong, A., Xiao, X., & Stone, P. (2020). Benchmarking metric ground navigation. Paper presented at the 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR).

Petkovic, M.D. On the linear weighted sum method for multi-objective optimization. Facta Acta Univ. 2011, 26, 49–63.

Petrou, L. A review of global path planning methods for occupancy grid maps re-gardless of obstacle density. J. Intell. Robot. Syst. 2016, 84, 829–858.

Piegl, L., & Tiller, W. (1997). *The NURBS Book*. Springer.

Plaku, E., Kavraki, L. E., & Vardi, M. Y. (2008). Impact of workspace decompositions on discrete search leading continuous exploration (DSLX) motion planning. Paper presented at the 2008 IEEE International Conference on Robotics and Automation.

Qureshi, A. H., Ayaz, Y., & Nawaz, R. (2018). Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments. Robotics and Autonomous Systems, 108, 13-27.

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.

Ran˜o´, I., & Minguez, J. (2006). 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.

Raphael, B. A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. 1968, 4, 100–107.

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.

Ravankar, A. A., Ravankar, A., Emaru, T., & Kobayashi, Y. (2017, September). A hybrid topological mapping and navigation method for large area robot mapping. In 2017 56th annual conference of the Society of Instrument and Control Engineers of Japan (SICE) (pp. 1104-1107). IEEE.

Reid, I. A hybrid SLAM representation for dynamic marine environments. Paper presented at the 2010 IEEE International Conference on Robotics and Automation.

Reinoso, O. (2021). A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data. Expert Systems with Applications, 167, Article 114195.

Remacle, J. F. (2009). Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. International journal for numerical methods in engineering, 79(11), 1309–1331.

Rong, Y., Shao, W., Meng, J., Li, G., & Huang, Y. (2023). Hybrid metric-feature mapping based on camera and Lidar sensor fusion. Measurement, 112411.

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.

Rössl, C., Schneider, R., & Vorsatz, J. (2000). Geometric modeling based on polygonal meshes (Vol. 1): Max-Planck-Institut für Informatik Saarbrücken, Germany.

Sample, J. T., & Ioup, E. (2010). Tile-based geospatial information systems: principles and practices: Springer Science & Business Media.

Santos, F. N., Valente, A., & Petry, M. (2020). Occupancy grid and topological maps extraction from satellite images for path planning in agricultural robots. Robotics, 9(4), 77.

Santos, L. C., Santos, F. N., Valente, A., & Costa, P. (2022). Path Planning with Hybrid Maps for processing and memory usage optimisation. Paper presented at the 2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC).

Sartori, D., Ermacora, G., Pei, L., Zou, D., & Yu, W. (2020). A CNN approach to assess environment complexity for robotics autonomous navigation. Paper presented at the 2020 3rd International Conference on Mechatronics, Robotics and Automation (ICMRA).

Setiawan, Y. D., Pratama, P. S., Jeong, S. K., Duy, V. H., & Kim, S. B. (2014). Experimental comparison of A* and D* lite path planning algorithms for differential drive automated guided vehicle. In AETA 2013: Recent Advances in Electrical Engineering and Related Sciences (pp. 555-564). Berlin, Heidelberg: Springer Berlin Heidelberg.

Sgarbossa, F.; Strandhagen, J.O. Increasing flexibility and productivity in Industry 4.0 pro-duction networks with autonomous mobile robots and smart intralogistics. Ann. Oper. Res. 2020, 1–19.

Shankar, M., & Sushnigdha, G. (2022). A Hybrid Path planning approach combining Artificial Potential Field and Particle Swarm Optimization for Mobile Robot. IFAC-PapersOnLine, 55(22), 242-247.

Shirley, Peter, Michael Ashikhmin, and Steve Marschner. Fundamentals of computer graphics. AK Peters/CRC Press, 2009.

Snoeyink, J. (1994). Computing minimum length paths of a given homotopy class. Computational geometry, 4(2), 63-97.

Song, K., Liu, W., Chen, G., Xu, X., & Xiong, Z. (2024). FHT-Map: Feature-based Hybrid Topological Map for Relocalization and Path Planning. IEEE Robotics and Automation Letters.

Speray, D., & Kennon, S. (1990). Volume probes: Interactive data exploration on arbitrary grids. ACM SIGGRAPH Computer Graphics, 24(5), 5–12.

Sprunk, C.; Burgard, W. Efficient grid-based spatial representations for robot navigation in dynamic environ-ments. Robot. Auton. Syst. 2013, 61, 1116–1130.

Staniaszek, M., Flatscher, T., Rowell, J., Niu, H., Liu, W., You, Y., Hawes, N. (2024). AutoInspect: Towards Long-Term Autonomous Industrial Inspection. arXiv preprint arXiv:2404.12785.

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.

Stefanakis, E. (2017). Web mercator and raster tile maps: two cornerstones of online map service providers. Geomatica, 71(2), 100-109.

Stone, P. (2020). Benchmarking metric ground navigation. Paper presented at the 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR).

Strandhagen, J.O. Increasing flexibility and productivity in Industry 4.0 pro-duction networks with autonomous mobile robots and smart intralogistics. Ann. Oper. Res. 2020, 1–19.

Sushnigdha, G. (2022). A Hybrid Path planning approach combining Artificial Potential Field and Particle Swarm Optimization for Mobile Robot. IFAC-PapersOnLine, 55(22), 242-247.

Svishchev, N. (2025). Advanced embedded systems for autonomous robots control.

Svestka, P.; Latombe, J.-C.; Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE transactions on robotics and automation, 12(4), 566-580.

Taha, T., Miro, J. V., & Liu, D. (2006). An efficient path planner for large mobile platforms in cluttered environments. Paper presented at the 2006 IEEE Conference on Robotics, Automation and Mechatronics.

Tahir, Z., Qureshi, A. H., Ayaz, Y., & Nawaz, R. (2018). Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments. Robotics and Autonomous Systems, 108, 13-27.

Taylor, L., & Nitschke, G. (2018). Improving deep learning with generic data augmentation. Paper presented at the 2018 IEEE symposium series on computational intelligence (SSCI).

Thrun, S. (1997). The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, 4(1), 23–33.

Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence, 99(1), 21-71.

Thrun, S., & Bücken, A. (1996). Integrating grid-based and topological maps for mobile robot navigation. Paper presented at the Proceedings of the national conference on artificial intelligence.

Thrun, S. Robotic mapping: A survey. Explor. Artif. Intell. New Millenn. 2002, 1, 1.

Tian, J., Hu, H., & Peng, X. (2020). Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment. Journal of Intelligent & Robotic Systems, 99, 65-77.

Tomatis, N. (2008). Hybrid, metric-topological representation for localization and mapping. In Robotics and cognitive approaches to spatial mapping (pp. 43-63): Springer.

Torelli, J.C.; Bruno, O.M. 2D Euclidean distance transform algorithms: a comparative survey. ACM Comput. Surv. (CSUR) 2008, 40, 1–44.

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.

Truong, A., Xiao, X., & Stone, P. (2020). Benchmarking metric ground navigation. Paper presented at the 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR).

Tsardoulias, E., Iliakopoulou, A., Kargakos, A., & Petrou, L. (2016). A review of global path planning methods for occupancy grid maps regardless of obstacle density. Journal of Intelligent & Robotic Systems, 84(1), 829-858.

Valente, A., & Petry, M. (2020). Occupancy grid and topological maps extraction from satellite images for path planning in agricultural robots. Robotics, 9(4), 77.

Valente, A., & Costa, P. (2022). Path Planning with Hybrid Maps for processing and memory usage optimisation. Paper presented at the 2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC).

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.

Vardi, M. Y. (2008). Impact of workspace decompositions on discrete search leading continuous exploration (DSLX) motion planning. Paper presented at the 2008 IEEE International Conference on Robotics and Automation.

Vega-Rodríguez, M. A., & Ferruz, J. (2016). Applying the MOVNS (multi-objective variable neighborhood search) algorithm to solve the path planning problem in mobile robotics. Expert Systems with Applications, 58, 20–35.

Vorsatz, J. (2000). Geometric modeling based on polygonal meshes (Vol. 1): Max-Planck-Institut für Informatik Saarbrücken, Germany.

Wang, J., & Chen, Z. (2018). A novel hybrid map based global path planning method. Paper presented at the 2018 3rd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS).

Wang, J., Li, B., & Meng, M. Q. H. (2021). Kinematic Constrained Bi-directional RRT with Efficient Branch Pruning for robot path planning. Expert Systems with Applications, 170, Article 114541.

Wang, J., Zhang, T., Ma, N., Li, Z., Ma, H., Meng, F., & Meng, M. Q. H. (2021). A survey of learning-based robot motion planning. IET Cyber-Systems and Robotics, 3(4), 302–314.

Wang, L. (2023). An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling. Journal of Computational Science, 67, Article 101937.

Wang, P. (2023). Path planning techniques for mobile robots: Review and prospect. Expert Systems with Applications, 120254.

Wang, X., Yang, X., Liu, H., Li, J., & Wang, P. (2023). Path planning techniques for mobile robots: Review and prospect. Expert Systems with Applications, 120254.

Wang, Y., Zhai, Z., & Ma, X. (2023). Improved bidirectional RRT* algorithm for robot path planning. Sensors, 23(2), 1041.

Warnell, G., Dhamankar, G., & Stone, P. (2022). Appl: Adaptive planner parameter learning. Robotics and Autonomous Systems, 154, Article 104132.

White, M. (2008). Real-Time Optimally Adapting Meshes: Terrain Visualization in Games. International Journal of Computer Games Technology, 2008(1), 753584.

Withey, D., & Kuchwa-Dube, C. (2021). RRT based path planning for mobile robots on a 3D surface mesh. In 2021 Southern African universities power engineering conference (pp. 1-6). IEEE.

Wolfram, S. (1984). Cellular automata as models of complexity. Nature, 311(5985), 419-424.

Woods, A. C., & La, H. M. (2015, December). Dynamic target tracking and obstacle avoidance using a drone. In International Symposium on Visual Computing (pp. 857-866). Cham: Springer International Publishing.

Xiao, X., Wang, Z., Xu, Z., Liu, B., Warnell, G., Dhamankar, G., & Stone, P. (2022). Appl: Adaptive planner parameter learning. Robotics and Autonomous Systems, 154, Article 104132.

Xiao, X., Xu, Z., Liu, B., Dhamankar, G., & Stone, P. (2020). Benchmarking metric ground navigation. Paper presented at the 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR).

Xu, Z., Liu, B., Warnell, G., Dhamankar, G., & Stone, P. (2022). Appl: Adaptive planner parameter learning. Robotics and Autonomous Systems, 154, Article 104132.

Yang, G. (2007). A proposed measure of environmental complexity for robotic applications. Paper presented at the 2007 IEEE International Conference on Systems, Man and Cybernetics.

Yu, W. (2020). A CNN approach to assess environment complexity for robotics autonomous navigation. Paper presented at the 2020 3rd International Conference on Mechatronics, Robotics and Automation (ICMRA).

Yufka, A., & Parlaktuna, O. (2009, May). Performance comparison of bug algorithms for mobile robots. In Proceedings of the 5th international advanced technologies symposium, Karabuk, Turkey (pp. 13-15).

Zacarias, E., de Lima Silva, L. A., & de Freitas, E. P. (2019). Pathfinding in hierarchical representation of large realistic virtual terrains for simulation systems. Expert Systems with Applications, 138, Article 112812.

Zacarias, E., de Lima Silva, L. A., & de Freitas, E. P. (2022). Hierarchical and smoothed topographic path planning for large-scale virtual simulation environments. Expert Systems with Applications, 189, Article 116061.

Zammit, C., & Van Kampen, E. J. (2022). Comparison between A* and RRT algorithms for 3D UAV path planning. Unmanned Systems, 10(02), 129–146.

Zhang, G., Rong, Y., Shao, W., Meng, J., Li, G., & Huang, Y. (2023). Hybrid metric-feature mapping based on camera and Lidar sensor fusion. Measurement, 112411.

Zhang, T., Ma, N., Li, Z., Ma, H., Meng, F., & Meng, M. Q. H. (2021). A survey of learning-based robot motion planning. IET Cyber-Systems and Robotics, 3(4), 302–314.

Zhao, Y., Huang, Y. (2023). Hybrid metric-feature mapping based on camera and Lidar sensor fusion. Measurement, 112411.

Zheng, K. (2021). Ros navigation tuning guide. Robot Operating System (ROS). The Complete Reference (Vol. 6,, 197–226.

Zheng, Z., & Lu, H. (2023). Hybrid Map-Based Path Planning for Robot Navigation in Unstructured Environments. Paper presented at the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Zhong, X., Tian, J., Hu, H., & Peng, X. (2020). Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment. Journal of Intelligent & Robotic Systems, 99, 65–77.

Zhou, Q. (2019). PyMesh—Geometry processing library for Python. Software available for download at https://github. com/PyMesh/PyMesh.

Zhou, Y., Huang, X., Song, K., Lu, S., & Wang, L. (2023). An improved RRT* algorithm for robot path planning based on path expansion heuristic sampling. Journal of Computational Science, 67, Article 101937.

Zlatanovic, M.L.; Petkovic, M.D. On the linear weighted sum method for multi-objective optimization. Facta Acta Univ. 2011, 26, 49–63.

Zou, D., & Yu, W. (2020). A CNN approach to assess environment complexity for robotics autonomous navigation. Paper presented at the 2020 3rd International Conference on Mechatronics, Robotics and Automation (ICMRA).