American Journal of Electronics & Communication

Mohd Aaqib Lone, American Journal of Electronics & Communication, Vol. I (3), 1-7

A Survey on Robot Behavior and Distance Estimation in IndoorGML Maps Implementation

Mohd Aaqib Lone, Owais mujtaba khanday, Aadil Ganie Gani (University of Miskolc, Hungary)
mohdaaqiblone@gmail.com

Abstract— *The significance of behavior-based robots and the indoor navigation system has increased significantly over the past few years in its monitoring and wayfinding, taking into account the existence of robot behavior in various fields of social interest. The basic strategy of the overall behavior tasks in a behavior-based system are divided into smaller independent behaviors that focus on the performance of specific tasks such as the behavior of a robot for indoor environments. One behavior will concentrate on the wall, while the other focuses on avoiding obstacles, etc. Indoor navigation is the main concern since almost a lot of individuals spend more time in the indoor climate. In different fields such as hospitals, transportation, marketing, and military purposes, it becomes an essential thought. The main goal of this paper to give a survey of distance estimation in IndoorGML, behaviors system, and the questions of their robotic implementation possibilities. In this essay, we define robot activities and two basic approaches to distance estimation, such as combinatorial planning and sampling-based planning. These two planning strategies are the principles of motion planning. Combinatorial planning is used for finding the path over the continuous configuration space without restoring it to approximations. The most used concept in planning is sampling-based planning, it provides a successful solution in wayfinding path planning, and because of this, it is performed in different robotics fields. Therefore, for distance estimation in IndoorGML maps, we apply combinatorial and sampling-based approaches.*

Keywords: *IndoorGML, Combinatorial Planning, Sampling Based Planning, Probabilistic Roadmap Method, Rapidly-exploring Random Trees.*

1. Introduction

These days, navigation and way-finding are becoming essential tasks because people spend much of their time in the indoor setting where way-finding might be difficult. Navigation in a hospital, for instance, where each floor is equally arranged, and the atmosphere is rather monotonous, for inexperienced workers, wayfinding and guidance are difficult tasks. Widespread smartphones will allow us to provide reliable indoor navigation and way-finding services that are rarely used in indoor environments such as hospitals, airports, shopping centres, etc. For the indoor application of these devices, finding the shortest path to the destination and avoiding the barriers are necessary. Way-finding requires a comprehensive environment definition and a distance estimation function in the static empty indoor environment. Different formats are used to model indoor environments. To model their layout, fixed, and infrastructure, architects, use various models of the same building. These models are generally referred to as the Model of Construction Details. Besides, for indoor navigation purposes, floor plans are preferred and generally accepted. IndoorGML is a new standard used for navigation purposes to model indoor environments. Since IndoorGML only describes the building's structure, to provide way-finding services, a distance estimation feature should be found. Behaviour-based robotics is introduced for situated robots, enabling robots to adjust for the dynamics of real-world environments based on animal behaviour and also providing robots more computational capacity and expressivity. The behaviours are patterns of the activity of the robot that emerges from an external spectator's contact with the robot and its environment. According to a programmer, behaviours are the control modules that cluster constraint collections to achieve and sustain a goal. Each behaviour gathers inputs from sensors or/and from other device behaviours and gives outputs to other behaviours or the actuators of the robot. The behaviour-based controller system is, therefore, a structured network of interacting behaviours with no centralized world representation or control emphasis. In contrast, individual behaviours and behaviour networks maintain some state data and models. For example, we can characterize the dog's ethological robot behaviour such as missing, exploring and fear states, which reflect the conditions in the world. Miss state determines the level of stress of a dog when it is separate from the owner or caregiver, anxiety state explains the level of stress from the conditions of the elderly, and the level of enthusiasm to see in the room is displayed by exploring the state. A suitable robot behaviour is selected from the behavioural set, and ethological awareness of the social behaviour of dogs explains the selection rule and behavioural group, for example, there are several primary dog behaviours, such as dog explores the room and dog moves to the door that dogs display according to the situation. The dog investigates the actions of the room, which defines a dog investigation operation in which the dog watches around in a strange setting, and precisely after a dog enters the room, this form of behaviour can be determined. The dog moves towards the action of the door suggests a typical dog occurrence in which his owner or caregiver is missing from the dog and moves towards the door and stands/sits in front of the door. Different ideas are estimated to contribute to animal intelligence and adaptability such as concurrent activation, hierarchical organizations of behaviour, and coordination of motivational tendencies and individual behaviour excitation and inhibition via thresholds.

Distance approximation is vital for way-finding. Although IndoorGML defines the structure of the building, it does not contain functions for distance estimation. The IndoorGML model may contain distance between the objects, but it has to be set manually, which is time-consuming. Hence there is a need for an automatic distance estimation method for IndoorGML. IndoorGML is an Open Geospatial Consortium (OGC)[1] XML-based specification implemented for Indoor Environments. IndoorGML provides an important source of building documentation, it also gives users a context for location-based applications running and can be of benefit to computer systems where an understanding of the environment is required, IndoorGML purpose is to reflect and facilitate the exchange of geoinformation needed to construct an indoor navigation system. Some facts about indoor spaces are defined in the OGC standard, such as navigation constraints, subdivisions of space, geometric and semantic properties of spaces, and logical and metric navigation networks. IndoorGML combines additional features such as a geometric graph for navigation and a multilayer space model of indoor spaces that apply to indoor navigation with 3D specifications such as CityGML. IndoorGML uses the Geography Markup language (GML)[2] to encode geo-information and offers a context for the indoor navigation system to suggest a description of the indoor space.

2. IndoorGML

Indoor spaces are those spaces within one or multiple buildings which consist of architectural components such as rooms, doors, stairs, corridors, etc. IndoorGML can be defined in two parts, core data model and data navigation model. The core data model is used to describe the topological connectivity, geometry, and different contexts of indoor space. The data navigation model provides semantics for the navigation process (Lee et al [3]) and establishes a methodology to classify spaces and their indoor characteristics. Indoor spatial information is described into two categories. The first category describes the management of building components and indoor facilities, which are used to construct building components such as rooms and walls. The second category is the usage of indoor space, it is defined as usage and localization features in indoor space and represent space components such as rooms, corridors, and doors. Some basic concepts of IndoorGML are Cellular space model, Cell geometry, Topology between cells, and Cell semantics, multi-layered space model, and the last one is the Modular structure of IndoorGML. The IndoorGML uses the concepts of primal and dual space and automatic derivation of dual space that is part of IndoorGML (Diakite et al. [4]). The space subdivision is defined into two significant features such as primal space and dual space. Primal Space describes the control of the indoor space that influences the representation of indoor cells and the dual space describes the information of the norm to support the automation of the space subdivision process. The classification of IndoorGML can be represented in 2D, 3D, and the link between indoor and outdoor. (Kim and Lee, [5]) have proposed an approach called semi-automatic, to create IndoorGML data from images, and within the same direction, Mirvahabi and Abbaspour have introduced an automated method to extract IndoorGML data from OpenStreetMap [6]. The purpose of IndoorGML is to follow the requirements for indoor navigation by supporting the common framework for determining the spatial models for indoor navigation. The extensible markup language design for IndoorGML core module consists of four basic types such as CellSpace, CellSpaceBoundary, State and Transition. The First two CellSpace and CellSpaceBoundary are the key units of the primal space of IndoorGML, while as the last two State and Transition belongs to the dual space of IndoorGML.

Primal Space describes the control of the indoor space that influences the representation of indoor cells and the main components of primal space of IndoorGML are CellSpace and CellSpaceBoundary [7]. CellSpace is used to determine the main types of the cellular indoor spatial model, such as room, corridor, and hall, and it also includes a GML identifier (an identifier which is assigned to an object by the maintaining authority which is used in references to the object), this identifier is usually unique either globally or inside an application domain with its attributes. The cell geometry for primal space is described by the International Organization for Standardization (ISO 19107 [8]), which can be either surface or solid depending upon the dimensionality of space object either 2D or 3D. The CellSpaceBoundary describes the geometric edges of a CellSpace object, and its geometry may be either surface or curve depending on the dimensionality. Different options are employed to define the geometry of CellSpaceBoundary, the first one is to include only topological relationships between cells. The second is to present the geometry within IndoorGML data by geometric types defined in ISO 19107. The last one requires external references to the object in another dataset that has geometric data. The location class of Primal Space can be described based on some different concepts such as construction, functional area, and security, for building and walking and driving for the user. Each specific location area can be formulated in a space layer, i.e., a multi-layered space model, in which spaces from different types of layers may overlap and no overlapping is allowed between the cells in the same space layer.

Dual space describes the information of the norm to support the automation of the space subdivision process and this dual space is derived from the primal space and used to describe the node relation graph (NRG). Node relation graph describes the topological relationship between indoor spatial environments such as adjacency and connection (Lee, 2004 [9]). The basic thoughts which are employed in dual space are state and transition [7]. They describe the feature types of dual space equivalent to cellspace and cellspace boundary that is already defined in primal space in terms of the connectedness of topologies. In dual space nodes represent cellspaces and edges represent transitions. In the Dual Space graph, the dashed line represents the non-navigable links that are the adjacency of graph e.g. walls, obstacles, etc, and the bold line represents navigable links that is the connectivity of graph e.g. doors, rooms, corridors, etc. To navigate from one room to another and find the distance between them, just go through the bold lines because these lines show the connectedness of rooms, and then to determine the shortest distance Dijkstra algorithm is applied if the distances or costs are known.

I. ROBOT BEHAVIORS

Behavior-Based Robotics is a behavior what an external observer sees a robot doing. Robots are programmed to display desired behavior. For example behavior of a robot, one behavior task can focus on traversing a path from start to goal state, while the other one focus on avoiding obstacle. However there are some fundamental ways to describe robot behavior such as Deliberative, Reactive, and Hybrid, Behavior-Based Control [10].

Deliberative control method in this the robot is going to uses all its internally stored knowledge and available sensory information to reason regarding what actions to be taken next. Robot decision making is described as the process in which the robot is getting information regarding the environment with the robot's sensors, preparing it as essential in the form to build decisions about how to act, and performing activities in the environment. Reasoning in deliberative systems occurs usually in the form of planning, needing an exploration of possible state–action sequences and their consequences. Planning needs the occurrence of an internal and symbolic representation of the world that supports the robot to watch forward into the future and predict the results of possible actions in many states to generate plans. Meanwhile, the internal model needs to be maintained accurate and up to date, if there is enough time to generate a plan and the world model is accurate, this strategy enables the robot to behave strategically, choosing the most suitable way of action for a given situation.

Reactive control method for tightly coupling sensory inputs and effector outputs, usually including no intervening reasoning to allow the robot to respond very immediately to developing and unstructured environments. Reactive control is encouraged by the biological knowledge of stimulus-response, there is no need for acquisition or maintenance of world models. Reactive control does not react on the kinds of complex reasoning rules employed in deliberative control rather it uses a rule-based method that involves a minimum quantity of calculation. Reactive systems perform fast real-time responses by mapping the robot's controller in a group of the preprogrammed, simultaneous condition–action rules with the minimum inner state that produces a well suited reactive control for dynamic and unstructured worlds. Reactive systems are able to react suitably for quickly changing environments with the presence of minimum quantity of calculation [11].

Hybrid control method merges the advantages of both reactive and deliberative control: such as the real-time reply of reactivity and the rationality and optimality of deliberation. As a conclusion, the hybrid control method includes a pair of different components such as the reactive/simultaneous condition–action rules and the deliberative decision making, which need to communicate in order to construct an understandable output [12]. The method is challenging because the reactive component is based on the robot's quick requirements, such as moving towards the target while avoiding obstacles, and therefore operates on a very fast time scale and applies direct external sensory data and signals, while the deliberative component applies extremely abstracted, symbolic, internal descriptions of the world, and perform functions on them upon a longer time scale such as to perform global path planning or plan for high-level decision-making. The reactive system needs to override the deliberative system if any unexpected and immediate challenge existing in the world and the deliberative system should inform the reactive system to guide the robot to more efficient and optimal trajectories and goals.

Behavior-based control system operates on a set of distributed, interacting modules known as behaviors that collectively accomplish the desired system-level behavior. The behaviors are patterns of the robot's action developing from communications between the robot and its environment in an outside spectator. Accordingly to a programmer, behaviors are the control modules that cluster collections of constraints in order to obtain and maintain a goal. Each behavior collects inputs from sensors or/and from other behaviors in the system and gives outputs to other behaviors or to the robot's actuators. Therefor behavior-based controller system is a structured network of interacting behaviors with no centralized world representation or focus of control while individual behaviors and networks of behaviors keep any state information and models. Well-designed behavior-based systems holds the advantage of the dynamics of communication within the behaviors themselves, and between the behaviors and the environment [13]. Unlike reactive control system that applies the collections of reactive rules with few if any state and no representation while as the behavior-based control system applies sets of behaviors which have state that can be employed to build representations, thereby enabling reasoning, planning, and learning.

Wall Following behavior [14] example for robot behavior navigation. In this Robot can follow the wall, therefore allowing this to walk along the boundaries of the testing environment. The basic procedure for wall following procedure is shown in figure 1. In this, the robot uses ultrasonic sensors to detect the left and right side distances (dr and dl). The sensing system determines the side distances of the robot, later on it judges which side (left or right) is nearby to the robot and selects the distance to the nearest side as the baseline, allowing the robot to move along the nearest wall. After moving to the nearest wall the robot estimates the current location against that in the previous (far away from or closer to the sidewall). If the distance of the previous is lesser than of current, then it implies that the robot moved farther from the wall and then the robot needs to turn toward the wall. If the distance of the current is shorter than previous then it implies the robot has moved closer to the wall and the robot must turn away from the wall now. de is the difference between the wall distances in the two cycles. Fuzzy logic controller takes two inputs de and vc. vc is the relatively fast velocity which means that the robot has obtained significant speed by moving away from or closer to the wall. θe is the output of the fuzzy controller which allows the robot to turn toward the wall and return to correct the wall-following behavior.

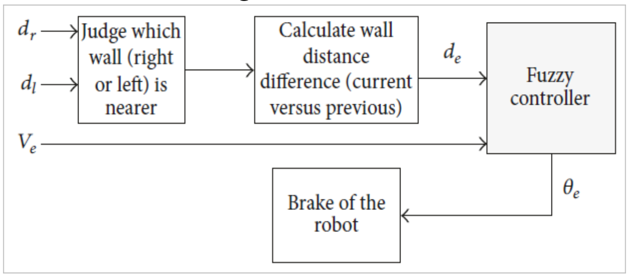

Figure 1. Wall following behavior procedure

II. Combinatorial Planning

Combinatorial planning is an approach of motion planning algorithm [15]. The motion planning algorithm can be expressed such as a robot is given with its initial state, final state, and geometric description of the robot and world. Then determine a path or sequence of valid configurations that move the robot gradually from an initial to the final state without hitting any obstacle. The motion planning algorithm approach is also referred to as piano mover's problem in which a piano is presented to move from one room to another without hitting an obstacle. Translation and rotation are the main ideas in robot motion planning that is required to move a robot. The C-space (configuration space) is the key idea in motion planning. It is the space of all possible (valid) configurations and can also be described as a topological manifold (is a topological space in which for any two distinct points there exists a neighborhood of each of them but which is disjoint from the neighborhood of the other also called euclidean hausdorff space). There are two basic approaches to represent C-space [16] such as combinatorial and sampling-based planning. Combinatorial planning is finding the path over the continuous C- space without resorting to approximations. Different methods are defined in combinatorial planning such as visibility graphs, voronoi diagrams, and exact cell decomposition.

Visibility graph [17] is defined as the graph of intervisible locations, usually for a set of points and obstacles in the Euclidean plane. In this graph, each node represents a point location (stop area), and each edge shows a visible connection, shown in figure 2(a). The basic idea is to draw a straight line from start (qI) and goal (qG) to all visible vertices, and then again draw a straight line from one vertex to all visible vertices. After connecting these vertices the shortest path is shown as in figure 2(b).

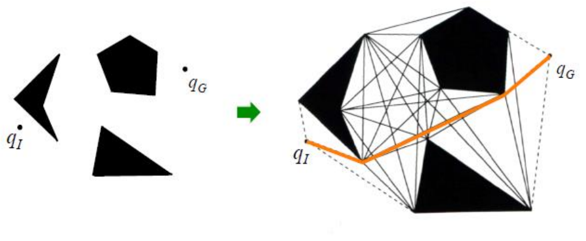

Figure (a). Figure 2(b)

Figure 2. Visibility Graph

Voronoi diagram [18, 19] division of a plane into regions based on distance to points in a specific subset of the plane. In this set of points that are closest to two or more obstacles boundaries are set to equal distance, the basic idea in this is to maximize the clearance between the points and obstacles as shown in figure 3. And then calculate the shortest path (by using algorithms) from start to goal. The path yielded will be equally from each obstacles. Three types of nodes exist in the voronoi diagram such as intersection nodes, terminal nodes, and pseudo-nodes. An intersection node is formed where three or more arcs of the voronoi diagram intersect. A terminal node compares to a dead end of a voronoi diagram arc and the terminal nodes appear during curved vertices are present. Pseudo-node also called a source or a goal node, these are the artificial nodes inserted in the graph near to the source and goal states individually and these nodes also represent entry and exit points.

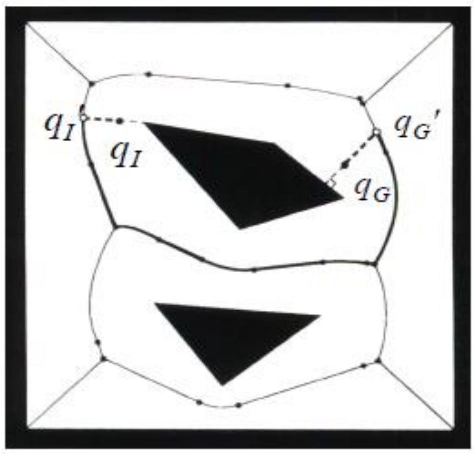

Figure 3. Voronoi diagram

Exact cell decomposition [16] is defined as the division of free space in C-space into cells, shown in figure 4. The basic method is decompose the free space with vertical lines from vertices without crossing a forbidden space (obstacle space). Mark the center to each of these vertical vertices and trapezoid a graph node and add centre point between the two vertical lines. Find shortest path in this graph with a graph searching algorithms such as Dijkstra's algorithm [20] etc.

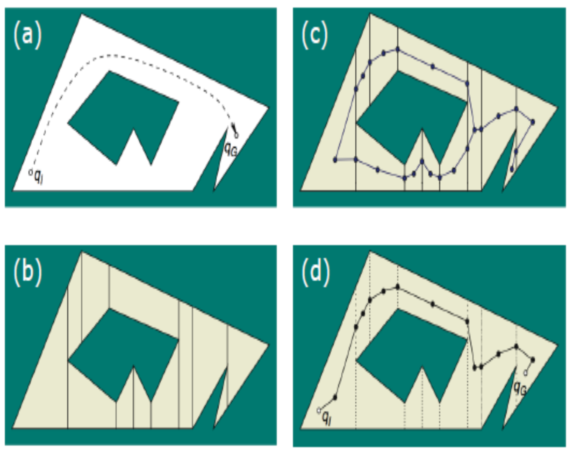

Figure 4. Exact cell decomposition

III. Sampling Based Planning

Sample Based Planning (SBP) is the fact that planning occurs by sampling the configuration space (C-space [21]). The responsiveness of the sampling-based planning approach is to try to catch the connectivity of the C-space by sampling it. Sampling-based planning gives a feasible solution because it is treated as a black box that means it returns a collision-free path once information about the start and goal configurations is provided, as shown in figure 5. Probabilistic Roadmap Method (PRM) and Rapidly-exploring Random Trees (RRT) are the two basic algorithms that are used in sampling-based planning. In SBP figure several terms are used such as Sampling, Metric, and Nearest Neighbor, Local planning, etc. Sampling means to select a configuration randomly and add it to the tree or roadmap. The selected samples can be present either in the free, or obstacle configuration space. In metric, given two configurations qa and qb, this procedure returns a value, or cost, that signees the effort required to reach qb from qa. Nearest-Neighbor (NN) is a searching algorithm that returns the closest node(s) to the new sample. Select Parent says to select an existing node to connect to the newly sampled node and the existing node is considered a parent. Given two configurations qa and qb then Local planning attempts to establish a connection between these configurations. Collision checking (CC) is mostly a Boolean function that returns true (success), or false (failure) when connecting two configurations. A connection is successful if it does not intersect Cobs.

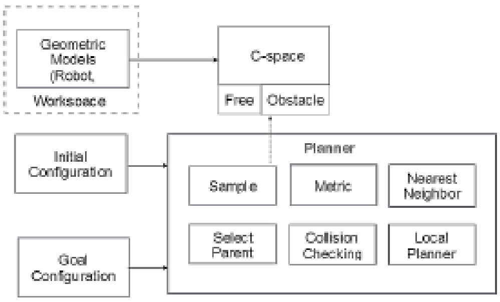

Figure 5. General procedure for sampling based planner

Probabilistic Roadmap Method (PRM) is applied to resolve the problem of deciding a path from a start to goal configuration of the robot while avoiding collisions. The basic idea of PRM is to take random samples from the configuration space of the robot [22], testing them for whether they are in the free space if so declare them as vertices, and use a local planner to try to connect them to other nearby vertices in configurations space. The two basic keywords that are used in PRM are Learning and query phase. In the Learning phase the first idea is to find the random sample of free configuration space and try to connect pair of nearby vertices of free C-space with the local planner if the possible valid path is found then add the edges to the graph. In the Query phase find the local connection of graph from start to goal positions, then use A* searching algorithms [23] to find the path. PRM is also called a multi-query planner. In PRM firstly it is required to construct the roadmap [24] in the learning phase and after roadmap construction we can describe the basic steps of this algorithms.

• By sampling method select a random node q-random from the configuration space.
• If q-random is found in Cobs (obstacles) then q-random is discarded otherwise add q-random to the roadmap.
• With the help of nearest neighbor algorithm find all the nearest nodes that are within the range of q-random using nearest-neighbor algorithm.
• Try to connect all these neighboring nodes to q-random using the local planner method.
• Checking for collision [25], if collision is found then disconnect the colliding paths.
• Repeat the process until the certain number of nodes are sampled.

Rapidly-exploring Random Trees (RRT) in this the basic intention is to select the random configuration from configuration space [26] if the selected random configuration is present in free space then try to connect the nearest vertex in the tree. The tree is growing incrementally from start to goal configuration. RRT also is referred to as a single-query planner and mostly used for single queries because it is faster than PRM that is RRT does not need to construct the roadmap i.e, learning phase. The following steps describe the working procedure of the RRT algorithm.
• Initialize with q-start.
• Select a random node q-random from the C-space using the sampling method.
• Discard the random node (q-random-discard) if it is found in obstacle space Cobs.
• Use Nearest Neighbor search q-near which is returned to see nearby configuration according to the metric.
• Connect the q-random and q-near using the local planner method. The local planner may recover q-new*q-random, that may not be accessible (reachable). If it is not reachable then it is discarded.
• Check for the collision to ensure that the path from q-near and q-new is collision free or not. If the path is collision free then q-new is added to the tree.
• Terminates the search (q-near) when q-new= q-goal.

CONCLUSION

IndoorGML is an XML-based standard of Open Geospatial Consortium introduced for Indoor Environments and it presents an important source of documentation of buildings, it also gives context to users operating location-based applications. IndoorGML describes the indoor environment and IndoorGML defines some basic concepts such as Cellular space model and Cell Geometry, Topology between cells and Cell semantics, multi-layered space model, and the core part Modular structure of IndoorGML. The core part of IndoorGML describes the basic components of the IndoorGML data model. It includes the schema definitions of basic classes for cell geometry, topology between cells, and multi-layered space model, and it also provides the semantic extension model for indoor navigation. The measurement of indoor distances is accomplished by horizontal and vertical distances and multi-modal transportation. The two main components that are described in IndorGML is Primal and Dual Space. The primal spaces are used to define the main types of the cellular indoor spatial model, such as room, corridor, and hall, and it also defines the edges of the geometric object, and its geometry maybe both surface or curve depending on the dimensionality. The dual spaces are derived from primal spaces, but in dual space, nodes represent cell-spaces, and edges represent transitions.

Choosing a suitable control behavior and designing a structure in it is best described by the situatedness properties of the problem, the nature of the task, the level of efficiency or optimal needed, and the capabilities of the robot, both in terms of computation, hardware, and world modeling. For example, reactive control is the most suitable choice for circumstances requiring an instant reply but such activity of reaction occurs at the price of being myopic not seeing into the past or the future. Deliberative systems are the only opportunity for domains that require a vast understanding of strategy and optimization, and in-service exploration and planning. Hybrid systems are suitably for environments and tasks where planning and internal designs are needed, and also the real-time requirements are few or adequately independent of the higher-level reasoning. Behavior-based systems, in contrast, are completely fit for environments including significant dynamic changes, where quick reply and adaptivity are significant, but the capability to do some watching progressing and avoid past mistakes is required.

Initial achievement in planning is to develop deterministic planning techniques. In motion planning, a robot is given with its start state, goal state, and geometric description of robot and world, and then find a path or sequence of valid configurations that move the robot gradually from start to goal while never touching any obstacle. Translation and rotation are the main focus in robot motion planning that is needed to move the robot. Combinatorial planning is used to find the path over the continuous configuration space without resorting to approximations and in this method, we define some concepts such as visibility graph, Voronoi diagram, and exact cell decomposition. After determining these methods it is required to try to connect the connectivity of start to goal state to find the valid optimal path using some graph searching algorithms such as Dijkstra. Simple based planning produces essentially in the configuration space (C-space). Configuration space is a space of all possible transformations such as free space and obstacle space. Combinatorial planning algorithms could be applied in a static environment where the obstacles are stationary while Sample-based planning methods are more complex and can be used dynamically changing environment. Because the layout of the building never changes, the usage of Combinatorial Planning algorithms is suggested. Visibility Graph yields a path which approaches the obstacles as close as possible. Hence the Visibility Graph method would give a good approximation of natural movement. The result of the Voronoi Diagram based method is quite the opposite of the previous one because it creates a route that is as far the obstacles as possible and this method seems to be quite useful for autonomous devices.

REFERENCES

[1] Markus Lupp. Open geospatial consortium. In Encyclopedia of GIS, pages 815–815. Springer, 2008.

[2] Portele Clemens. Geography markup language (gml) encoding standard, 2007.

[3] Abdullah Alattas, Sisi Zlatanova, Peter van Oosterom, and Ki-Joune Li. Improved and more complete conceptual model for the revision of indoorgml (short paper). In LIPIcs-Leibniz International Proceedings in Informatics, volume 114. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[4] Abdoulaye A Diakite´ Sisi Zlatanov, and Ki-Joune Li. About the subdivision of indoor spaces in indoorgml. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 4, 2017.

[5] M Kim and J Lee. Developing a method to generate indoorgml data from the omni-directional image. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 40:17, 2015.

[6] SS Mirvahabi and RA Abbaspour. Automatic extraction of indoorgml core model from openstreetmap. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 40(1):459, 2015

[7] Kang, H.K. and Li, K.J., 2017. A standard indoor spatial data model—OGC IndoorGML and implementation approaches. ISPRS International Journal of Geo-Information, 6(4), p.116.

[8] TC ISO. 211. ISO 19107 geographic information–spatial schema, 2003.

[9] Lee, J. 2004. A Spatial Access Oriented Implementation of a Topological Data Model for 3D Urban Entities, GeoInformatica 8(3), 235-262.

[10] Siciliano, Bruno, and Oussama Khatib, eds. Springer handbook of robotics. Springer, 2016.

[11] Vadakkepat, Prahlad, et al. "Fuzzy behavior-based control of mobile robots." IEEE Transactions on Fuzzy Systems 12.4 (2004): 559-565.

[12] J.H. Connell: SSS: A hybrid architecture applied to robot navigation, Proc. IEEE Int. Conf. Robotics Autom. (1992) pp. 2719–2724

[13] Towle, Bradford A., and Monica Nicolescu. "Real-world implementation of an Auction Behavior-Based Robotic Architecture (ABBRA)." 2012 IEEE International Conference on Technologies for Practical Robot Applications (TePRA). IEEE, 2012.

[14] Chen, Ching-Han, et al. "Fuzzy logic controller design for intelligent robots." Mathematical Problems in Engineering 2017 (2017).

[15] LaValle, Steven M. Planning algorithms. Cambridge university press, 2006.

[16] Burgard, Wolfram, et al. "Introduction to mobile robotics." University of Freiburg (2011).

[17] Asano, T., Asano, T., Guibas, L., Hershberger, J., & Imai, H. Visibility-polygon search and euclidean shortest paths. The 26th Annual Symposium is Foundations of Computer Science (SFCS 1985) (pp. 155-164). IEEE.

[18] Canny, John. "A Voronoi method for the piano-movers problem." Proceedings. 1985 IEEE International Conference on Robotics and Automation. Vol. 2, IEEE, 1985.

[19] Takahashi, O., & Schilling, RJ (1989). Motion planning in a plane using generalized Voronoi diagrams. IEEE Transactions is robotics and automation, 5 (2), 143-150.

[20] Dijkstra, Edsger W. "A note on two problems in connexion with graphs." Numerische mathematik 1.1 (1959): 269-271.

[21] Elbanhawi, Mohamed, and Milan Simic. "Sampling-based robot motion planning: A review." Ieee access 2 (2014): 56-77.

[22] Amato, Nancy M., and Yan Wu. "A randomized roadmap method for path and manipulation planning." Proceedings of IEEE international conference on robotics and automation. Vol. 1. IEEE, 1996.

[23] Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. "A formal basis for the minimum cost paths." IEEE transactions are Systems Science and Cybernetics 4.2 (1968): 100-107.

[24] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, ``Probabilistic roadmaps for path planning in high-dimensional conguration spaces,'' IEEE Trans. Robot. Autom., vol. 12, no. 4, pp. 566_580, Aug. 1996.

[25] Sánchez, Gildardo, and Jean-Claude Latombe. "On delaying collision checking in PRM planning: Application to multi-robot coordination." The International Journal of Robotics Research21.1 (2002): 5-26.

[26] M. LaValle, ``Rapidly-exploring random trees: A new tool for path planning,'' Dept. Comput. Sci., Iowa State Univ., Ames, IA, USA, Tech. Rep. TR 98-11, 1998.