Navigation, Path Planning, and Task Allocation Framework For Mobile Co-Robotic Service Applications in Indoor Building Environments

by

Bharadwaj Mantha

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in the University of Michigan
2018

Doctoral Committee:

Associate Professor Carol C. Menassa, Chair
Assistant Professor Clive D'Souza
Professor Vineet R. Kamat
Associate Professor SangHyun Lee

Bharadwaj Mantha

baddu@umich.edu

ORCID iD: 0000-0002-3173-3966

DEDICATION

I would like to dedicate this dissertation to my loving and caring parents (Bhavani Shankar Mantha and Suryakumari Peri), wife (Ramani Ayyagari), brother (Dheeraj Mantha), and in laws (Srinivas Rao Ayyagari and Padmavathi Ayyagari).

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Carol Menassa for all the support and guidance throughout my PhD. Her ever smiling face made me feel comfortable right from the moment I met her. I found her to be uniquely capable of capturing my attention and helping me learn confidently. She was always supportive, optimistic and encouraging in all the encounters. I have learned a great deal from her especially hardworking, professional etiquette, academic writing skills, presentation skills, and most importantly the technical expertise to name a few.

I would be deeply indebted to Dr. Vineet Kamat who I closely collaborated with since the beginning of my graduate studies. His passion for research, simplicity, patience, and technical abilities helped me polish my skills. I will never forget the conversations I had with him during a few successful and challenging phases of my PhD. It has been an honor working with him.

I also want to thank the rest of my committee members Dr. SangHyun Lee and Dr. Clive D'Souza. Especially, Dr. Lee's constructive feedback during my proposal defense to improve the technical and logical continuity of my thesis was invaluable. In addition, the wheelchair navigation study wouldn't have been possible without Dr. D'Souza's guidance and suggestions. I would also like to specially thank Dr. Branko Kerkez for having inspired me with his out of the box approach during my preliminary examination.

I also thank my lab mates (SICIS'ers and LIVE'ers) Albert Thomas, Kurt Lundeen, Yong Xiao, Da Li, Chen Feng, Lichao Xu, Xi Wang, for round the clock advice, sleepless working nights, and for all the fun we had in the past few years.

Last but definitely not the least I would like to thank all the members of my family and friends who directly or indirectly made this possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

ABSTRACT

Recent advances in computing and robotics offer significant potential for improved autonomy in the operation and utilization of today's buildings. Examples of such building environment functions that could be improved through automation include: a) building performance monitoring for real-time system control and long-term asset management; and b) assisted indoor navigation for improved accessibility and wayfinding. To enable such autonomy, algorithms related to task allocation, path planning, and navigation are required as fundamental technical capabilities. Existing algorithms in these domains have primarily been developed for outdoor environments. However, key technical challenges that prevent the adoption of such algorithms to indoor environments include: a) the inability of the widely adopted outdoor positioning method (Global Positioning System - GPS) to work indoors; and b) the incompleteness of graph networks formed based on indoor environments due to physical access constraints not encountered outdoors.

The objective of this dissertation is to develop general and scalable task allocation, path planning, and navigation algorithms for indoor mobile co-robots that are immune to the aforementioned challenges. The primary contributions of this research are: a) route planning and task allocation algorithms for centrally-located mobile co-robots charged with spatiotemporal tasks in arbitrary built environments; b) path planning algorithms that take preferential and pragmatic constraints (e.g., wheelchair ramps) into consideration to determine optimal accessible paths in building environments; and c) navigation and drift correction algorithms for autonomous mobile robotic data collection in buildings.

The developed methods and the resulting computational framework have been validated through several simulated experiments and physical deployments in real building environments. Specifically, a scenario analysis is conducted to compare the performance of existing outdoor methods with the developed approach for indoor multi-robotic task allocation and route planning. A simulated case study is performed along with a pilot experiment in an indoor built environment to test the efficiency of the path planning algorithm and the performance of the assisted navigation interface developed considering people with physical disabilities (i.e., wheelchair users) as building occupants and visitors. Furthermore, a case study is performed to demonstrate the informed retrofit decision-making process with the help of data collected by an intelligent multi-sensor fused robot that is subsequently used in an EnergyPlus simulation. The results demonstrate the feasibility of the proposed methods in a range of applications involving constraints on both the environment (e.g., path obstructions) and robot capabilities (e.g., maximum travel distance on a single charge). By focusing on the technical capabilities required for safe and efficient indoor robot operation, this dissertation contributes to the fundamental science that will make mobile co-robots ubiquitous in building environments in the near future.

Chapter 1

Introduction

Recent advancements in technology have given rise to the use of intelligent robots for several service applications. Some of the examples of deployed indoor robotic systems for professional and domestic service applications include museum guide robots (Burgard et al. 1998), hotel butler robots (Boltr 2014), vacuum cleaning robots (iRobot 2016), and surveillance robots (SDR 2016). As per a report published by Robotics Virtual Organization in 2013, the annual growth rate in professional and service robots is 30% and 20% respectively. In addition, Baeg et al. 2011 and a report published by Robotics VO, 2009 emphasized the significance, usability, and the potential of service robots for everyday activities. It can thus be reasoned that intelligent robots will soon be ubiquitous and there is a strong need to explore the potential of robots to improve autonomy in the operation and utilization of today's buildings. Specifically, this thesis is focused on two main applications a) building performance monitoring for real-time system control and long-term asset management; and b) assisted indoor navigation for improved accessibility and wayfinding.

To enable such autonomy, robots need to be able to a) optimally divide the task among themselves (task allocation); b) plan their respective paths for the assigned task locations (route/path planning); c) identify and orient their location in the physical environment (localization); and d) direct to the respective locations of interest (navigation). Prior work has primarily either made application-specific assumptions (e.g. shortest distance path) or attempted to adapt and modify existing algorithms (e.g. Traveling Salesman Problem) from other domains to solve such technical problems. This dissertation proposes a generic and unifying interdisciplinary approach to address the limitations of existing studies and solve the problem of optimized task allocation, route planning, autonomous navigation, and preferential path planning for indoor single and multiple co-robotic systems. The proposed methodology is general, not domain specific, and can be used for several single and swarm indoor robotic system applications. The applicability and feasibility of the proposed approaches are illustrated with the help of simulated case studies and physical deployments in real building environments.

1.1 Background and Literature Review

Based on the previous discussion, the literature review is broadly classified into three categories. The first category deals with task allocation and route planning algorithms, the second category relates to localization and navigation for autonomous indoor robots and the third path planning algorithms for assisted indoor wayfinding. The subsequent discussion in this section provides an overview of the previous methods and their limitations. A comprehensive analysis is further provided in the following chapters of this dissertation.

1.1.1 Task Allocation and Route Planning

Optimized tour planning (in case of single and multiple robots) and task allocation (in case of multiple robots) to achieve dynamic goals are not only critical but also challenging because of the complex geometry in typical indoor multi-level structures. A tour is defined as a route to cover all the desired locations originating and ending at the same location whereas a path is defined as a route to cover all the desired locations (Bellman 1962). For example, with an increase in the complexity of building geometry, the computational requirements increase for finding the most optimal tour through a certain set of locations (LaValle and Kuffner 2001). This can be broadly classified as a combinatorial optimization problem and a special case of Vehicle Routing Problem (VRP) known as Travelling Salesman Problem (TSP) (if single agent or robot is involved) and Multiple Travelling Salesman Problem (mTSP) (if multiple agents or robots are involved) (Rao and Biswas, 2008; Ponraj and Amalanathan, 2014).

In addition, it is tedious, costly, and computationally complex to find the most optimal solution in polynomial time. For example, in a network that has n nodes (or task locations to be visited), the algorithm takes k (polynomial function of n) steps to find the most optimal solution making this computationally intensive for large networks typically encountered in indoor built environments (Gorbenko et al. 2011; Brumitt and Stentz 1996).

Thus, alternate methods/techniques have been developed to determine a near optimal solution which is closest to the optimal solution. For example, Ryan et al. 1998 solved the routing problem for Unmanned Aerial Vehicles (UAVs) in outdoor environments with the help of a heuristic based approach (Reactive Tabu Search (RTS)) within a discrete-event simulation. However, the same cannot be applied to indoor built environments since the graph network created is incomplete due to physical access constraints (e.g., nodes can only be accessed through corridors and stairs).

Yu et al. 2002 utilized a Genetic Algorithm (GA) based approach to solve the route planning problem for cooperative autonomous mobile robots where each robot is responsible for a unique Hamiltonian path, a traceable path where the entire network of nodes or the specific collection of nodes can be traced by visiting each and every node exactly once. However, this might be difficult in indoor environments where obstructions might require the robot to take an alternate path to achieve tasks or return to a recharging station. Gorbenko et al. 2011 show that the planning of a typical work day for indoor service robots is Nondeterministic Polynomial (NP) complete and described an approach to solve the problem by converting it into a problem of finding a Hamiltonian path. However, unlike the above methods, due to the incomplete nature of the graph (in case of indoor environments) network, the most optimal tour solution for each of the robots might not be a Hamiltonian path or Hamiltonian circuit or might not contain one.

In addition to the limitations discussed, the aforementioned studies are context specific, subjected to certain assumptions and lack a general framework to solve the problem where a group of robots needs to optimally divide the set of tasks and plan their tour from a single depot (or charging station), to cover a set of destination locations (at least once) and return to the same base station. Example applications include a robot (or group of robots) that needs to deliver equipment or medicines from a storeroom or pharmacy to various rooms in a hospital and return back to the storeroom or pharmacy. Another example can be periodic data collection of ambient parameters in a building for making informed operation or retrofit decisions and determine real-time energy saving measures (Mantha et al., 2016).

1.1.2 Localization and Navigation

In the current context, localization is defined as the robot’s ability to identify its current location in a given indoor environment setting (Levitt and Lawton, 1990). For example, a robot being able to recognize its current location to be in room 201, or knowing its location and orientation in the global coordinate reference system. The robot’s navigation can be briefly defined as the robot’s ability to plan a course of action to reach the destination location while accurately localizing itself in its frame of reference at strategic locations (Levitt and Lawton, 1990).

Several indoor localization and navigation techniques have been explored previously. Literature suggests that every method has advantages and limitations. Some of the previous approaches explored for robot localization include Wireless Local Area Network (WLAN), Radio Frequency Identification (RFID), Ultra-Wide Band (UWB), Inertial Measurement Unit (IMU), Bluetooth, Cameras, and Lasers. Wifi is an economical solution because most of the existing infrastructure consists of wireless nodes required for localization. However, it suffers from the significant error in localization accuracy (Montañés et al. 2013; Torres-Solis et al. 2010). Bluetooth and RFID based localization tends to be expensive, time-consuming and also have space constraints because of the requirement of wireless infrastructure deployment indoors (Raghavan et al. 2010). Similarly, UWB based systems require a large number of receivers making it inconvenient and infeasible (due to space constraints) indoors (Montañés et al. 2013).

Laser scanner and natural marker (camera) based techniques eliminate the need to instrument the physical space but they are highly expensive, sensitive to obstructions and lighting conditions and require high computational capabilities (Feng and Kamat 2012; Habib 2007; Thrun et al. 2005; Bar-Shalom et al. 2004; Burgard et al. 1998). To summarize, common disadvantages affecting a majority of the reviewed methods include low accuracy, significant upfront investments, high computational requirements and complex instrumentation of the indoor environment.

One of the vision-based methods using fiducial markers, however, is particularly immune to the aforementioned disadvantages afflicting other methods. Fiducial markers (Olson, 2011) offer high accuracy in determining and estimating their relative 3D pose in an environment, require relatively less computing capabilities, are cost-effective and are easy to install (Iwasaki and Fujinami, 2012). In addition, fiducial markers have the capability to store virtual information regarding a multitude of things such as information regarding physical location (floor and room level information), emergency evacuation directions, indoor navigational information, and inspection related data regarding building systems helpful for facility managers (Feng and Kamat 2012). Feng and Kamat (2012) have demonstrated how markers having virtual information and navigational directions can help humans navigate indoors. To take this further, this research utilizes the virtual location information (for localization), navigational direction (for navigation), and 3D pose estimates (for drift correction) to achieve autonomous behavior of the mobile robot.

1.1.3 Path Planning

In the context of indoor navigation, planning refers to the ability to make higher-level decisions based on available information and cognition is the ability to continually process environmental information and take corresponding actions over time without external help. Several technological interventions (i.e. indoor navigation technologies) were suggested to reduce the planning and cognitive burden on the user.

To overcome the disadvantages faced by the other approaches (as discussed previously in localization and navigation section), studies explored vision-based navigation systems that use cameras and markers and provide turn by turn instructions (Oliveira et al., 2016; Kim et al., 2015; Gionata et al., 2014; Carlson and Demiris 2012). Markers are landmarks which help identify and localize the users in the indoor environment. Neges et al., (2015) created an indoor assisted navigation system using natural landmarks and a camera. However, some of the limitations of this approach are a) the start location has to be manually determined by the user; and b) the proposed method works only for navigation within a single floor and does not take into account the staircases and elevators or need to navigate the wheelchair across different floors. These systems, in general, require prior training for determining the visual cues in the environment and thus are context specific and sensitive. However, fiducial markers such as April tags developed by Olson (2011) are highly accurate, require relatively less computing capabilities, are cost-effective (they can be printed on paper) and are easy to install (Iwasaki and Fujinami, 2012). Though physical instrumentation of the space is required, it is cost effective and has less to no maintenance compared to other approaches where beacon/receiver installations are required. Despite these advantages, the current fiducial marker based systems still lack several features that are crucial to assisting the general population indoors. For example, most of these approaches do not provide crucial indoor environment information such as access ramps, escalators, elevators, exits, and other significant locations (e.g. water fountains and restrooms).

Furthermore, none of the proposed solutions describe the general principles of marker network and graph network creation based on the indoor environment. This is crucial for successful assisted indoor navigation as the path computations are determined based on the networks formed. For instance, these approaches did not consider a situation where a marker is missed by the user. That is, if the user takes a wrong turn along the path, he/she will only be redirected at the next landmark (or marker in this case). This not only causes additional burden on the user but also has implications on the total journey time. In addition, markers are only placed at strategic locations making it difficult for the users to be cognizant of the immediate future event. That is, the users do not have any knowledge of their next action until a marker is scanned.

Proper navigation systems designed for all sects of the population must also take into account human preferences (Oliveira et al., 2016). However, most of the approaches considered only distance and determined the shortest path (Wu et. al., 2011; Teo and Cho 2016; Gecko 2017) with the help of standard algorithms such as Dijkstra's or A* (Dijkstra 1959; Dorigo et al.,

2008) neglecting other factors such as accessibility and human preferences which are especially crucial for individuals with disabilities (Morales et al., 2015). For example, a preferred path can include a path with automatic caution doors along the way or a path which consists of fewer turns and not necessarily the shortest distance from the current location to the destination. In such cases, the path cannot be directly computed with the help of existing algorithms.

To summarize, the following are some of the key limitations and research gaps identified in the existing studies on indoor robotic navigation, path planning, and task allocation:

- The challenges and/or factors that prevent the application of existing algorithms (mainly adopted for outdoor networks) to indoor service networks are as follows:
    a. The path networks formed are incomplete and hence the corresponding cost matrix is incomplete
    b. To visit a group of nodes (or locations), a unique Hamiltonian tour might not exist
    c. Network topology plays a key role (i.e. the placement and the connectivity of the node matters unlike for outdoor networks)
    d. The shortest distance between any two nodes (or locations) in the network is not the straight line joining them
- Lack of fundamental basis of the path planning process which is graph network and the corresponding marker network map creation. That is, the process of graph/marker network creation based on the indoor environment which involves defining a node, defining an edge, defining an edge attribute, defining a marker location, and its corresponding marker placement
- Low accuracy, significant upfront investments, high computational requirements and complex instrumentation of the physical space for indoor localization and navigation

1.2 Research Objectives

The overall objective of this research is to develop novel navigation, planning, and task allocation algorithms for indoor mobile robots that are not context specific and oversome the aforementioned limitations and real-world constraints. In addition, the proposed framework were validated and tested with the help of case studies and real-world experiments. Specifically, the research objectives are identified as follows:

1 Develop a generalized framework (which can be applied to any indoor robotic service network irrespective of the application) for optimized route planning and task allocation in complex indoor environments for single and multi-robotic systems

- Conduct scenario analysis (along with relevant practical built environment applications) and compare the performance of proposed approach with existing algorithms, and analyze them with and without pragmatic constraints

2 Develop navigation and drift correction algorithms for autonomous mobile robots in buildings

- Investigate the practical feasibility of the proposed algorithms on a real physical system

3 Develop a human preferential path planning algorithm based on the user preferences and indoor graphs

- Perform a case study with the help of four scenarios to show the applicability of the proposed approach
- As a proof of concept, develop a navigation interface that can assist disabled users in the built environment and pilot test it with the help of individuals without disabilities

1.3 Research Methodology

The methodology of this research takes prior knowledge from the application domain (e.g. facility management) into consideration and adapts existing general outdoor route planning algorithms or navigation algorithms to better meet the requirements of built environment applications. In particular, algorithms developed for outdoor logistic applications make assumptions that do not meet the needs of indoor building environments, an issue that this research specifically addresses.

In addition, the frameworks and methods developed in this research are not limited to the specific applications demonstrated in the case studies but are also applicable to other built environment, construction, and infrastructure-related applications. For example, the proposed iterative algorithm for optimized task allocation and route planning can accelerate the search of an entire floor plan (or building) for human survivors and monitor the noise levels of a social event in a dormitory building. The marker-based localization and navigation can be applied to structural health monitoring of bridges. The attribute loaded graph networks can help individuals with disabilities with preferential path planning in outdoor navigation. The key methods and applications of this research are shown in Figure 1.1. Therefore, robotics and operations research domains also stand to benefit from the algorithms and frameworks developed.

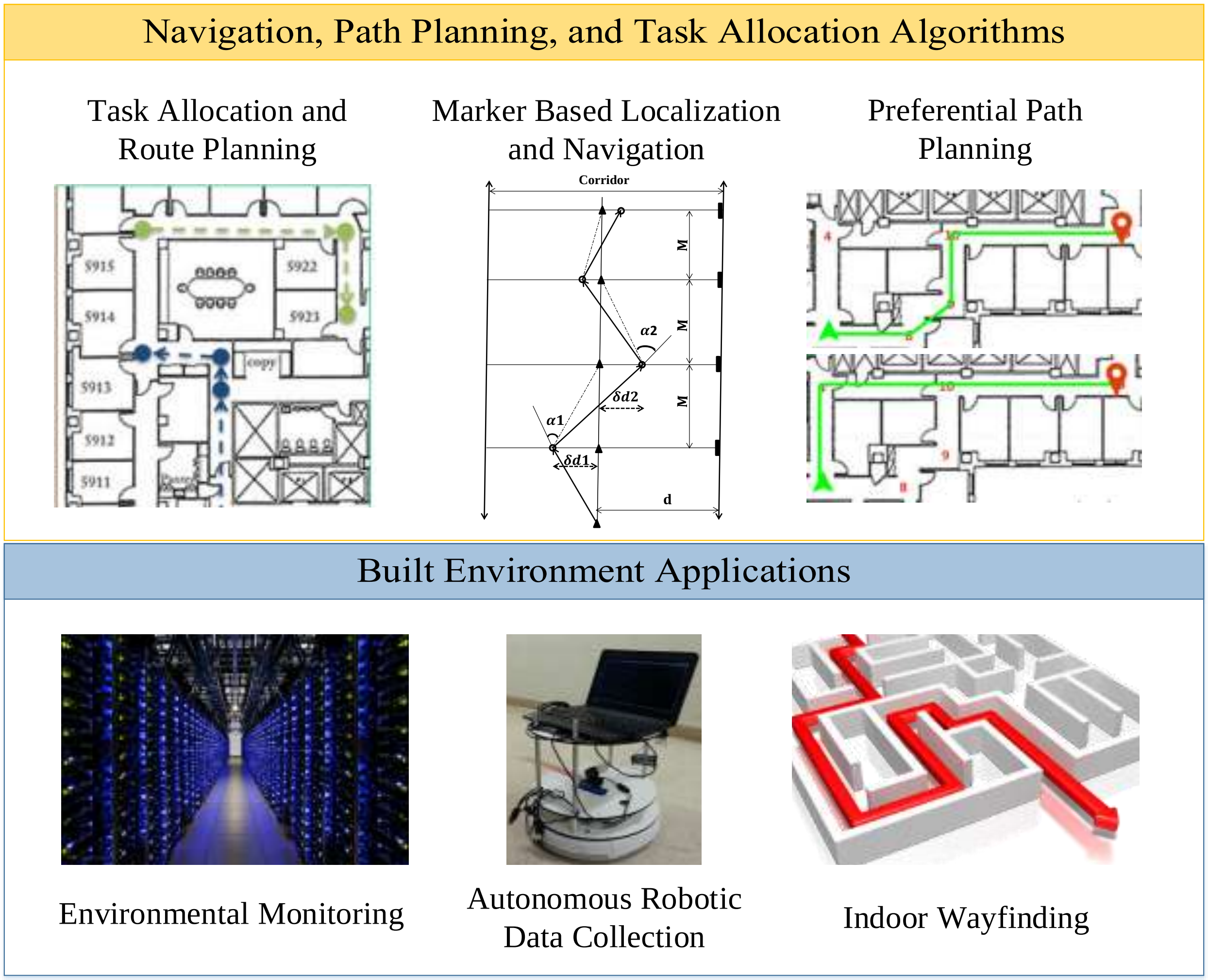


Figure 1.1 Research Overview (Topics and Applications)

1.4 Dissertation Outline

This dissertation is a compilation of different peer-reviewed scientific manuscripts. Since each of these chapters (chapters 2 to 4) represents a standalone paper, it has to be noted that there might be some overlap in the introduction, background, and literature review sections. The current chapter (Chapter 1) outlined the importance of this research. Chapter 2 presents the route planning and task allocation algorithms for indoor multi-robotic systems, i.e. optimally divide the set of tasks among the robots (task allocation) and plan their respective tours to several locations of interest (planning). Chapter 3 discuss the need for an efficient data collection system in buildings, describe navigation algorithms developed for a mobile agent (robot) based data collection, and finally develops a framework to utilize the data collected by robot in making retrofit decisions. Chapter 4 describes a preferential path planning algorithm for assisted indoor navigation in complex indoor environments. Chapter 5 summarizes the major contributions of this research and discusses further work directions.

Chapter 2

Planning and Coordination Algorithms for Indoor Robotic Service Networks

2.1 Introduction

A tour is defined as a route to cover all the cities (or locations) originating and ending at the same city (or location) whereas a path is defined as a route to cover all the cities (Bellman 1962). Most of the prior works have either solved the path planning problem (for indoor and outdoor networks) or tour planning for outdoor networks but did not consider indoor networks (Jose and Patinder 2016; Ponraj and Amalanathan 2014). This paper presents a generalized framework for achieving optimized task allocation (i.e., divide the tasks assigned amongst the group of robots in the most desirable way) and route planning (i.e., determine the most optimal way to visit several locations) for multiple concurrent or recurring (repeating or periodic) tasks in dynamic indoor built environments using a robotic system. The robotic system can consist of a single or swarm of robots which have similar mobility characteristics, physical constraints, and all starting from a single depot (i.e., charging/base station) within the building.

Robots have become increasingly pervasive in our day to day lives, with global experts predicting that intelligent robots will soon be ubiquitous (Pinto 2016; Tidlen 2016). Early examples of indoor robotic systems include museum guide robots (Burgard et al. 1998), hotel butler robots (Boltr 2014), vacuum cleaning robots (iRobot 2016), and surveillance robots (SDR 2016). Most of the aforementioned applications typically utilize a single robotic system that does not require inter-robot coordination (i.e., communication among the robots in a group) to complete specific tasks. Depending on the need and type of application, however, it can be more reliable and robust to use a swarm robotic system consisting of multiple robots (Chen and Luh 1994). For example, it is easy to perform tasks such as structural health monitoring, building equipment inspection, search, and rescue operations with the help of a swarm robotic system instead of a single robot (Chen and Luh 1994). This is because a swarm robotic system allows for parallel task accomplishment where distributed multiple targets can be visited with significantly reduced effort and time, provide increased capacity to accomplish simultaneous time constrained tasks, and are robust against failures in the system (Dias and Stentz, 2001). Such capabilities make swarm robotic systems ideal to assist with repetitive tasks such as equipment delivery in hospitals, occupant comfort monitoring in commercial buildings, ambient parameter monitoring in data centers, building energy monitoring, tunnel inspection, carrying food and supplies, and real-time occupancy feedback.

In this paper, a generalized goal definition for tasks to be accomplished by a swarm robotic system would be to visit a set of targeted locations in a time constrained way before returning to the recharging (base) station. For all such task oriented robots, the robots need to be able to optimally divide the tasks assigned (task allocation) and have the cognitive ability to plan their respective tours (tour planning) to several locations of interest (or task requirement locations).

Optimized tour planning (in case of single and swarm of robots) and task allocation (in case of swarm of robots) to achieve dynamic goals are not only critical but also challenging because of the complex geometry in typical indoor multi-level structures. For example, with increase in complexity of the building, the computational requirements increase for finding the most optimal tour (LaValle and Kuffner 2001). The complexity here refers to the type of the floorplan where the increase in number of corridors results in larger network size and thus the increase in computational burden. This can be broadly classified as a combinatorial optimization problem and a special case of Vehicle Routing Problem (VRP) known as Travelling Salesman Problem (TSP) (if single agent or robot is involved) and Multiple Travelling Salesman Problem (mTSP) (if multiple agents or robots are involved) (Rao and Biswas, 2008; Ponraj and Amalanathan, 2014).

In addition, it is tedious, costly, and computationally complex to find the most optimal solution in polynomial time. For example, in a network that has n nodes (or task locations to be visited), the algorithm takes k (polynomial function of n) steps to find the most optimal solution making this computationally intensive for large networks typically encountered in indoor built environments (Gorbenko et al. 2011; Brumitt and Stentz 1996).

Thus, alternate methods/techniques have been developed to determine a near optimal solution which is closest to the optimal solution. For example, Ryan et al. 1998 solved the routing problem for Unmanned Aerial Vehicles (UAVs) in outdoor environments with the help of a heuristic based approach (Reactive Tabu Search (RTS)) within a discrete-event simulation. However, the same cannot be applied to indoor built environments since the graph network created is incomplete due to physical access constraints (e.g., nodes can only be accessed through corridors and stairs).

Yu et al. 2002 utilized Genetic Algorithm (GA) based approach to solve the route planning problem for cooperative autonomous mobile robots where each robot is responsible for a unique Hamiltonian path, a traceable path where the entire network of nodes or the specific collection of nodes can be traced by visiting each and every node exactly once. However, this might be difficult in indoor environments where obstructions might require the robot to take an alternate path to achieve tasks or return to recharging station. Gorbenko et al. 2011 show that the planning of a typical work day for indoor service robots is Nondeterministic Polynomial (NP) complete and described an approach to solve the problem by converting it into a problem of finding a Hamiltonian path. However, unlike the above methods, due to the incomplete nature of the graph (in case of indoor environments) network, the most optimal tour solution for each of the robots might not be a Hamiltonian path or Hamiltonian circuit or might not contain one.

In addition to the limitations discussed, the aforementioned studies are context specific, subjected to certain assumptions and lack a general framework to solve the problem where a group of robots need to optimally divide the set of tasks and plan their tour from a single depot (or charging station), to cover a set of destination locations (at least once) and return to the same base station. Example applications include a robot (or group of robots) that needs to deliver equipment or medicines from a store room or pharmacy to various rooms in a hospital and return back to the store room or pharmacy. Another example can be periodic data collection of ambient parameters in a building for making informed operation or retrofit decisions and determining real time energy saving measures (Mantha et al., 2016). Other examples include indoor mail delivery, indoor worker safety monitoring, military reconnaissance, facility management applications such as security, surveillance, home automation, tracking people in a museum, space monitoring (e.g., chemical spills, gas, and oil leaks), structural health monitoring, wayfinding, search and rescue.

In this paper, the authors propose a generic and unifying interdisciplinary approach to address the shortcomings of existing studies and solve the problem of optimized task allocation and route planning for known indoor single and swarm robotic systems. The proposed methodology is general, not domain specific, and can be used for several indoor single and swarm indoor robotic system applications. The applicability and feasibility of the multi-robot task allocation and route planning system is illustrated with the help of a case study where four scenarios are considered.

2.2 Related Work

The Travelling Salesman Problem (TSP) was first introduced in the 1930's and has been extensively studied and developed in the field of combinatorial optimization (Bellman 1962; Bellmore and Nemhuser 1968; Karp 1977; and Papadamitriou 1977). A more generalized and practical extension of TSP is multiple TSP (mTSP). A simple yet formal definition of the problem is as follows: Given n locations (cities) and m agents (salesmen), the problem is to find out optimized routes for each of the agents while optimizing the total sum of the tour length traveled by each of the agents. Although TSP was extensively studied and applied in several domains, the research on mTSP is limited (Kara and Bektas 2006; Sofge et al. 2002).

The mTSP can be comprehended as a twostep optimization problem. The first step involves determining the subdivision of all the n cities into m groups (or clusters representing the m salesmen), while the second step is concerned with identifying the shortest tour covering all the cities in each of the clusters. The second step is traditionally known as TSP and is a simplified version of the mTSP where the number of agents (or m) is one (Christophides 1976; Golden et al. 1981; Lawler et al. 1985; Laporte 1992; Hoffman et al. 2013). These steps are interrelated since the final solution is dependent on the way groups are divided and the way the route is optimized in each of the clusters. The hierarchy of the steps depends on the solution methodology employed. For example, some approaches divide the clusters first and then optimizes the route next and vice versa.

Sofge et al. 2002 proposed a neighborhood attractor schema for forming optimized clusters based on Shrink-Wrap algorithm and a variety of evolutionary computation algorithms and paradigms. Nallusamy et al. 2009 used K-means clustering, Shrink-Wrap Algorithm, and Meta-Heuristics to form optimized clusters for each of the salesman. However, both these approaches will not work with networks based on indoor built environments because they do not account for the network topology. That is, the basic assumption in these solutions is that the network is complete and thus any changes in the network topology yield the same optimal result. A network is said to be complete if each of the nodes (e.g., location in a building) in the network is accessible to every other node in the network which is not necessarily the case in a building environment. For example, two locations in different hallways in a building cannot always be accessible by the line segment joining them but only through the hallways.

Other heuristic based approaches explored by researchers for solving mTSP include Neural Networks (NNs) (Modares et al. 1999; Goldstein 1990; Wacholder et al. 1989), Tabu Search (Ryan et al. 1998), Simulated Annealing (SA) (Song et al. 2003), and Genetic Algorithm (GA) (Yu et al. 2002; Zhang et al. 1999). However, the aforementioned techniques cannot generally be used to solve the indoor service network case since the cost matrix is incomplete. A cost matrix is a square matrix where the size is the number of nodes in the network and the entries are the respective edge weights (such as distance between the nodes and time it takes to travel between the nodes). Furthermore, the initialization for most of the aforementioned algorithms is either a random tour or solution based on greedy search (i.e., determining global optimum by choosing the local optimum at every stage). This might result in having to run some of the existing algorithms for more number of iterations to reach the optimal solution, or in some cases never reach optimality at convergence.

In summary, the challenges and/or factors that prevent the application of existing algorithms (mainly adopted for outdoor networks) to indoor service networks are as follows: 1) The networks formed are incomplete and hence the corresponding cost matrix is incomplete; 2) To visit a group of nodes (or locations), a unique Hamiltonian tour might not exist; 3) Network topology plays a key role (i.e. the placement and the connectivity of the node matters unlike for outdoor networks); 4) The shortest distance between any two nodes (or locations) in the network is not the straight line joining them; and 5) The most optimal solution in case of multiple robots need not visit mutually exclusive nodes. That is, even though each of the robots is responsible for a distinct set of nodes, due to the network constraints involved, a specific node might be visited by more than one robot. Table 2.1 shows the list of significant characteristics of the problem statement considered in this paper and the Table 2.2 shows which of these characteristics are not being addressed by the existing studies.

Table 2.1 Characteristics of the problem statement

SNO	List of significant characteristics of the problem statement
1	Incomplete networks (or Cost matrix)
2	No unique hamiltonian tour
3	Network topology
4	Shortest distance between two nodes is not the straight line joining them
5	In case of multiple robots, each of the robots (or agents) need not visit mutually exclusive nodes
6	Initialization (informed instead of random)

Table 2.2 Limitations of the existing studies

Prior Study	Unaddressed Problem Characteristics
Sathyan et al. 2015	1, 2, 4, 5
Jose and Patinder 2016	5, 6
Yu et al. 2002	1, 2, 6
Ding and Castanon 2016; Gorbenko et al. 2011	2, 5
Nallusamy et al. 2009; Sofge et al. 2002	3

Song et al. 2003; Modares et al. 1999; Ryan et al. 1998; Goldstein 1990; Wacholder et al. 1989	1
Zhang et al. 1999	1, 6
Singh and Lodhi 2014	1, 2, 5, 6

This research addresses these limitations and presents a generalized, scalable, and adaptable framework for planning the tour and allocating the tasks to single and swarm robotic systems given any arbitrary indoor service network. Given ‘n’ tasks to be accomplished with the help of ‘m’ robots in an indoor environment, the proposed methodology is capable of determining near optimal solution for dividing the ‘n’ tasks among m robots and plans an optimal tour (such as least distance, least time, and most accessible) for each of the robots respectively.

The proposed framework is also adaptable to constraints such as minimum or maximum distance a robot can travel and rerouting in case of inaccessible paths due to temporary blockages or emergency situations in real time. Given the basic parameters such as the average velocity of the robot and frequency of task requirements, the algorithm can determine the optimal number of robots required for accomplishing a specific set of tasks. Conversely, given the basic parameters such as number of available robots and average velocity of the robots, the algorithm can determine the most optimal frequency at which the tasks can be accomplished.

2.3 Problem Statement And Characteristics

The problem statement can be mathematically expressed as follows. Given ‘m’ robots, $R_1, R_2, R_3, \dots., R_m$ all located at a single charging station location referred to as the base node. These robots are responsible to complete ‘n’ tasks located at different locations (nodes) $V_1, V_2, V_3, \dots., V_n$ where distance between any two nodes $V_i, V_j \; in \; V = \{V_1, V_2, V_3, \dots \dots, V_n\}$ is

$d\{V_i, V_j\} \in R^+$. $d\{V_a, V_b\} + d\{V_b, V_c\} + d\{V_c, V_d\} + d\{V_d, V_e\}$ denote the distance for a tour visiting the nodes $\{a, b, c, d, e\}$ starting from a. $E_{V_i V_j}$denotes an edge between nodes $V_i and\ V_j$ which is formed with the help of connecting links such as corridors, stairs, and elevators. The objective is to visit all the nodes in the network starting from and returning back to the base node. That is, determine 'm' tours (for each of the robots) such that the total distance traveled by the robots is minimized and each of the nodes is visited at least once by any of the robots. This means that the product of number of times an edge and the corresponding distance between $V_i$and V_j is visited needs to be minimized as shown in Eq. (1). This in linear programming terms is termed as an objective function or cost function.

$$\sum_{j=0}^{n} \sum_{i=0}^{n} k_{ij} d\{V_i, V_j\} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..Eq. (1)$$

where $k_{ij} \geq 0$ is the the number of times the V_i, V_j edge is visited.

Typically, in indoor built environments, the network is undirected or symmetric as shown in Eq. (2). An undirected network is a network that consists of all the edges which are bi-directional. For instance, shortest distance between node i and node j is equal to the shortest distance between node j and node i.

$$d\{V_i, V_j\} = d\{V_j, V_i\}\ \forall\ i, j \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..Eq. (2)$$

Eq. (3) denotes that the triangle inequality holds for the edge distances in the network. Thus, the sum of distances of any two adjacent edges is greater than or equal to the third edge formed by the end nodes as starting node of the first edge and ending node of the second edge.

$$d\{V_i, V_j\} + d\{V_j, V_k\} \geq d\{V_i, V_k\} \; \forall \; V_i, V_j, V_k \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots Eq. (3)$$

Some of the important characteristics and research questions pertaining to this problem are as follows: Given the battery life of the robot, an average angular, and linear speed of the robot; what is the objective/cost function (i.e., distance traveled); what are the optimal number of robots required to perform the required tasks based on required frequency; or conversely, what is the minimum frequency at which the robots can visit each node based on the number of robots available. All these questions need to be addressed given constraints such as distance (minimum and maximum distance robot needs to travel), accessibility (e.g., maintenance related activities can block certain corridors in the building), feasibility based on type of mobility (e.g. inability of some ground robots to use stairs), and other social issues such as privacy and human interaction with the physical robots.

2.4 Methodology

We propose a combinatorial approach to solve the problem of multi robot task allocation and route planning problem in indoor built environments. The proposed methodology can be broadly classified into two main steps, network formulation and network solution as described below. In addition, a detailed descriptive step by step procedure of the entire proposed methodology including the network formulation and network solution is shown in Figure 2.1 and discussed in detail in the following sections of this paper. As can be seen from the flow chart in the figure, the objective of the network formulation step is to create an indoor graph network based on the given task requirements. Similarly, the objective of the network solution is to solve the created graph network for single and multiple robots.

2.4.1 Network formulation

The first important step of the route planning problem is network generation which involves creating and setting up the network. Network formulation includes identifying the nodes in the network – where each of the nodes represent a location in the building that robots need to visit; forming the edges with pairs of nodes as end nodes for each of the edge – where each of the edges represent links such as corridors, stairs, and elevators; defining each of the nodes – denoting a nomenclature for each of the nodes and associating each of the nodes with corresponding locations in the building; defining each of the edges - denoting a nomenclature for each of the edges and associating each of the edges with corresponding links in the building; and assigning weights to each of the edges – where weights can represent a variety of things such as Euclidian distance, time, and ease of accessibility between the nodes. The whole process of network formulation is described in detail in Multi-layer information generation section. The network generated will mostly be an incomplete network given that it is based on an indoor environment and physical connections are not possible between all the nodes (locations) in a building. The significance and process of converting an incomplete network to a complete network is discussed in detail in graphical representation section along with the concept of pseudo edges which do not represent actual physical connection between the nodes.

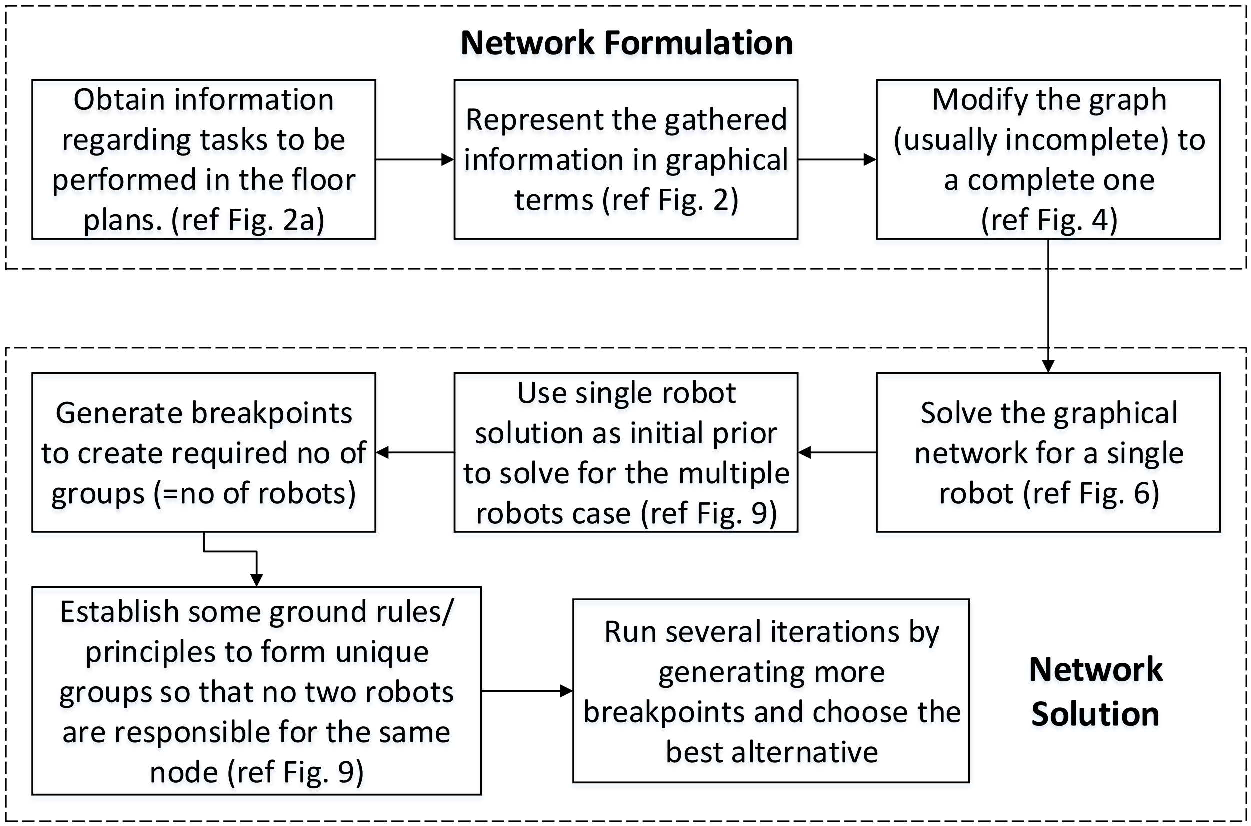


Figure 2.1 Detailed descriptive procedure of the proposed methodology

2.4.1.1 Multi-layer information generation

The generation, transformation, and storage of the given information are done in three different layers namely a) Task requirement layer b) Landmark network layer and c) Reduced node network layer. Given a set of task requirements in a known indoor environment, it can be replaced with the respective locations in the floor plan of the building as shown in Figure 2.2(a). For the purpose of localization and navigation of the robots, a network of landmarks needs to be created in the environment.

Without loss of generality, assume fiducial markers (developed by Olson 2011) which have high accuracy in determining the relative 3D pose in an environment and also require relatively less computing capabilities are used as landmarks in the environment. In addition, they are highly cost-effective, reconfigurable, and very easy to install. These markers link between actual physical locations and virtual information stored regarding those locations) for indoor robot localization. For this study, unique fiducial markers (landmarks) are required to be placed in strategic locations to define all of the notable locations in a building (such as rooms, lounges, start and end of hallways, stair cases, and elevators) along the navigational path of the robot as shown in Figure 2.2b. These markers, which are printed on regular paper, are used to store the physical location information (e.g., Room 101) which is necessary to help the robot determine its current location. The information regarding landmark nodes is stored in a different layer namely landmark network layer where each of the nodes represent the location of each of the landmarks (fiducial markers in this case). Further information regarding the design, science, and workability of the markers can be found in Mantha et al. 2016, Feng and Kamat 2012, and Olson 2011. Though this step is mostly independent of specific details of the task requirement locations, the boundaries and the extent of the map is the critical information deduced. For any given floor plan, a standard procedure (as discussed above) can be adopted to create the landmark network and thus the landmark network layer. That is, the landmark network layer is independent of the task requirement layer or the user inputs regarding the task requirements. In addition, the type and size of the floor plan alone determines the size of the landmark network. It has to be noted that the landmark network layer and task requirement layer creation processes are independent and both these layers are required to create the reduced node network layer which is discussed in the next section.

The final step of network generation is to create a reduced node network layer in conjunction with the two aforementioned layers generated. The primary goal of this step is to optimize the size of the network by combining multiples nodes into a single node (i.e. removing redundant nodes). The process is manually done once for the entire floor plan and it can be subsequently used for further optimization process. It is very challenging to automate the process for all general floor plans of the buildings since some context specific assumptions (e.g. every hallway consists of a task requirement location) are required for every case. However, it has to be noted that it's a one-time process for any specific floor plan in consideration. Furthermore, it was also observed that the manual process was easily conducted for the complex case study examples considered.

In summary, each of the edges in the reduced node network is a combination of one or more task requirement and landmark locations. For example, consider end to end of the topmost hallway in Figure 2.2. There are 2 task requirement locations (T_1 and T_2) (Figure 2.2a), 5 landmark locations (L_1, L_2, L_3, L_4, and L_5) (Figure 2.2b), and finally, only 2 nodes in the reduced node network (L_1 and L_5) (Figure 2.2c). This is because, all the locations in between the end to end of the hallway can only be accessible through either of the end nodes (L_1 and L_5). That is, the edge between L_1 and L_5 represents (L_1 to T_1 to T_2 to L_5) including task requirement locations or (L_1 to T_1 to L_3 to T_2 to L_5) including task requirement and landmark locations. Thus, the nodes (or locations) in between are redundant and the entire hallway can be defined by an edge with the ends as nodes (L_1 and L_5). This process is of significance since the size of the network plays a crucial role in the computational efficiency of the optimization algorithm.

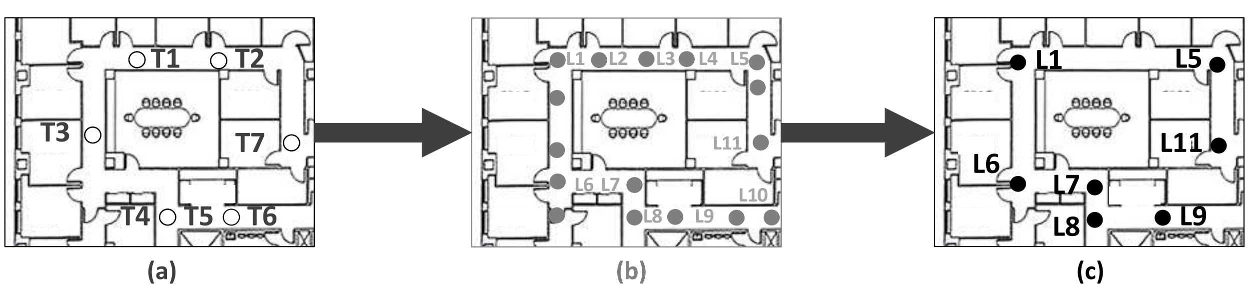


Figure 2.2 Multilayer information generation and simultaneous storage of graph network data (i.e. creating the network): (a) task requirement layer (data collection locations); (b) landmark network layer (for localization/navigation); (c) reduced node network layer

2.4.1.2 Graphical representation

The graphical representation of the reduced node network formed is defined as a weighted undirected graph $G = (V, E)$ where V refers to the nodes representing distinct locations (e.g., rooms and elevators) and E refers to the arcs or edges representing the network topology connecting any two different nodes (e.g., corridors and passages). Every edge $E_{V_iV_j}$is defined by the end nodes V_i and V_j it is connected to. $C = (c_{ij})$ represents the cost or the weight (e.g. distance) matrix. There can be various weight parameters in case of an indoor building such as distance, time, accessibility factors (e.g. based on the dimensions of the corridor), feasibility factors (e.g. based on the type of the mobility of the robot), and degree of privacy.

Based on the type of connectivity, an undirected graph can be complete or incomplete. In a complete graph, each node has an edge connecting it to all the other nodes in the network (i.e. it has all possible edges in the graph network). On the contrary, an incomplete graph has missing edges. In case of indoor built environments, generally the graph network formed is incomplete because of the physical constraints. A complete graph and an incomplete indoor graph with 4 nodes are shown in Figure 2.3a and Figure 2.3b . A complete graph network is required to generate a complete cost matrix which is necessary to determine the optimal routing for the robots.

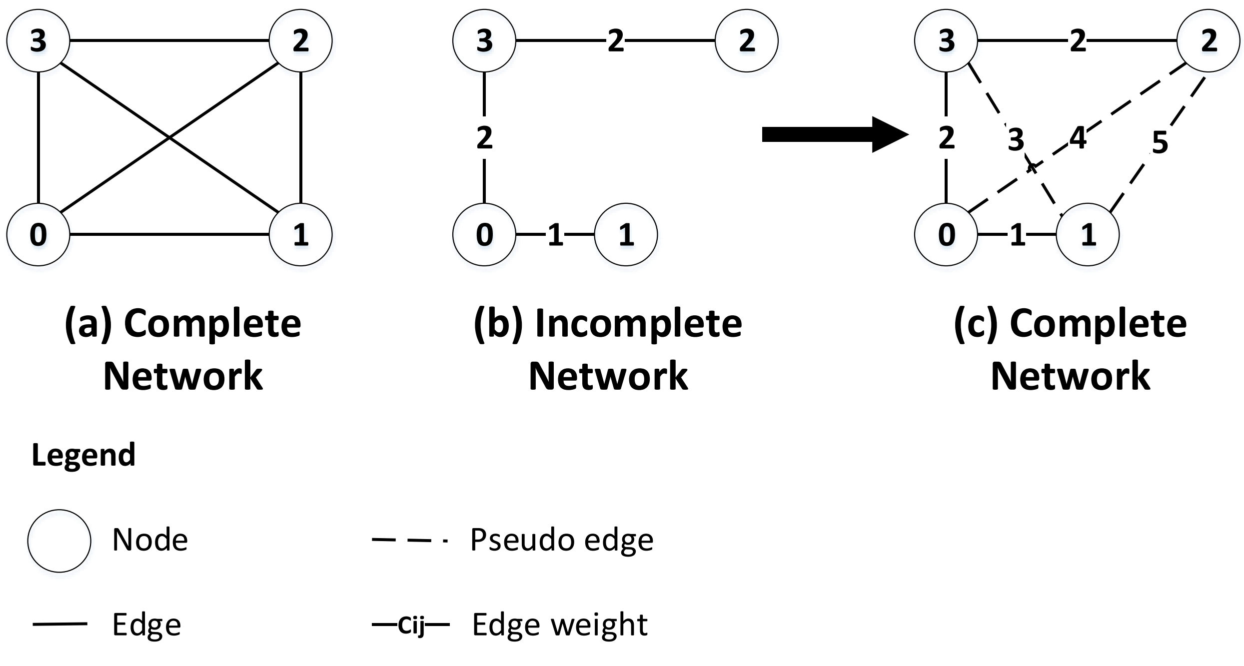


Figure 2.3 (a) Complete graph network with four nodes; (b) random incomplete network; (c) random incomplete graph network converted to a complete graph by including pseudo edges

An incomplete graph as shown in Figure 2.4b can be converted to a complete graph as shown in Figure 2.4c by introducing pseudo edges between each pair of nodes which do not already have an edge (refer to Figure 2.4). The edge weight of a pseudo edge is the shortest distance it takes to visit from one node to another. A shortest tour between any pair of nodes or nodes in a graph represent, finding a tour that minimizes the sum of the weights of the constituent edges. Dijkstra's algorithm (Dijkstra, 1959, Dorigo et al. 2008) is implemented to compute the shortest distance between specific network nodes. For example, the shortest tour between node 0 and node 10 (refer to Figure 2.4b) constitutes the edges $E_{0\,3}, E_{3\,8}, E_{8\,9}, E_{9\,10}$ and the corresponding shortest distance is 7.8 (2 + 2 + 1.7 + 2.1) units. The pseudo edge between nodes V_i and V_j represented as $\overline{E_{V_i V_j}}$. These edges are denoted by dashed lines instead of solid lines to distinguish them from the actual edges in the network.

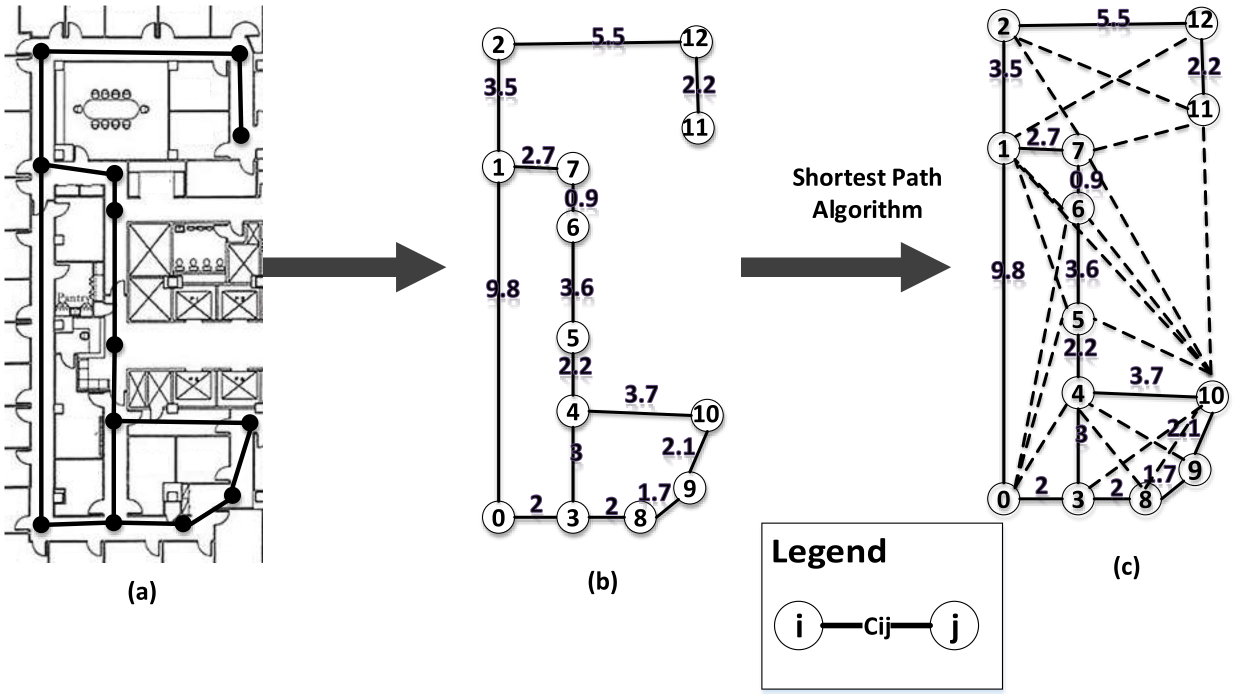


Figure 2.4 Graphical representation of the conversion from created network (reduced incomplete node network) to setting up the network (complete network): (a) reduced graph node network layer; (b) undirected incomplete graph network (G = V, E); (c) complete graph network with pseudo edges (dashed)

A similar procedure can be adopted for graph network representation of multi-level or multi-storied buildings. The network in each of the floor plans can be generated and represented as previously discussed in this section. The only addition would be to interlink all the graph networks with the help of logical physical connections such as elevators and stairs. An example representation of a network based on a two storied building is shown in Figure 2.5. After the network is formed, all the nodes and edges can be viewed on a normal 2D plan just like any other network.

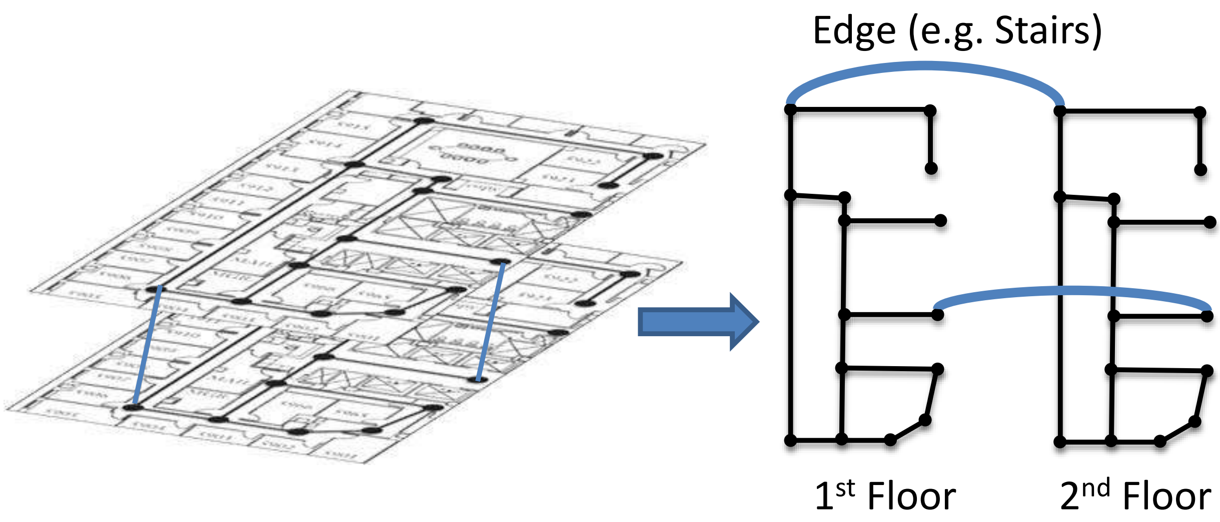


Figure 2.5 Two-dimensional graph network representation of indoor multilevel building network

2.4.2 Network Solution

Thus far, a complete graph network is generated based on the spatial distribution of tasks in an indoor environment. This step is very crucial since the basic assumption for a TSP based problem (as previously described in the problem statement section of this paper) solution is that the graph network is complete. Consider a complete graph network which consists of a total of ‘n’ nodes in it, there are $(n-1)!$ possible tours for a robot to visit all the nodes starting from and ending at the same node.

The overarching idea in the first phase (Single robot solution) is to determine the optimal route for the robot amongst all the possible alternatives. Thereafter, use the single robot solution as an initial guess to determine the multi-robot solution instead of using a random or greedy search solution like most of the existing algorithms. This has the ability to reduce the number of iterations in the process of determining an optimal solution. The two primary objectives in case of multi robot solution will then be to: a) determine the optimal way of dividing the graph network into m groups, where m is the number of robots required or available and b) determine optimal route for each of the robots to visit all the nodes in their respective groups starting from and ending at the base node. In addition to illustrating the applicability of the proposed algorithm, the case study section provides a discussion on how possible constraints in the built environment such as robot capabilities (maximum distance a robot can travel with a single charge), feasibility based on type of mobility (e.g., inability of wheeled robots to use stairs), and accessibility (e.g., blocked passage due to maintenance or emergency) are handled by the algorithm.

2.4.2.1 Single Robot Solution

A three-step methodical process is involved in determining the solution for a single robot case as shown in Figure 2.6. First, the optimal route is determined with the help of a TSP algorithm. The solution obtained might contain pseudo edges which do not represent a physical link in the building as shown in Figure 2.6b. Second, all the pseudo edges are converted, replaced, and represented with actual edges as shown in Figure 2.6c. The solution route obtained at this point will represent an actual physical route in the indoor environment. Finally, each of the edges is divided in the route to incorporate the respective task requirements.

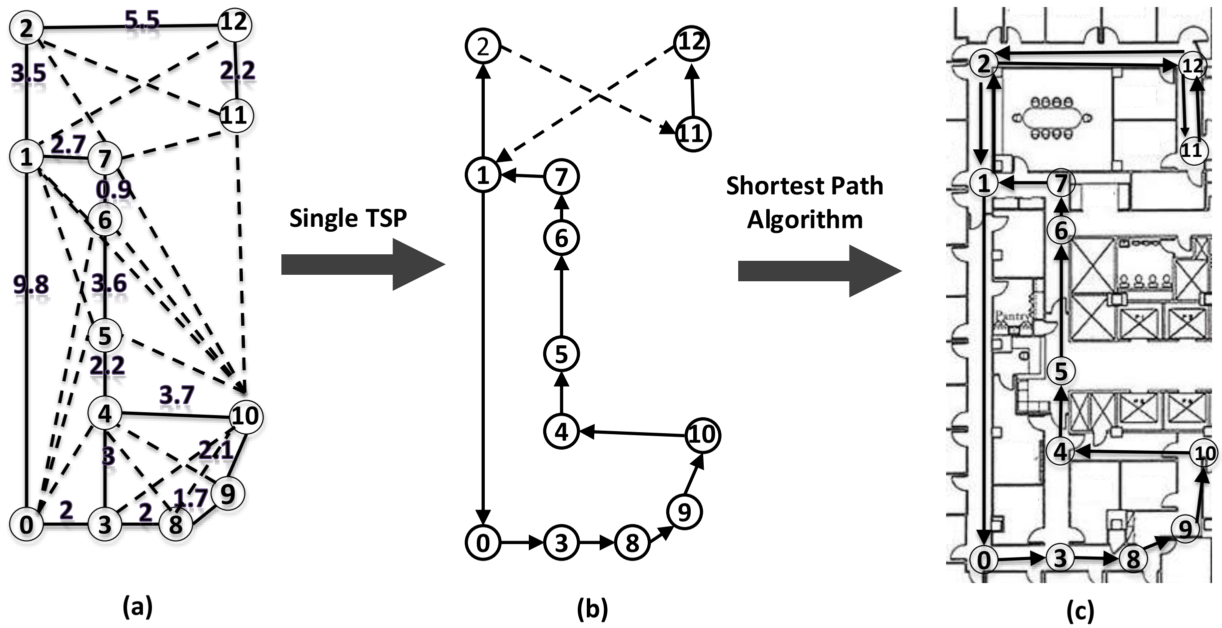


Figure 2.6 Methodical process to find the shortest path for a single robot to visit all the nodes in a graph (at least once): (a) complete graph network with pseudo edges (dashed); (b) shortest path to visit all the nodes with a single robot (dashed lines represent pseudo edges); (c) shortest path to visit all the nodes with a single robot

Determination of Optimized Tour

Starting from a base node in a network which can represent the robot charging station, the shortest tour for a single robot to visit all the nodes in the network (at least once) and return back to the base node needs to be estimated with the help of a TSP algorithm. In the case of built environments, the optimized or reduced node network is generally of reasonable size (of the order of 100) and thus the widely adopted Branch and Bound optimization approach (Ali and Kennington 1986; Gromicho et al. 1992) is employed in this study. The inputs to the algorithm are number of nodes in the network and cost matrix of the network where the cost matrix (e.g., distances) is a square matrix with size being the number of nodes in the network and the entries in the matrix are the edge weights. The outputs of the algorithm are the most optimal route for the single robot and the respective total cost of the tour. It has to be noted that the most optimal tour obtained from the algorithm might most likely contain pseudo edges because of the incomplete nature of the network. Since the pseudo edges do not physically signify anything in the indoor built environment, they need to be converted to actual edges and the entire process is discussed in the following section of this paper

Pseudo to actual edge conversion

The second step is to convert all the pseudo edges between any pair of nodes in the most optimal tour into a path which involves physical or the actual edges in the network. This is done with the help of Dijkstras shortest path algorithm (Dijkstra, 1959, Dorigo et al. 2008). To achieve this, every edge in the solution of the most optimal tour needs to be checked for the possibility of it being a pseudo edge and the conversion has to be done accordingly. For example, consider the following tour 11 → 12 → 2 --> 11 (i.e. $E_{11\,12}E_{12\,2}\overline{E_{2\,11}}$) as shown in the Figure 2.6b. The algorithm would detect the presence of $\overline{E_{2\,11}}$ as a pseudo edge, recalls the shortest path algorithm and gets 2 → 12 → 11 (i.e. $E_{2\,12}E_{12\,11}$) as the shortest tour to go from 2 to 11 and modifies the total path as 11 → 12 → 2 → 12 → 11 (i.e. $E_{11\,12}E_{12\,2}E_{2\,12}E_{12\,11}$) which consists of all the edges in the given network as represented in Figure 2.6c. This process is repeated for all such edges and the final solution which consists of all the actual edges is obtained.

Final representation

The ideal requirement is to determine a tour for the single robot to visit all the task requirement locations. Currently, the tour obtained from the previous step consists of all the actual edges in the network. As discussed in creation and setup of the network, each edge in the reduced node network virtually represents bi-layered information. The first layer is a combination of one or more task requirement locations and the second layer is a combination of one or more landmark locations which is critical for robot's localization and navigation. Thus, by dividing each of the edges in the optimal tour obtained from previous step, the final tour incorporating all the task requirement locations along with landmark locations is determined.

2.4.2.2 Multi-robot solution

Figure 2.7. shows the process involved in solving the route planning problem for the swarm of robots. Steps 1 through 3 discuss how groups (each group corresponding to each of the robots) of nodes in the network are formed with the help of an initial prior (single robot solution) and a random division. Steps 4 through 6 discuss how the optimized route in each of the groups is determined along with estimation of total cost calculations. This is iterated several times to check for different combinations of groups. The data is then stored for each of the iterations and the group combination which gives the minimum total sum (total distance traveled by all the robots) is chosen as the final solution.

Step 1: Initialization

The initialization for several of the existing algorithms such as Genetic Algorithm (GA) and Neural Network (NN) to solve mTSP problems is random or based on greedy search (where starting from a node, locally optimal node is chosen at each stage). This might be the reason why the algorithm takes more number of iterations to converge to the near optimal solution, or obtain a solution that is far off optimality. Thus, the proposed methodology utilizes the tour obtained by single robot solution as initialization or initial guess instead of a random tour. This serves as a better initial guess and helps in identifying a better solution with less number of iterations.

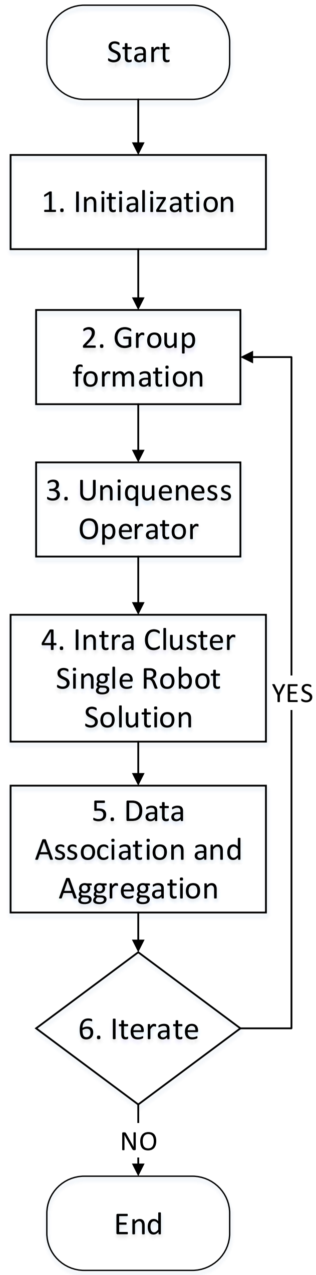


Figure 2.7 Process flow for solving the multi-robot task allocation and path planning problem

Step 2: Group formation

Yu et al. (2002) made use of cross operator as a genetic operator to compress the length of the parental individuals in the GA process to solve for the route planning problem of groups of robots. However, in this paper it is used to divide the tour obtained from single robot solution into different groups. In the context of this paper, a cross over point is an integer value of the edge number at which the division is made. For example, let the single robot solution tour for a network in Figure 2.8 with starting and ending as node 1 be: 1 → 2 → 3 → 2 → 4 → 5 → 4 → 6 → 4 → 7 → 4 → 2 → 1. If the division is made at the edge 2 → 4, it means that the crossover operator is applied at that edge and the integer value of the crossover point is 4. Thus, the minimum value of a crossover point is always 1 and the maximum is equal to the length of the tour. In addition, to create ‘x’ groups (non-empty), ‘x-1’ unique crossover operators (points) would be required.

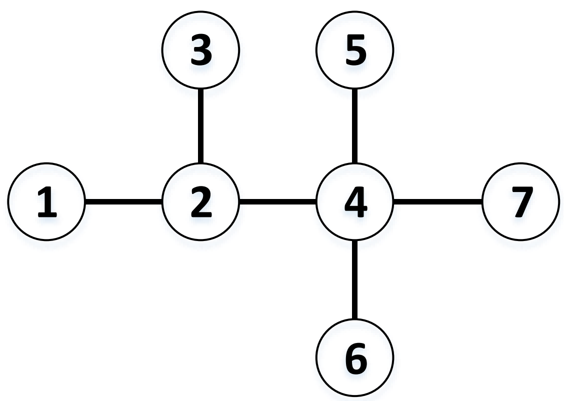


Figure 2.8 Edge representation of an example indoor network

First, generate ‘m-1’ unique integer crossover points randomly with a minimum of 1 and maximum as the total length of the tour (which is total number of edges in the most optimal tour for a single robot). That is, crossover points belong to the closed interval $[1, len(tour)]$. Based on the ‘m-1’ crossover points generated, form ‘m’ groups of nodes (which most likely are not unique). The following rules need to be adopted during the random selection of cross over points: a) Each of the pairs consisting of crossover points should be unique. If the integer values are repeated, there is a risk of forming less number of groups than required. For example, if three crossover points such as {2,6,6} are generated, instead of forming 4 groups, only 3 groups will be formed; b) Avoid repetition of the pairs already formed since the tour length will be the same. This will lead to additional computational burden and increase in inefficiency of the algorithm; c) Extra care needs to be taken if additional constraints are involved. For example, if a constraint is imposed that each of the robots needs to visit at least 2 nodes, then the difference in any pairs of the crossover points generated should be at least 2.

Consider the following tour assuming three robots (i.e., m = 3) for the network in Figure 2.8 with base node as 1. Tour (single robot): 1 → 2 → 3 → 2 → 4 → 5 → 4 → 6 → 4 → 7 → 4 → 2 → 1. Let the two (i.e., m-1) crossover points randomly generated are {4, 8}, both ϵ [1,12]. Then, the three groups generated would be as follows

1 → 2 → 3 → 2 **|** 4 → 5 → 4 → 6 **|** 4 → 7 → 4 → 2 → 1

Note: " | " represents a crossover operator.

Step 3: Uniqueness operator

It can be observed from the previous example that each of the respective groups has repeated nodes in them and also some specific nodes are observed in more than one group. This is due to the incomplete nature of the graph network which in turn is because of the physical constraints involved in the edge formation of the indoor graph network. It is important that each of the groups formed in the final multi-robot solution are unique and no two groups have nodes in common. This is to avoid planning a tour for single or multiple robots through any specific node multiple times. For instance, consider the groups generated with the help of crossover operator in the previous section. The three groups formed were C_1 = {1, 2, 3, 2}, C_2 = {4, 5, 4, 6}, and C_3 = {4, 7, 4, 2, 1}. As per the current group formation, node 2 is redundantly visited by robot 1 and robot 3. However, as per the requirement of the problem, it is only required that each of the nodes has to be visited at least once. Thus, by using a uniqueness operator, the redundancy in group formation can be eliminated.

According to set theory, the set formed by initial group of nodes is $V = \{V_1, V_2, V_3, \dots \dots, V_n\}$, where V represent a set consisting of all possible nodes in the network and n is the total number of nodes in the network. Let $C_1\ C_2\ C_3\ C_4\ \dots\dots\ C_m$ represent the final multi-robot solution where each set $C_i\ \forall\ i \in [1, m]$ C_1 consists of nodes that are assigned to robot 1, C_2 by robot 2, and so on. However it has to be noted that each of the robots might visit more nodes (other than the nodes they are assigned to) because of the network constraints. Then the obtained solution is termed as a feasible solution if the sets abide by the following rules:

- All the sets (groups) are mutually exclusive (i.e., the intersection of all the sets is a null set as shown in Eq. (4)).

$$C_1 \cap C_2 \cap C_3 \cap C_4 \dots \dots \cap C_m = \phi \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots Eq. 4$$

- The union of all sets (groups) is equal to the universal set of nodes V which consists of all the nodes in the network as shown in Eq. (5).

$$C_1 \cup C_2 \cup C_3 \cup C_4 \dots \dots \cup C_m = V \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots Eq. 5$$

  - *Intra group Uniqueness*– As the name suggests, this pertains to each of the groups being unique separately. Each of the groups formed at the end of this step represent a set since they do not possess repetition of elements (nodes). For instance, consider the previous example in the previous section (crossover operator). The groups formed are as follows:

c_1 = {1, 2, 3, 2}

c_2 = {4, 5, 4, 6}

c_3 = {4, 7, 4, 2, 1}

After applying this step, the groups would be modified as follows. This can be achieved by applying a set operator to remove the repeated nodes. That is, use functions like set() and unique() functions in python and matlab coding platforms respectively.

u_1 = {1, 2, 3}

u_2 = {4, 5, 6}

u_3 = {4, 7, 2, 1}

Note that there are still some nodes which are repeated in more than one group such as nodes 1, 2 in u_1 and u_3, and node 4 in u_2 and u_3.

- *Inter group Uniqueness* – At the end of this step, the goal is to form mutually exclusive groups (as previously described in this section of the paper). Starting from any group, remove the common nodes and form 'm' unique groups. This is a crucial step and extra care needs to be taken to check that the sum of the number of nodes in each of the groups is equal to the total number of nodes in the graph network or the set V(as described earlier in this section of the paper). For instance, consider the previous example in the previous section (crossover operator). The groups formed are as follows

u_1 = {1, 2, 3}

u_2 = {4, 5, 6}

u_3 = {4, 7, 2, 1}

After applying this step, the groups would be modified as follows

U_1 = {1, 3}

U_2 = {4, 5, 6}

U_3 = {2, 7}

It can be observed from the unique groups formed that no two robots are responsible for visiting or assigned to the same node. At this stage, the multi-robot task allocation is completed and each of the robots is responsible for unique set of nodes in the network as shown in the Figure 2.9c.

Step 4: Intra Group Single Robot Solution

The immediate next step is to calculate the total cost to visit all the nodes (by all the robots) in the network as shown in Figure 2.9d. The total cost is equal to the sum of each robot's cost in its group where each of the robots starts and ends at the base node. That is, the total tour length to visit all the nodes at least once by all the 'm' robots. For instance, consider the previous example. The unique groups formed are U_1 = {1, 3}, U_2 = {4, 5, 6}, and U_3 = {2, 7}. Assuming that the base node is 1, the total cost to visit all the nodes is summation of robot 1 cost to visit nodes {1, 3}, robot 2 cost to visit nodes {1, 4, 5, 6}, robot 3 cost to visit nodes {1, 2, 7}. It has to be noted that each of the robots start and end at node 1 after visiting the respective nodes in the group.

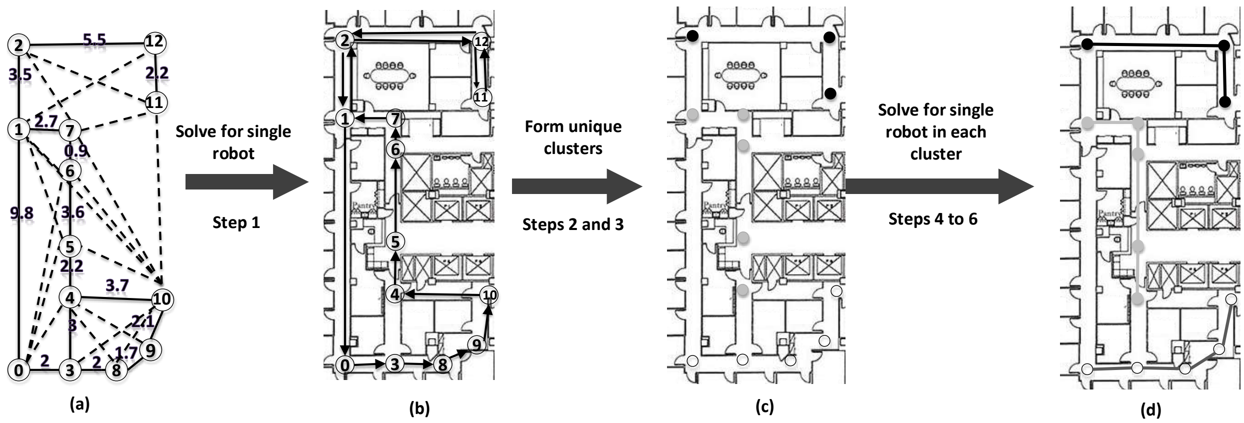


Figure 2.9 Methodical process to determine optimized task allocation and path for swarm of robots to visit all the nodes in a graph (at least once): (a) complete graph network with pseudo edges (dashed); (b) shortest path to visit all the nodes with a single robot; (c) optimized multi-agent task allocation; (d) shortest path to visit all the nodes (at least once) with multiple robots

Though the groups formed are unique, based on the intricacies of the network, a node might be visited by one or more robots. For instance, consider the part of the network shown in Figure 2.8 with nodes {2,4,5,6,7}. Let the solution of the network with two robots be U_1 = {2, 4, 7} and U_2 = {5, 6} and the respective shortest paths for the robots are R_1= 2 → 4 → 7 → 4 → 2 and R_2= 2 → 4 → 5 → 4 → 6 → 4 → 2 (2 being the base node). It can be seen that the groups formed are unique. However, due to the nature of the network formation, node 4 is visited by both the robots. Thus, except node 4 which is visited five times, all the rest of the nodes are visited only once. It has to be noted that robot returning back to its respective base station in the group is not counted as visiting same node twice. To summarize, the goal is to minimize the number of multiple visits to such nodes in the network. This is a unique challenge especially in case of indoor networks because of the topology of the network and does not occur in outdoor networks.

Step 5: Data Association and Aggregation

Significant data elements such as the total cost of the robot and robot groups are associated and stored in an important data structure in this step. Data association in this context refers to associating the information regarding the total cost with the corresponding set of unique groups (groups of nodes). Several iterations are run with different combinations of crossover points. For each of the iterations, steps 2 to 5 of the algorithm are repeated. The total cost and its respective groups are generated and stored accordingly. Generally, the higher the number of iterations the better the solution of the algorithm will be. The goal of this step is, optimized group formation based on a logical initial guess and a well-informed subsequent breakdown of steps as discussed instead of an exhaustive search of all the possible combinations of groups.

Step 6: Termination rule

Due to the randomness of the crossover points generated, it is not possible to predict if the solution will converge. Generally, the algorithm is iterated until there is no significant improvement in the optimized value or for a certain number of iterations (Florin et al.2016). Based on preliminary simulation experiments with the considered case study floor plan, the total cost variation within 5% of the optimized value was found to be a good value. After that, choose the minimum of all the total costs and the respective unique groups as the near optimal solution for the respective problem.

A pictorial representation of all the steps of the multi-robot solution is shown in Figure 2.9. The proposed methodology also gives near optimal solution but the performance of the proposed algorithm is compared and analyzed with other algorithms in the case study, results and discussion sections of this paper.

2.4.2.3 Constraints

First, the user input module needs to take the following inputs such as floor plan of the building, set of task requirement locations, location of the base depot, average linear and angular speed of the robot, and battery life of the robot. Based on the inputs provided, ideal parameters such as number of robots required (with the help of network and average speed) and optimal frequency (with the help of network solution and average speed of the robot) are determined. When robots navigate along the computed optimal tour, emergent situations may arise if any corridor, passage, or elevator is temporarily inaccessible due to construction, maintenance, or other issues. This situation is analogous to the dynamic or real-time vehicle routing problem discussed in Pillac et al. (2012). In our approach, “detour” (alternate path-planning) management is implemented when such cases arise. This does not require major modifications to the underlying algorithm. In order to compute the new optimal path from the encountered obstruction to the intended destination, the algorithm will neglect the last visited link and re-compute the optimal path in the respective robot’s group.

One of the main challenges in multi-robot task allocation is unequal division of labor where one of the robots in the network solution is responsible for most of the path compared to the other robots. For all practical purposes, this is very crucial since the frequency at which each of the nodes in their groups is visited depends mainly on the distribution of the groups. For example, consider two groups with varying tour lengths. One of the groups has 10 times the tour length as the other. This implies that, by the time the robot in former group completes a single tour, the other robot would have completed 10 tours thereby visiting the nodes 10 times within the same time frame. To address this issue, depending on the type and distribution of the network some constraints can be imposed on each of the robots such as minimum tour length and minimum number of nodes each robot must visit.

On the other hand, every robot has battery life constraints due to which it is limited by a maximum distance (or time) it can travel or before it needs to be recharged. This varies depending on several factors such as the type of the robot, the speed at which the robot is operated, tour length (distance traveled) and number of turns in the entire tour. This can be addressed by imposing a constraint during group formation where each of the robots cannot be responsible for more than a maximum threshold tour length. It is assumed in this case that the robot will autonomously charge itself at the base node (similar to existing vacuum cleaning robotic platforms) before and after completing every cycle if the situation demands.

Furthermore, there can be accessibility constraints where depending on the type of the robot (e.g., wheeled, legged, and aerial), the edges in the network might or might not be accessible. For example, a general wheeled robot cannot access stairs but can access elevators. To accommodate this, network formulation (especially graphical representation) needs to be meticulously done by storing the required information with the help of different edge weight representations. For instance, in this example, one of the edge weights can be distance or time and the other can be a binary variable with '0' and '1' representing the edge is accessible and not accessible respectively.

Another example can involve rating the ease of using a corridor or hallway (edges in the graph network) on a scale of 1 to 5 based on its width or occupancy. Once the network is generated with different edge weight representations as discussed, depending on the situational constraints, network formulation can be finalized by applying filters accordingly. For instance, to deliver mail in a building with the help of a wheeled robot, the accessibility filter needs to be applied on the general network generated. Another example can be where a robot needs to monitor real-time occupant comfort in commercial buildings through most occupied edges (i.e., with high occupants' traffic).

2.5 Case Study

A simple (close to symmetric) and an intricate (widespread floor plan containing different zones with only one or two connecting links between them) large building floor plan are chosen for simulation experiments. The graphical representation of the reduced node network for floor plan-1 along with task requirement locations superimposed on the floor plan -2 are shown in Figure 2.10 and Figure 2.11 respectively. The reduced node network consists of 28 nodes, 34 undirected edges or arcs, and 30 task requirement locations (shown as light grey shaded circles) with '0' being the base depot (or base station or charging station). Four different scenarios are chosen to show the workability of the proposed framework.

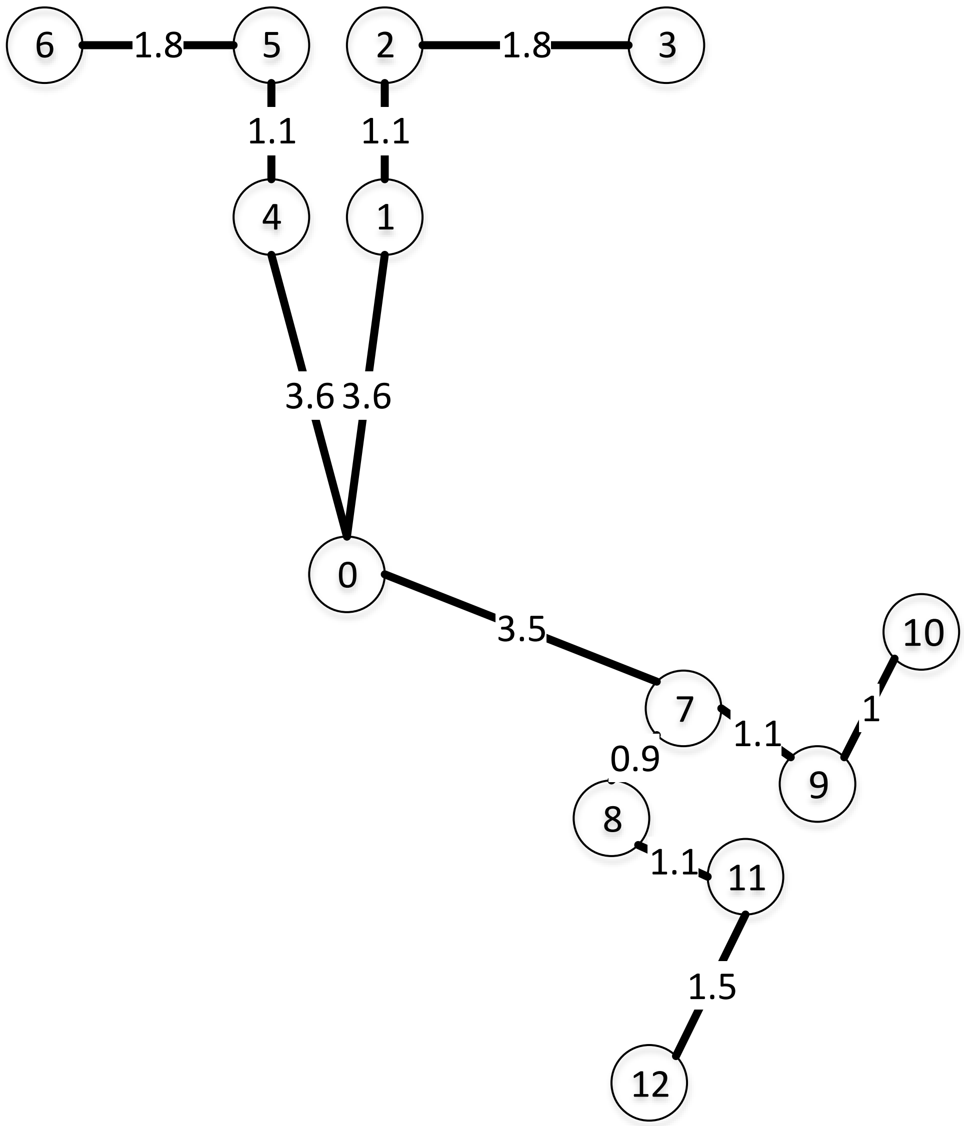


Figure 2.10 Graphical representation of the reduced node network for Floor Plan 1

2.5.1 Scenario 1: No Constraints

In this scenario, the goal is to find the shortest tour to visit all the nodes (at least once) and return back to the base station with the help of single and a swarm of robots (a group of 2 to 3 robots is considered). Initially, creation and setting up of the network is done as proposed in the network formulation section of this paper. This is very crucial since all the algorithms require a complete cost matrix as an initial step in solving the problem for a single or the swarm robotic systems. Then, with the help of the complete network and cost matrix, network solution is obtained by both proposed method and existing algorithms. The performance of the proposed method is compared with the results of the existing algorithms. The evaluation is based on the optimized tour returned by the algorithm (i.e. the total distance traveled by each or all of the robots). Furthermore, the proposed algorithm is run for only 100 iterations in comparison to GA which is run till the solution converges. This is done to show that the proposed approach takes less number of iterations to get a better optimal solution compared to GA. Though the aim is to optimize the total distance even in the case of swarm of robots, the distance traveled by each of the robots is also of great significance. This is because each of the robots is constrained by a maximum distance that can be traversed with a single charge.

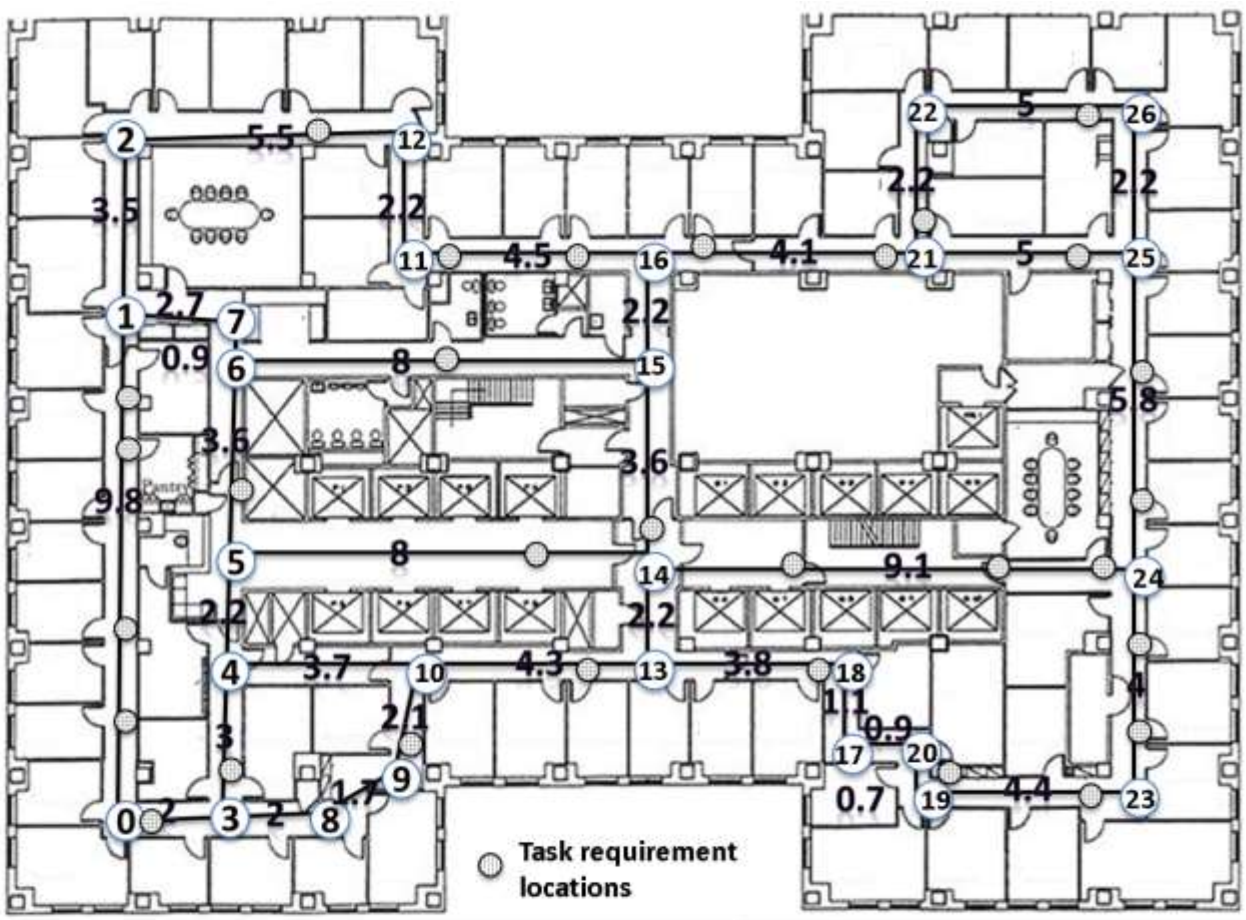


Figure 2.11Graphical representation of the reduced node network for Floor Plan 2

Consider a group of maintenance or housekeeping tasks to be done in a hotel building (Boltr 2014). The aim is to start from the base location (the maintenance or the housekeeping room), visit all the task requirement locations (with required maintenance in this case) and return back to the base location. It is highly likely in this case that at each of the task requirement location, the robot can be charged if necessary and thus constraints on distance traversed with a single charge need not be imposed on the robot.

2.5.2 Scenario 2: Distance Constraints

The goal of this scenario is exactly same as scenario 1 but with minimum and maximum distance constraints. The minimum and maximum distance constraints ensure that there is approximately equal division of labor and robot distance capabilities are met respectively in the case of swarm robots. In addition to the previous logic of the algorithm (discussed in Multi-robot solution section), an additional check is required during group formation. If the groups formed do not adhere to the distance restrictions, different set of crossover points are chosen to form the groups and the process is repeated. The termination logic is same as previously discussed. The algorithmic logic is shown as a flowchart in Figure 2.12.

Consider a search and rescue (in case of an emergency or disaster response) operation performed by a swarm of robots (Balta et al. 2016; Xu et al. 2016). The objective of such situation would be to assist humans to the nearest safest exits or search for the entire floor plan (or building) for human survivors. That is, all the robots start from a known safe location, sweep the entire accessible locations in the floor plan and return back to the start location. In this case, as one can expect, distance constraints (a minimum and maximum distance) have to be imposed on each of the robots since the robot cannot afford to lose its charge midway. If required, an additional constraint namely, real-time battery life can be added to the algorithm which can keep a track of the robots distance from current location to the nearest charging station and automatically charge itself (a feature analogous to any of the autonomous vacuum cleaning robots).

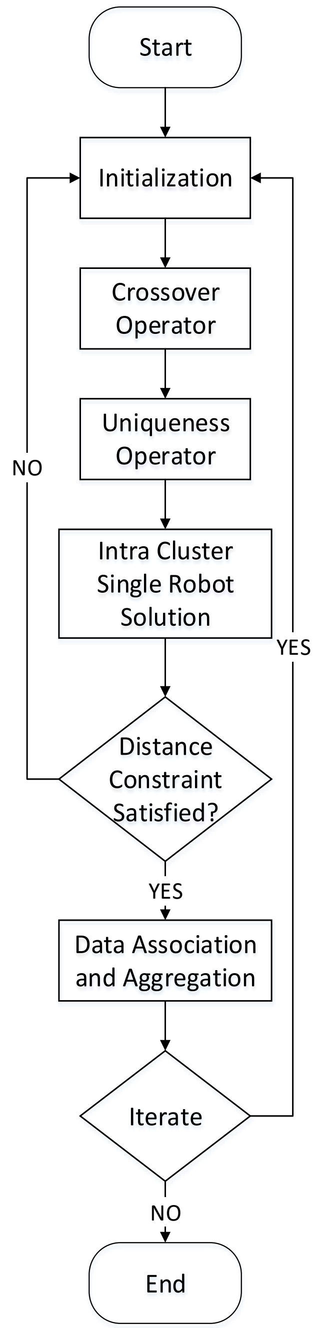


Figure 2.12 Process flow for solving the multi-robot task-allocation and path-planning problem with distance constraints imposed on each of the robots

2.5.3 Scenario 3: Tasks Requirements, Frequency, and Resource (Robots) Constraints

This scenario aims to investigate the effect that an increase or decrease in task requirement locations has on the size of the reduced node network layer. First, a *frequency constraint* is considered where given the set of task requirements, average speed of the robot, and number of robots, the maximum frequency at which the tasks can be accomplished is evaluated. This is critical for situations where tasks at a given node need to be completed repetitively and within a given time interval. It is reasonably assumed in this case study that it takes very minimal or no time to finish a particular task after visiting the task location. Second, a *resource constraint* is considered where given the set of task requirements, average speed of the robot, and the maximum required frequency at which the task location needs to be visited, the minimum number of robots required to complete the task is determined.

Consider a scenario where monitoring the occupants' comfort in a large conference room (Mantha et al. 2016), or monitoring the noise levels of a social event in a dormitory, where periodic information is required for the specific duration of the event. The goal is to monitor pre-defined set of locations in the given space in a time constrained way (e.g., every 10 or 30 minutes). In addition, it has to be also noted that there might be restrictions on the number of robots that are available for performing the task. With minor additions to the algorithm, it is capable of determining the minimum number of robots required given a desired frequency and vice versa.

2.5.4 Scenario 4: Occlusion Constraints

In this scenario, occlusion constraints are imposed to check if the algorithm will be able to reroute the robot through an alternative path to accomplish the remainder of the task requirements. For instance, consider part of the network shown in Figure 2.11 with nodes numbered from 0 to 16 with a single robot required to visit all the nodes with a constraint imposed on one of the edges (say edge between node 6 and 7), and assume that the edge is currently inaccessible due to maintenance issues. The logic of the algorithm needs to be modified to incorporate this information by introducing a temporary network update step where the inaccessible edges are removed from the network, cost matrix is updated accordingly, and the rest of the algorithm's logic remains the same. The goal is to compare the routes and corresponding tour lengths with and without the occlusion constraint imposed.

Consider a scenario of a robot guiding a visually impaired occupant to a specific room in a hospital, museum, or an airport (Feng et al. 2015). Some of the frequently observed phenomenon due to which a passage (or hallway) becomes inaccessible temporarily are maintenance related activity (in museum), security related reasons (airports), and emergent patient transport (in hospitals). This is addressed with the help of detour management as discussed previously in constraints section.

2.6 Results And Discussion

2.6.1 Scenario 1

The results obtained by the algorithm are compared with the most widely implemented algorithms such as Genetic Algorithm (GA) and K-Means Clustering. The usual K-Means clustering where k centroids are defined (one for each group or cluster) and set of nodes closest to these centroids are assigned to one particular group (or cluster) thereby forming k groups is employed (Pandoricweb 2011). The GA adopted is a multi-chromosome genetic representation technique developed by Kiraly and Abonyi (2015). A novel method was introduced where a new set of genetic and complex operators were used for mutation and algorithmic convergence respectively. For both K-Means and GA, the output taken from the algorithms is the groups (or clusters) formed. Based on the groups formed, the total sum of the tour lengths (numerical value) for each of the algorithms (including the proposed one) is compared and the performance is evaluated accordingly. The GA however cannot be directly employed for the current network because of the missing edges or since the network is incomplete. Pre and post processing of the network such as initial network formulation and final representation (as discussed previously) needs to be done before the algorithm can be implemented to get the network solution. It has to be noted that K-Means clustering does not take into account the topology of the network (i.e. it considers that any node is accessible to another and the distance between the nodes is the geometric distance between them). However, this is considered to show that such algorithms cannot be directly implemented for indoor networks as done for outdoor networks. The results of the analysis without any constraints for both the floor plans 1 and 2 are shown in Table 2.3, Figure 2.13 and Table 2.4, Figure 2.14 respectively.

Table 2.3. Simulation results without any constraints for floor plan-1

		Distance (in units)			Nodes assigned to the robot				
No of Robots	Method	Robot 1	Robot 2	Robot 3	Robot 1	Robot 2	Robot 3	Total distance (in units)	Iteration
1	Iterative	44.2	0	0	0-12	NA	NA	44.2	-
	GA	44.2	0	0	0-12	NA	NA	44.2	-
2	Iterative	13	31.2	0	0,1-3	0,4-12	NA	44.2	10
	GA	26	18.2	0	0,4-6	0,1-3,7-12	NA	44.2	176

3	Iterative	13	13	18.2	0,1-3	0,4-6	0,7-12	44.2	10
	GA	13	13	18.2	0,1-3	0,4-6	0,7-12	44.2	473

Note: ‘0’ is considered to be the base node.

Table 2.4 Simulation results without any constraints for floor plan-2

		Distance (in units)			Nodes assigned to the robot				
No of Robots	Method	Robot 1	Robot 2	Robot 3	Robot 1	Robot 2	Robot 3	Total distance (in units)	Iteration
1	Iterative	85.2	0	0	0-26	NA	NA	85.2	-
	GA	88	0	0	0-26	NA	NA	88	-
2	Iterative	4	85.2	0	0,3	0,1,2,4-26	NA	89.2	100
	GA	15.6	85.2	0	0,3,8,9,10	0,1,2,4-7,11-26	NA	100.8	2697
	K-Means	50.2	66.4	0	0-12	0,13-26	NA	116.6	-
3	Iterative	8	14.4	85.2	0,3,8	0,4,5	0,1,2,6,7,9-26	107.6	100
	GA	24.2	15.6	80.6	0,1,3,4,5,6,7	0,8,9,10	0,2,11-26	120.4	3138
	K-Means	20.9	48.3	66.4	0,3,4,5,8,9,10	0,1,2,6,7,11,12,16	0,13-15,17-26	135.6	-

Note: ‘0’ is considered to be the base node.

As one can expect, the single robot solution is always better or same as the approximation algorithms (e.g., GA) considered since it is the most optimal solution. As can be seen and discussed earlier, the optimal solution obtained by the K-means algorithm is highly off from the optimality. For a simpler floor plan (floorplan-1 as shown in Figure 2.10) proposed method and GA return the same result (by comparing the sum of distances traveled by the robots) but the proposed method gets the solution with less number of iterations (10 vs 176 for 2 robots and 10 vs 473 for 3 robots). However, the proposed method works better for complex floorplan

(floorplan-2) in all the cases even with very less number of iterations (100 vs 2697 for 2 robots and 100 vs 3138, for 3 robots) as shown in the Table 2.4 and Figure 2.14 also shows that the task allocation in case of swarm of robots is not uniform (high variations in distance traveled by each of the robots). It has to be noted that the aim of the algorithm is to optimize the total distance traveled by all the robots and not to uniformly distribute the tasks among the robots. It turns out that, in this particular case, the non-uniform distribution gives better results compared to the uniform one when considering sum of the distances traveled by the robots. This is mainly attributed to the location of the base node in the network. For example, if the base node is located symmetrically in the network, the resulting task allocation will be more uniformly distributed. Since the floor plan 1 considered is slightly more uniformly distributed, the task allocation has better uniformity in comparison to the floor plan-2 as shown in Figure 2.13 (floorplan-1) and Figure 2.14 (floorplan-2).

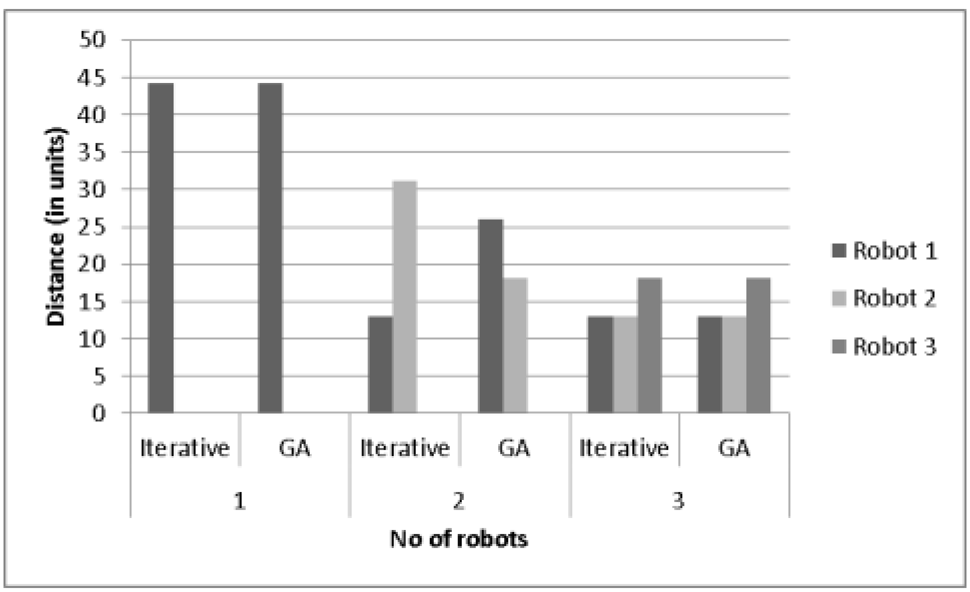


Figure 2.13 Performance comparison of proposed (iterative) method with other algorithms (without any constraints) for Floor Plan 1

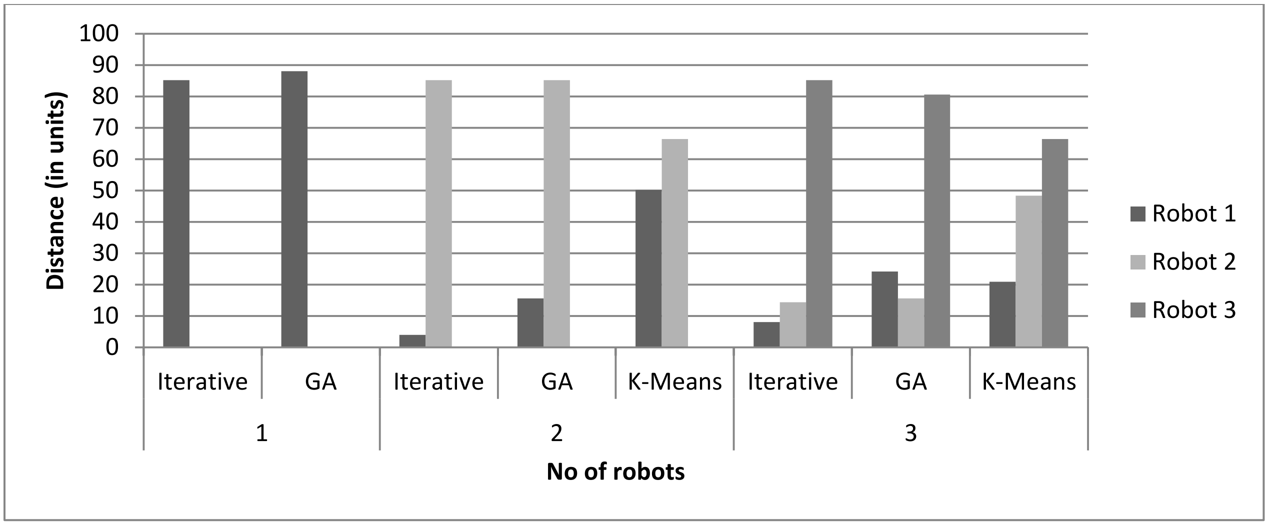


Figure 2.14 Performance comparison of proposed (iterative) method with other algorithms (without any constraints) for Floor Plan 2

The current solution will result in robots visiting the task requirement locations with extreme frequencies. For instance, consider the solution for iterative method with two robots in Table 2.4. By the time robot 2 visits all its respective nodes (1, 2, 4-26), robot 1 would have visited nodes 0 and 3 approximately 10 times. Thus, the tasks between these nodes will be completed 10 times faster than the rest of the tasks. Though it might be favorable for some situations where selective set of tasks have more weight than others, it is not ideal for all the scenarios. Similarly Robot 1 travels 50% more distance in each tour compared to that of robot 2 for iterative method with two robots as shown in Table 2.3.

2.6.2 Scenario 2

Considering the unequal division of labor (one robot travelling a lot of distance compared to the other) and battery constraints of the robots (maximum distance a robot can travel with a single charge), a minimum and a maximum distance each of the robots must travel is imposed as a constraint. For the purpose of this case study, the minimum and the maximum distances are 30 and 70 respectively. The results of this scenario are shown in Table 2.5. and Figure 2.15. The results from the table show that the iterative method gave better results with less number of iterations in both cases. It can also be observed from Figure 2.15 that the distribution of tasks is comparatively a lot better than the case where there were no constraints. Thus, imposing distance constraints is necessary if equal division of labor is preferred among the different robots to optimize travel and charge time.

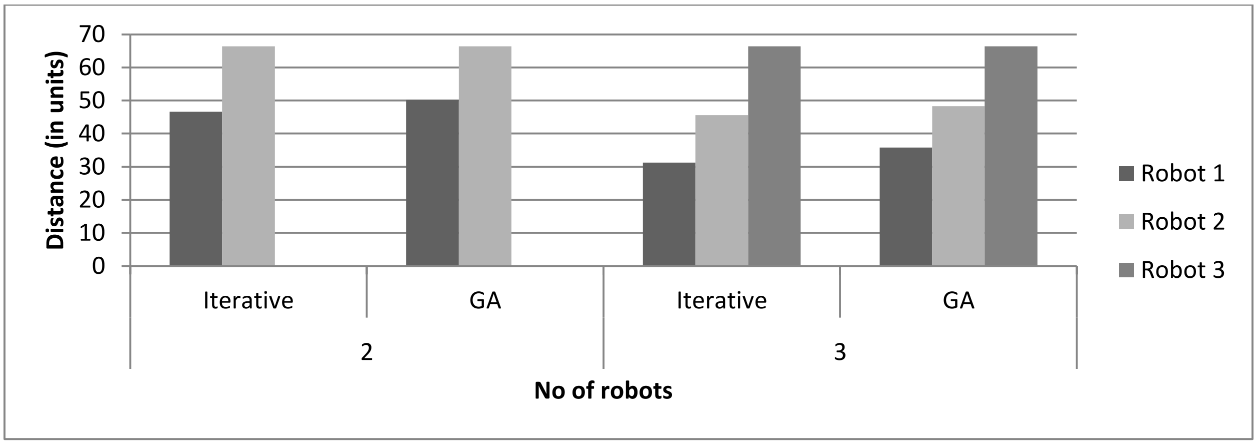


Figure 2.15 Performance comparison of proposed (iterative) method with GA (with distance constraints for Floor Plan 2)

Table 2.5 Simulation results with minimum and maximum distance constraints for floor plan-2 With Minimum (30 units) and Maximum (70 units) Distance Constraints

		Distance (in units)			Nodes assigned to the robot				
No of Robots	Method	Robot 1	Robot 2	Robot 3	Robot 1	Robot 2	Robot 3	Total distance (in units)	Iteration
2	Iterative	46.6	66.4	0	0,3,8-10,13-26	0,1,2,4-7,11,12	NA	113	100
	GA	50.2	66.4	0	0-12	0,13-26	NA	116.6	993
3	Iterative	31.2	45.6	66.4	0,1,2,4-7	0,11,12,15,16	0,3,8-10,13,14,17-26	143.2	100
	GA	35.8	48.3	66.4	0,3,8-10,13,17,18,20	0,1,2,4-7,11,12,15	0,14,16,19,21-26	150.5	379

Note: ‘0’ is considered to be the base node.

2.6.3 Scenario 3

The size (i.e. number of nodes) of the reduced node network is necessarily not proportional to the number of task requirement locations. Instead, it can be said that the spatial distribution of tasks play a significant role in deciding the size of the final reduced node network. The results of a similar case study considered are shown in Figure 2.16. The graphs show how a 5/14 tasks containing task requirement layer got converted into a 11/7 node containing reduced node network layer respectively. Additionally, the reduced node network layer converted from the sparsely distributed task requirement layer (Figure 2.16a and Figure 2.16b) corroborates the discussion regarding automation of the reduced node network layer generation process.

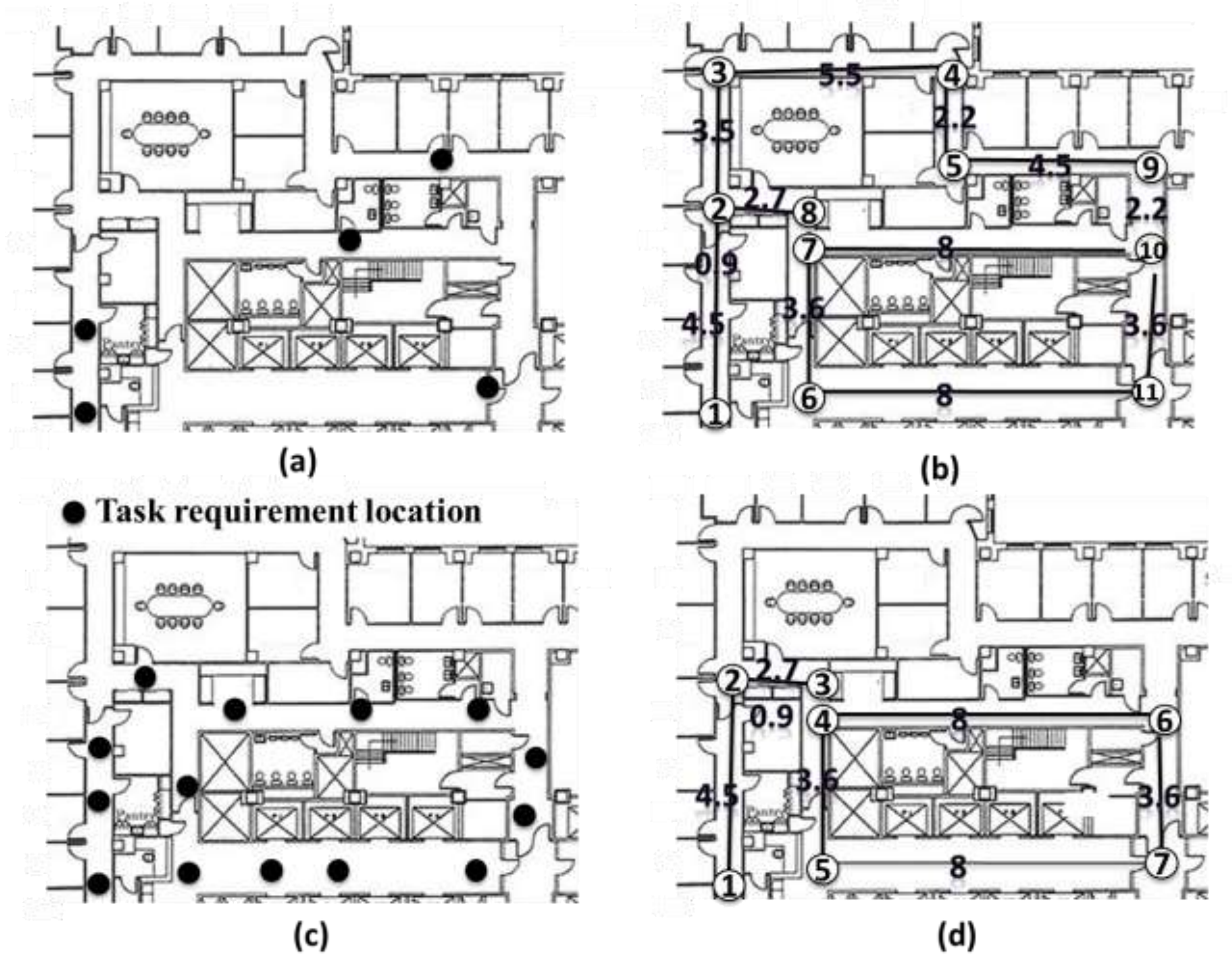


Figure 2.16 Transformation of different distributions [sparse (a) and dense (c)] of task requirement layers into respective reduced node network layers (b and d)

Table 2.6. shows the results of applying iterative algorithm on the reduced node network generated in Figure 2.11 with maximum distance constraint in such a way each of the robots travel approximately similar distances. It can also be observed that, the total distance traveled by the robots increase with increase in number of robots. However, it has to be noted that each of the robots will visit the nodes concurrently thereby increasing the frequency at which the nodes are being visited. That is, the task requirement locations will be visited more frequently in case of multiple robots than with a single robot. For example, if there are two robots available, the

tasks can be accomplished repetitively with a frequency of 21 minutes compared to that of 28 minutes using just one robot. This can be very useful for estimating the frequency at which the task locations can be visited given some of the parameters such as task requirement locations, average speed of the robot, and available number of robots (limited resources). Potential applications can be in small residential or commercial office buildings where resources are limited and time is not a big constraint.

Table 2.6 Interdependency between the number of robots and the frequency of task accomplishment (applied with distance constraints) for floor plan-2.

No of Robots	Method	Max Distance Constraint (in units)	Total Distance (in units)			Average Velocity of Robot (units/min)	Frequency of Task Accomplishment (in minutes)		
			Robot 1	Robot 2	Robot 3		Robot 1	Robot 2	Robot 3
1	Iterative	NA	85.2	NA	NA	3	28.4	NA	NA
2	Iterative	75	46.6	66.4	NA	3	15.5	22.1	NA
3	Iterative	65	50.4	62.8	46.6	3	16.8	20.9	15.5

Conversely, if time is a crucial factor and there is infinite number of resources available, the table can give the optimum number of resources required for a desired frequency. This scenario mostly arises in case of large warehouses where time or frequency at which tasks are completed may be of the essence. For instance, assume that the desired frequency is to visit each of the task locations at least every 25 minutes. It can be observed from Table 2.5 that a frequency of 25 minutes cannot be achieved using a single robot and requires at least two robots. The results also show that there is no improvement in frequency when the robots increase from two to three. This is because of the location of the base node and size of the network. The location of the base node plays an important role in optimizing the frequency of task accomplishment. For example, if the base node is located in the center of the network, a better frequency for task accomplishment is possible. To that extent, the network size also plays a dominant role and single base node will not be able to cater to the frequency requirements. To address these issues, either strategic placement of a single base node or multiple base nodes are required depending on the situation.

2.6.4 Scenario 4

The results of this scenario with and without occlusion constraints are shown in Figure 2.17. As it can be expected, the total cost of the tour with occlusion constraints is higher than the other. This is because of increase in complexity and incompleteness of the network. The results also show how the algorithm optimized and rerouted the robots path at the same time. Though in this case, imposing an occlusion restriction on the network increased the cost, it need not be the case every time. For example, imposing the same occlusion restriction on the edge between nodes 0 and 1 (instead of edge between nodes 6 and 7) will have the same result since the optimized final tour (without constraints) did not have the edge between nodes 0 and 1 in its initial solution and hence the solution will not be affected by that particular edge. Similarly, for the rest of the edges which are not present in the initial solution, the analysis would remain the same. On the other hand, if the restriction is imposed on any of the edges which are present in the initial solution, the tour length is usually bound to increase.

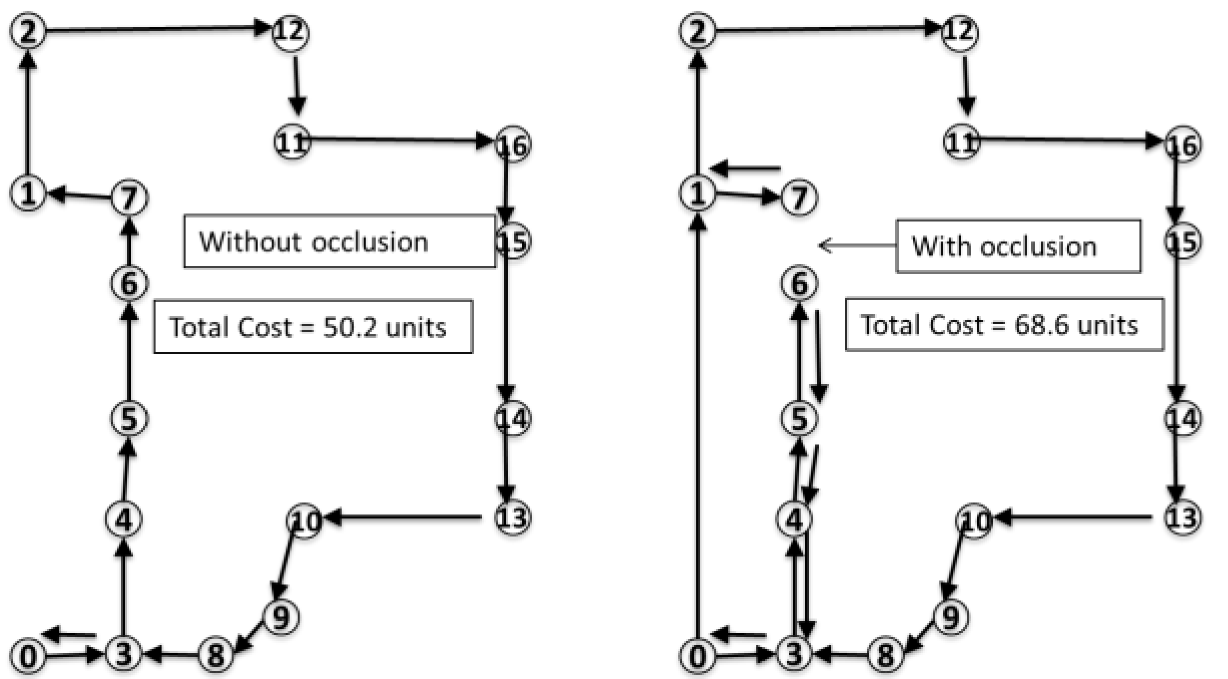


Figure 2.17 Results of the single-robot solution with and without occlusion restriction

CONCLUSIONS

Some of the key contributions of this proposed work are: a) Identified the research gaps in indoor route planning for single/multi-robotic systems; b) Provided a generalized framework (which can be applied for any indoor robotic service network irrespective of the application) for optimized route planning and task allocation in complex indoor environments for single and swarm robotic applications; c) Performed four different scenarios (along with relevant practical civil engineering applications), compared with existing algorithms, and analyzed with and without constraints such as occlusions (due to maintenance related issues) and travel distance restrictions. The results demonstrate the workability of the proposed framework for real world applications and how the approach adapts to some of the pragmatic constraints; and d) Identified the limitations of the current approach and discusses further research areas that needs investigation.

Scenario analysis was performed with an objective function to optimize the total distance traveled by all the robots. The results have shown that some of the existing algorithms (e.g., K-means clustering) do not work very well for indoor environments as they do for outdoor environments. This is because K-means clustering do not take into account the network topology (information regarding the edges in the network). It was found that the proposed method performs better (compared to other algorithms) in almost all the cases considered. However, the result of the optimal solution has unequal division of labor (one robot performing most of the tasks compared to the other) because of the location of the base node in the network. This can be addressed by imposing minimum and maximum (that robot can travel with a single charge) distance restrictions on each of the robots. This ensures the practical viability and uniform distribution of tasks among the group of robots.

Typically, the frequency of tasks visited will increase with increase in number of robots. However, it was observed that this can be significantly affected by the location of the base node in the network and the size of the network itself. For instance, if a robot that takes 10 minutes to reach its nearest group node, then the desired frequency can never be less than 10 minutes. To address these issues, in addition to optimizing the task allocation and routing of each of the robots, either strategic placement of a single base node or multiple base nodes are required depending on the situation. The results also show that the total distance traveled by the robot need not necessarily increase with occlusion constraints (e.g. blocked paths).

The proposed approach is applicable to numerous civil engineering applications such as scheduled (or regular) maintenance, structural health monitoring, building energy monitoring, indoor occupant-tracking (or wayfinding), search and rescue as discussed previously. However, there is a strong need to further investigate the privacy (intrusion of work or personal space), social (some people may not be comfortable working alongside robots), and technology related issues (the current state of the art technology may not be ready for complete autonomous swarm of robots). Though the privacy issues can be overcome by utilizing the proposed approach in places like warehouses and data centers, where there is very less occupant traffic, further research efforts to delve more into addressing the social and technology issues is required.

To summarize, the limitations of the proposed methodology are a) Predefined location selection (randomly choosing) of the base node and b) Single base station (start and end node location) is considered for the optimization algorithm c) Privacy, social, and technology issues. To address these limitations, further planned work includes a) Optimization of initial start location(s) (or the depot(s)) instead of random selection; b) Extending the existing framework to multiple start and destination depots with and without feasibility, accessibility, and occlusion constraints. That is, different robots starting at different start depots and all ending at their respective nearest destination; c) Physically test these robotic systems in several social settings to investigate more regarding these issues.

Chapter 3

Localization and Navigation Algorithms for Indoor Robotic Data Collection and Simulation

3.1 Introduction

Buildings are responsible for 40% of the total energy consumption in the Unites States (US) (US EIA 2014). According to the US EIA (2014), future predictions show that these energy demands will continue to increase indicating a significant need for innovative efforts to optimize the energy consumption of buildings. The United States Green Building Council (USGBC) suggests that retrofitting is the process through which buildings can be efficiently operated to achieve high performance (Coyle, 2014). Retrofitting involves changing or modifying building systems to achieve an improved and desired energy performance (Jaggs and Palmer, 2000). One of the primary goals of this process is to maintain comfort of the occupants inside the building while ensuring efficient performance of the building systems.

Commissioning is a process in which all the building systems are tested to meet the desired quality standards and the expectations of the building's owner and stakeholders (Grondzik, 2009). Though every building undergoes commissioning process initially, energy retrofits are important during operation and maintenance phase because buildings are subjected to system degradation, change in use, and unanticipated faults over time (Heo et al. 2012). A holistic understanding of several factors such as actual performance of building systems, energy use behavior of the occupants, and weather conditions is required to assess the energy performance of the buildings (Ma et al. 2012).

In addition, Ma et al. (2012) emphasize that analyzing building energy data, identifying areas with energy wastages, and understanding building energy use play a crucial role in the complex building energy retrofit problems. The significance of energy monitoring, performance management and understanding building operations along with the need to educate building managers, owners, and operators is also stressed in the energy information handbook developed by Lawrence Berkeley National Laboratory (LBNL) as a part of a Department of Energy (DOE) sponsored project (Granderson 2013).

The International Energy Agency (IEA) intends to develop new ways of collecting and analyzing data to better understand and predict the energy use in buildings with the help of Annex 53:Energy Conservation in Buildings and Community Systems (ECBCS), "Total Energy Use in Buildings: Analysis and Evaluation Methods" (IEA 2012). The DOE emphasizes the energy performance of buildings to help make engineering and economical decisions, and consequently evaluate the energy efficient practices and products in buildings (DOE 2017). It is thus apparent from this literature that a significant amount of data needs to be continuously collected, managed, and analyzed in a building at the room and floor level in order to effectively optimize energy use in buildings while maintaining comfort of the occupants.

In an effort to address the need for intelligent data collection in buildings, this paper introduces an autonomous data collection process that uses a mobile indoor robot equipped with a single set of sensors. The robot is capable of autonomously navigating in a known indoor environment with the help of onboard sensors (for data collection of various indoor ambient parameters), onboard computing capabilities, and an RGB camera (for localization, navigation, and drift correction). The entire process of autonomous robot navigation along with the localization, data collection, and geotagging are discussed in detail in the methodology section of the paper. One of the main advantages of the proposed data collection method is that it eliminates the need for instrumenting different locations in existing buildings with the same set of sensors, which is time-consuming and might result in integration issues with existing older building systems (Akyildiz and Kasimoglu, 2004). In addition, the proposed method of data collection helps identify any faulty and improperly calibrated Building Automation System (BAS) sensors in newer buildings as demonstrated by the authors' preliminary research (Mantha et al. 2015a; Mantha et al. 2016). Finally, a case study is conducted to demonstrate the utilization of ambient temperature data collected by the mobile robot to choose an optimal (e.g., most energy saving and most economic) retrofit option.

3.2 Review Of Data Collection Methods In Buildings

Data collection in buildings has progressed significantly in the past few years because of the advancements in technology. In the early stages, inspectors used to collect data manually by inspecting different locations in the building, a process considered to be extremely tedious, time consuming and generates limited amounts of data which limits it is practical applicability in large buildings (Swartz et al. 2012; Raftery et al. 2011; Wang et al. 2010a; Sacks et al. 2005). With the advent of wired and wireless sensors, in the newer buildings, sensors are installed, calibrated, and integrated with building systems before the operation and maintenance phase as part of a BAS. This BAS allows for automated monitoring and control of various building systems such as Heating Ventilation and Air Conditioning (HVAC), lighting, and plug loads (Granzer et al. 2010). The location in the building where the data is aggregated for storage and further investigation is typically called the BAS room (Park and Hong, 2009). The need for such information becomes more pronounced in existing buildings where informed energy retrofit decisions need to be made (Ma et al. 2012). For example, most of the energy simulation software(s) require data to create a calibrated model and use this to evaluate proposed design changes and/or retrofit measures (Evangelisti et al. 2015). This process is highly expensive and inconvenient because of the need to install, calibrate and uninstall the sensors to collect and analyze the data required. That is, if an energy retrofit decision has to be made for an existing facility without an installed sensor network, then, sensor installation, data collection, and sensor de-installation has to be done for obtaining the data.

The following are some of the key challenges to collect data in existing older buildings with wired and/or wireless sensors. a) High Costs: Economically infeasible for larger buildings (e.g., dormitory buildings, academic institutions, and office spaces) where there are a large number of rooms that need to be monitored because of the cost of the material and installation (Nourollah, 2009; Dermibas, 2005). b) Complex Design Requirements: Careful consideration of the disturbances in ambient indoor environment plays a crucial role in the design process of the wired networks in buildings (Wang et al. 2010b). c) Manual Supervision: The sensor wires installed in certain locations of buildings (e.g. warehouses and store rooms) might need frequent supervision because of the threat from rodents (Wang et al. 2010b). d) Tedious Maintenance: The wired/wireless sensors require intense calibration and maintenance process e) Limited Coverage: The extent of space that can be monitored is limited because of the higher costs involved (Kim and Lynch, 2011; Vlissidis 2008; Dermibas, 2005). f) Other Issues: Wireless systems suffer from power consumption, scalability, and limited information storage capacity issues limiting the extent and quality of the data that can be collected (Swartz et al. 2012;

Bhadauria et al. 2011). In addition, wired/wireless sensors mounted in old heritage buildings also have a direct damaging impact on the underlying surface thereby seriously affecting the historic value of the building or the monument itself (Raffler et al. 2015). Further details about these data collection methods along with a thorough analysis of the respective characteristics can be found in Mantha et al. (2015b). Although technological advancements have progressed tremendously in the realm of data collection in buildings, there is still a strong need to further investigate efficient and economic ways of gathering critical actionable data in these buildings.

3.3 Research Objectives

The primary contribution of this study is the development of technical methods to utilize mobile robots for collecting ambient data (which represent the building state) in old existing buildings without a BAS and subsequently develop a framework that utilizes the collected data to make informed retrofit decisions. The following is a brief overview of the proposed approach along with the objectives of this research. Detailed description of each of these processes is described in the methodology section of this paper. To achieve this, a mobile robot should be able to navigate to various strategic locations in a building space within a stipulated time frame to get a valid snapshot of the building's state. This is referred to as indoor localization and navigation in the mobile robotics research community (Bar-Shalom et al. 2004; Thrun et al. 1999; Burgard et al. 1998). Indoor localization, navigation, drift correction, and collision avoidance are some of the key aspects of this robotic based data collection method and are described in detail in this paper. Data is collected and stored with the help of onboard sensors and a computer on the mobile robot. The sensor data thus collected has to be cleaned before it can be consumed for further processing in building energy models such as EnergyPlus and eQuest. Data cleaning refers to the process of removing or replacing inadequate or additional data polled by the sensor and stored in the database (Lin 2006). For example, sensor data might contain multiple temperature readings associated with a single time stamp. It is essential to remove duplicates and ensure the completeness, consistency and uniformity of the collected data set. Finally, the cleaned sensor data (ambient data) is used to calibrate the building energy simulation model and evaluate the effect of different energy retrofit options on the total annual energy demand. Thus, the primary objectives of this study are to:

1. Introduce a novel concept of using mobile robots for collecting ambient parameter data in existing buildings.

2. Discuss in detail the localization, navigation, and drift correction algorithms for the two robotic platforms developed. These two platforms differ in terms of navigation, drift correction, and ability to reduce human operator intervention as follows:

    a. *Predefined path mode*: Where the mobile robot navigates a pre-defined path using a continuous drift correction approach.

    b. *Dynamically configurable path mode*: In this mode, the path traversed by the mobile robot can be dynamically configured by the user.

3. Investigate the practical feasibility of the proposed navigational algorithms on a real physical system in an indoor dynamic environment.

4. Perform a case study to show how the data collected (by the mobile robot) can be processed, analyzed and subsequently used for building energy retrofit decisions as a proof of concept.

3.4 Design Of The Mobile Robot

Recent studies have focused on autonomous indoor robots for achieving various tasks such as assisting the elderly people (e.g., the health care industry) (Linner et al. 2015), butler robots (e.g., the hospitality industry) (Boltr 2014), and helping visitor to navigate indoors (e.g., in large commercial buildings) (Biswas and Veloso 2010). Though autonomy is common in all the aforementioned robots, applications, sensors used, actuation mechanisms, and the respective algorithms differ a lot amongst each other. For example, Linner et al. (2015) uses Simultaneous Localization and Mapping (SLAM) for navigation whereas Biswas and Veloso (2010) utilize the WiFi infrastructure for localization and navigation.

Localization, navigation and drift correction are some of the main challenges for such task oriented robots (Biswas and Veloso 2010). Thus, one of the main contributions of this paper is to develop algorithms that help robots achieve autonomy in the data collection process. These algorithms are governed by the type of data that needs to be collected (accordingly the type of sensors to be placed on the mobile robot), frequency of data collection (how frequently the data needs to be collected at every location), waiting time at each location of data collection, the number of mobile robots required to monitor (depending on the size of the buildings) the entire building, optimizing the travel time, and path.

The robotic platform, equipped with the iCreate base (widely known as Turtlebot) is chosen as the mobile data collection platform and sensors such as Cozir® CM 0199 (for temperature, humidity, and CO_2 levels), HOBO U12 (for light and occupancy levels), Lutron (for natural light levels) and NinjaBlocks (for air speed) are used for the data collection. Figure 3.1 shows the mobile robot with the following components 1) TurtleBot – for navigating the indoor environment; 2) On-board netbook – to communicate with the TurtleBot; 3) RGB Camera - for the TurtleBot to detect fiducial markers, localize, and estimate its relative pose in the indoor environment; 4) Remote laptop - to execute the corresponding autonomous navigation programs on the on-board netbook; and 5) Sensors – for monitoring and data collection of various ambient parameters.

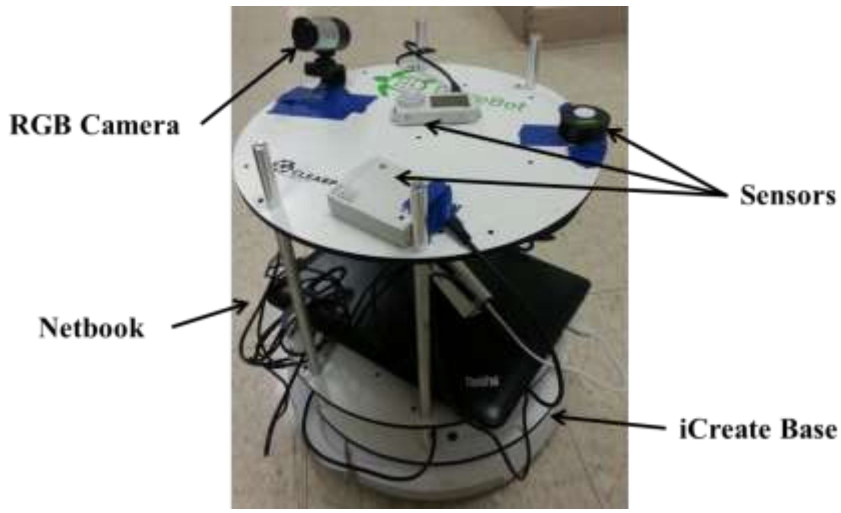


Figure 3.1 Components of mobile robot used for ambient data collection in buildingsS

3.5 Methodology

In order for the mobile robot to autonomously navigate indoors and ambient data, it needs to: 1) localize in the indoor environment, 2) navigate to the intended data collection locations, 3) collect the respective data (such as temperature, humidity, and light intensity), and 4) geo-tag (record the collected data with the physical location) for further analysis. Figure 3.2 shows the flowchart of the sequential steps involved in the robotic data collection process. Though the steps are sequential, the cyclic representation signifies that the data collection task is performed till the data is collected at all the required locations in the building. That is, after the data is geotagged at a particular location, the robot continues its journey to the next data collection location. Each of these steps and the associated algorithms is described in the subsequent paragraphs.

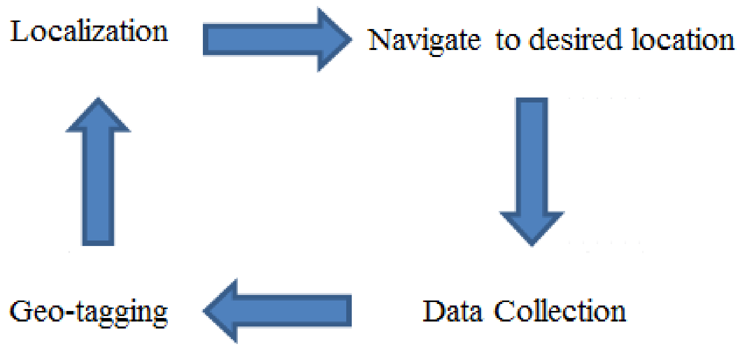


Figure 3.2 Sequential steps for robotic data collection in buildings

3.5.1 Localization

In the current context, localization is defined as the robot's ability to identify its current location in a given indoor environment setting (Levitt and Lawton, 1990). For example, a robot being able to recognize its current location to be in room 201, or knowing its location and orientation in the global coordinate reference system. Several indoor localization techniques have been explored previously. Some of the non-vision based techniques include Wireless Local Area Network (WLAN), Radio Frequency Identification (RFID), Ultra-Wide Band (UWB) and Bluetooth. Wifi is an economical solution because most of the existing infrastructure consists of wireless nodes required for localization. However, it suffers from the significant error in localization accuracy (Montañés et al. 2013; Torres-Solis et al. 2010). This might lead to data association problems where data collected by the robot at a particular location can be associated with an adjacent location in the building. Bluetooth and RFID based localization tends to be expensive, time consuming and also have space constraints because of the requirement of wireless infrastructure deployment indoors (Raghavan et al. 2010). Similarly, UWB based systems require large number of receivers making it inconvenient and infeasible (due to space constraints) indoors (Montañés et al. 2013). These aforementioned techniques are challenging especially for existing older buildings where building engineers or managers might not be willing to deploy infrastructure just for the data collection process because of additional time, cost and budgetary constraints involved.

Other sensor modalities explored in literature which are widely adopted are cameras and laser scanners. Laser scanner and natural marker (camera) based techniques eliminate the need to instrument the physical space but they are highly expensive, sensitive to obstructions/lighting conditions and require high computational capabilities (Feng and Kamat 2012; Habib 2007; Thrun et al. 2005; Bar-Shalom et al. 2004; Burgard et al. 1998). Dynamic occupancy movement in the buildings and higher costs might hinder the wide spread implementation of these methods in one time data collection applications (e.g. collecting data in old buildings for making retrofit decisions). Although 2D bar codes (camera based) offer a cheaper solution, they fail to provide 3D orientation information (Xu and McCloskey, 2011). This is a matter of concern in autonomous robotic data collection because relative pose information is very essential for determining navigational instructions of the robot. To summarize the techniques reviewed, common disadvantages affecting a majority of the reviewed methods include low accuracy, significant upfront investments, high computational requirements and complex instrumentation of the indoor environment. One of the vision-based methods namely Fiducial Markers, however, is particularly immune to the aforementioned disadvantages afflicting other methods.

Fiducial markers (Olson, 2011) are immune to most of the limitations of the previous approaches and offer high accuracy in determining and estimating their relative 3D pose in an environment (thus can provide precise navigational instructions to the robot), require relatively less computing capabilities (can be done with the help of a robot and an onboard laptop), are cost-effective (they can be printed on paper) and are easy to install (Iwasaki and Fujinami, 2012).

However, a one-time strategic deployment of markers needs to be done in the monitored space. It is also assumed that sufficient lighting conditions prevail during the course of the navigation process for successful detection of the markers.

In this section, the general computing framework that makes use of fiducial markers to link between actual physical locations and virtual information stored regarding those locations for indoor robot localization is described. Fiducial markers have the capability to store virtual information regarding a multitude of things such as information regarding physical location (floor and room level information), emergency evacuation directions, indoor navigational instructions, and inspection related data regarding building systems helpful for facility managers (Feng and Kamat 2012). Feng and Kamat (2012) demonstrated how markers having virtual information and navigational instructions can help humans navigate indoors. Lee et al. (2013) utilized the same virtual information to achieve autonomous navigation abilities for an Autonomous Guided Vehicle (AGV). Expanding on these ideas, this study utilizes the virtual location information (for localization), navigational instruction (for navigation), data collection instruction (to collect and geotag ambient data collected by the robot), and 3D pose estimates (for drift correction) to achieve autonomous behavior of a mobile robot. These methods are discussed in detail in the following sections of the paper.

3.5.1.1 Initial Setup

First, all the required software packages such as Robotic Operation System (ROS), ROS developer kit, TurtleBot software, and network connectivity need to be installed on the mobile robot's netbook and remote laptop or wireless controller. Second, a bidirectional communication has to be established between netbook and the remote laptop and then the netbook is placed on the iCreate base. Finally, the remote laptop is connected to the mobile robot's netbook with the help of Secure Socket Shell (SSH) connection in the terminal. More information regarding detailed installation, software packages, robot hardware, and sensor driver instructions can be found in Quigley et al. (2009).

3.5.1.2 Technical Approach

For this study, unique fiducial markers are required to be placed at strategic locations along the navigational path of the robot (e.g., corridors, entrances to rooms, etc.) as shown in Figure 3.3 to form a Marker Network Map (MNM). These strategic locations include examples such as: end of the corridors, the intersection of hallways, assists that transfer of public from one floor to another (e.g. staircase and elevator), and entrances to the rooms. The MNM can then be used to generate a graphical node network G = {N, E} where N includes nodes representing locations of markers and E represents edge links connecting these nodes (e.g. corridors and stairs). Each edge can have edge weight attributes that can represent quantities such as distance or time. The network formed henceforth can be used to determine the optimal paths in the building or reroute in case of maintenance or emergency situations.

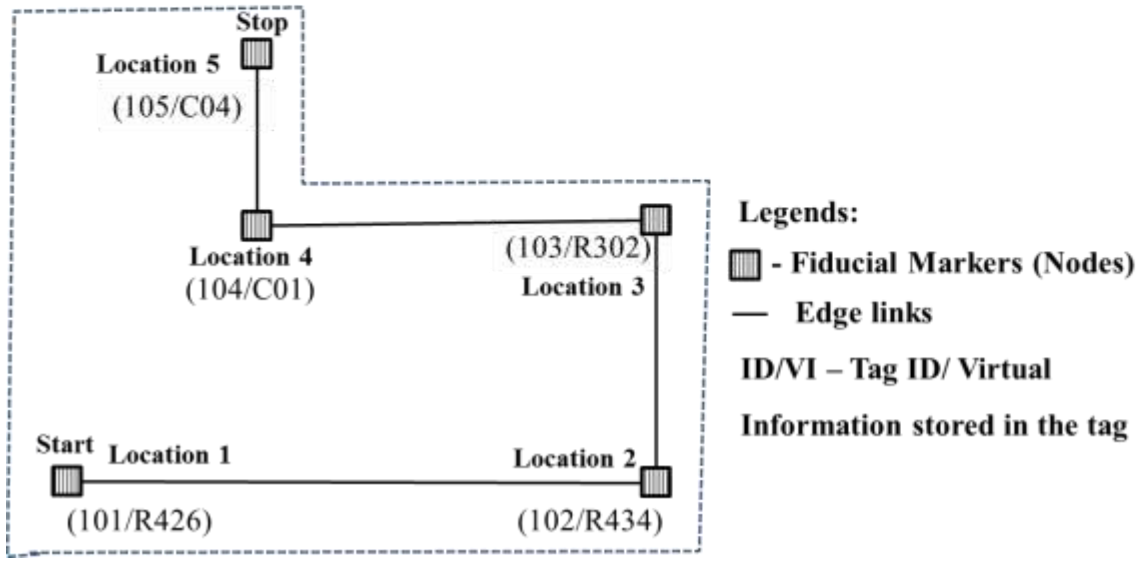


Figure 3.3 Marker Network Map (MNM) with virtual information regarding the location stored in each of the markers.

Multiple markers are generated by varying the positions of black and white squares (1's and 0's as detected by the computer algorithm) on the April Tag, and subsequently generating different unique combinations as shown in Figure 3.4. More information on characteristics and properties of the markers is available in Olson, (2011). These markers, which are printed on regular paper, are used to store the physical location information (e.g., Room 101) and navigational instructions (e.g., take right) which is necessary to help the robot determine its current location and its targeted headed direction.

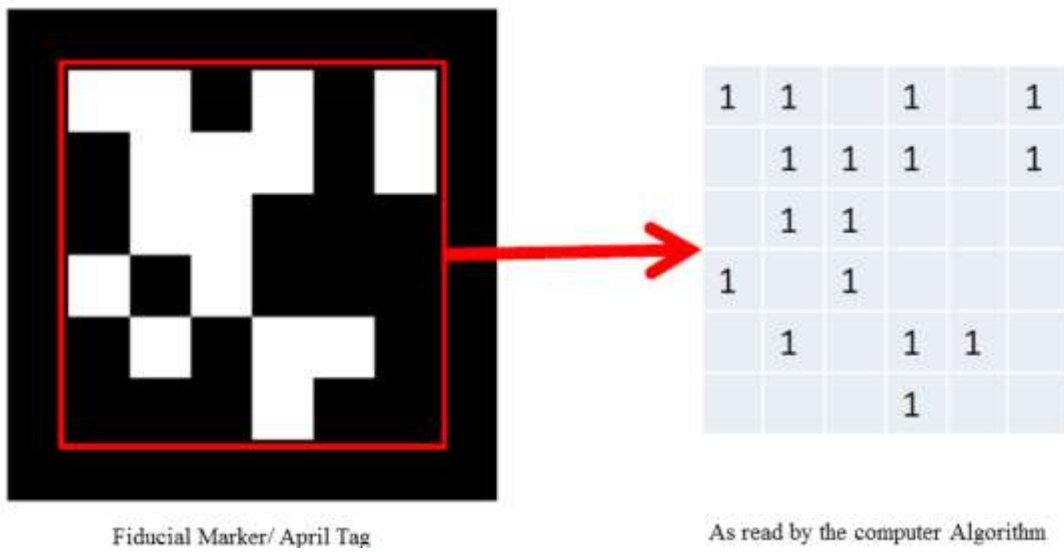


Figure 3.4 Fiducial marker as seen by a human eye and as detected by a computer algorithm

First, fiducial markers are detected with the help of the onboard RGB camera; images are captured at a very high rate and are analyzed for the presence of the marker. This is called segmenting as shown in Figure 3.5. Second, the computer decodes the information from the markers in the form of 1's and 0's and determines the ID of the marker by cross referencing (matching) with the database of markers. Then, virtual information stored in the marker along with relative pose information is extracted and interpreted (Olson 2011; Feng and Kamat, 2012). The logic of the localization algorithm is shown in Figure 3.5.

3.5.1.3 Marker Placement

Different marker placement techniques explored for localization/navigation in industrial/manufacturing robots are wall mounted, ceiling mounted, and floor mounted (Shneier and Bostelman 2015; Röhrig et al. 2012). Floor mounted markers have shown promising results in a structured industrial/warehouse setting where kiva robots manage the entire storage area of the warehouse (Röhrig et al. 2012). Horan et al. (2011) used a tape based path sensing method for mobile indoor robot navigation. A similar study was performed by NIST (Shneier and Bostelman 2015) where additional boundary markers were introduced along with the tape line. However, the aforementioned floor based marker mounting techniques would suffer from frequent wear and tear in an unstructured building environment with frequent occupant movement.

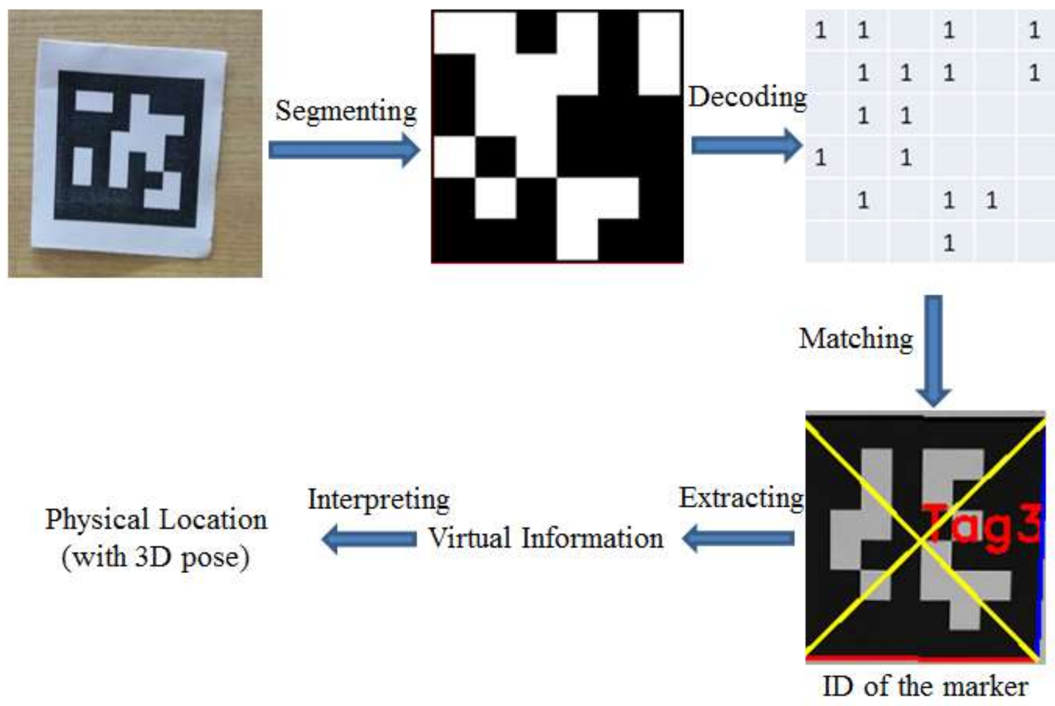


Figure 3.5 Localization algorithm

Ceiling mounted marker based techniques were explored in warehouses as an alternative to the laser triangulation method (Shneier and Bostelman 2015). However, the position accuracy of the mobile robot is a direct function of height of the ceiling. In the context of the current proposed methodology, the ceiling heights (especially near the atrium areas) might have a significant effect on the prediction of robots current state and drift correction. In addition, the natural/artificial illumination in the ceiling might affect the vision (camera) based marker recognition. Range based wall following techniques are generally used in indoor confined spaces

(Shneier and Bostelman 2015). A similar technique where wall mounted marker based localization/navigation is explored in this study.

Additionally, three of the important factors that are interrelated which need attention are the density of the makers (that needs to be deployed in the environment), the distance between the wall and the robot's ideal path, and the drift accumulation. For example, if the robot's ideal path is closer to the wall, the allowable drift accumulation will be very less (compared to the robot's path being on the centreline of the corridor) to avoid the robot colliding with the wall. This requires a more frequent drift correction. This will result in the need for higher density of markers (i.e., additional markers) to help in estimating the drift. The number of markers that needs to be instrumented in the building increases with the density of the marker network. To optimize the density of the marker network, the allowable drift should be maximized. For indoor building conditions, allowable drift can be maximized if the robots ideal path follows the centerline of the corridor (discussed in more detail in the drift correction sections of this paper).

Hence, a combination of wall and ceiling mounted markers would yield better results for complex indoor environments considering the advantages and limitations of each method. Further detailed discussion can be found in the navigation section of this paper. While in our experiments, wall mounted markers and middle robot travel (to optimize the density of markers) were considered, these factors can be easily adjusted. For example, for making the robot move along the side of the corridor, new marker to marker distance and maximum drift needs to be estimated (as discussed in navigation section of the paper). Accordingly, the markers need to be instrumented in the building and the respective input parameters to the algorithm need to be changed. However, no alterations or modifications need to be made in the logic of the algorithm.

3.5.2 Navigation

The robot's navigation can be briefly defined as the robot's ability to plan a course of action to reach the destination location while accurately localizing itself in its frame of reference at strategic locations (Levitt and Lawton, 1990). Two different kinds of navigation techniques namely predefined path mode, and dynamically configurable path mode are developed by the authors as part of this study and discussed in detail in this section. Predefined path mode navigation is mostly suited for places with regular occupancy and almost fixed spaces which require periodic data collection (e.g., every 15 minutes or 30 minutes) such as offices, data centers, and ware houses. Dynamically configurable path mode is highly applicable for flexible/ dynamically changing occupancy buildings where data collection locations change dynamically depending on occupancy all through the day such as retail stores, schools, colleges, shopping malls, and airports. The navigation logic for each of these three types is described in detail in the following paragraphs.

3.5.2.1 Predefined path mode

In this mode, the mobile robot is given a predefined path with a start location, end location, the path it needs to autonomously traverse to reach the end location, and also the data collection locations along the way. Several markers whose global positions and orientations are known in advance are placed at regular intervals along the navigational path as shown in Figure 3.3. For example, consider the situation where the user determines the path of the mobile robot to be from location 1 to location 5 through locations 2, 3, and 4. The information inputted by the user would be a list consisting of fiducial marker IDs, navigational instructions, and if the current location is a data collection location in sequential order (i.e. {[101,S,Y], [102,S,Y], [103,S,N], [104,S,Y], [105,L,N], [106,S,Y], [107,L,N], [108,S,N], [109,E,Y]} where S – go Straight, L – turn Left, R – turn Right, E – End location, Y – Yes, N - No).

The overview of the navigational algorithm logic is represented as a flowchart in Figure 3.6. First, the onboard traditional RGB camera continuously captures images that might potentially contain a known fiducial marker. The images are processed by the marker recognition module (an algorithm which detects the presence of the marker as previously discussed in the technical approach section of the paper). If a known marker is detected by the marker recognition module, the ID and relative pose of the robot with respect to the fiducial marker (in the camera's reference frame) is outputted by the module. Each ID is associated with a physical location in the indoor environment as shown in Figure 3.3. Current location of the robot based on the aforementioned information is estimated and pose correction is calculated based on the drift correction algorithms discussed below. The current approach of navigation can be termed as treasure hunt based navigation because the robot traverses the path from one marker to another marker with the help of clues provided at each marker. This means that the robot will follow the previously known navigational instruction (or clue provided by the last seen marker) until it finds a new marker. To avoid the cases of obstruction (robot not being able to see one or two markers in the navigation path), it can be easily programmed to store information regarding next couple of markers and still traverse the navigational path accordingly. None of the previous studies utilize 3D relative pose information and virtual directional information from fiducial markers for indoor robot navigation and thus one of the contributions of this study.

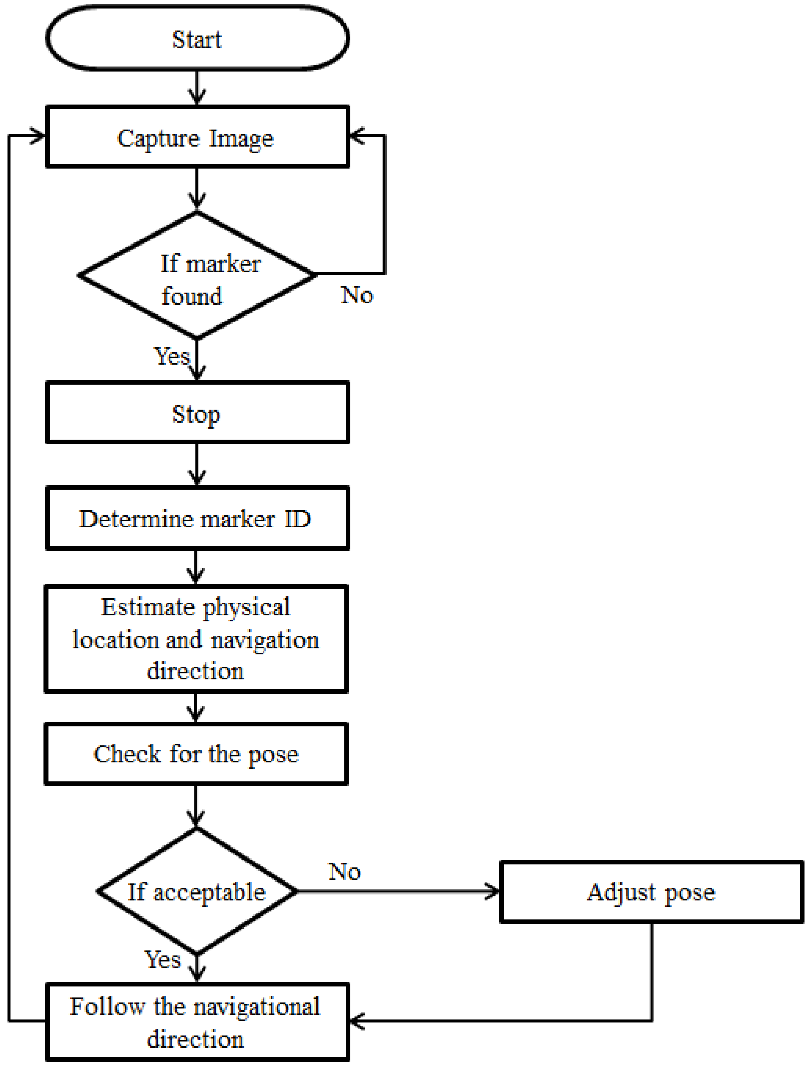


Figure 3.6 Autonomous indoor treasure hunt based navigation algorithm with the help of network of fiducial markers

Drift Correction (Predefined path mode): One of the most important factors that have a direct impact on whether the robot will traverse the entire path is drift accumulation of the robot. Drift is defined as the distance between the robots actual location and the robots ideal location at any given point of time. Negative drift (-ve drift) means that the robot has drifted inwards (towards the marker) and conversely positive drift (+ve drift) means that the robot has drifted outwards (away from the marker) as shown in Figure 3.7.

Since the drift is corrected at each marker location along the navigation path, the maximum drift in the entire path is guided by the marker to marker distance. Marker to marker distance is the distance between any two consecutively placed markers in the fiducial marker network. If the robot's ideal path is along the centreline in a building's corridor, then the maximum allowable drift in any direction (+ve or –ve) has to be less than half the width of the corridor as shown in Figure 3.8. This is to also account for the size of the robot itself. For example, the robot might collide with the wall before reaching the next marker if it drifts an amount of w/2 on either side of the robots ideal path. Thus, a confined path which is safe for the robot navigation (i.e., it would not collide with the walls or lose track of the markers during its path) has to be determined based on the width of the corridor mainly because of the possible errors in estimation of robots current pose, accumulated drift, and turning angle. A general consideration can be a robot confined path of w/2 with w/4 on either sides of the robots confined path as shown in Figure 3.9.

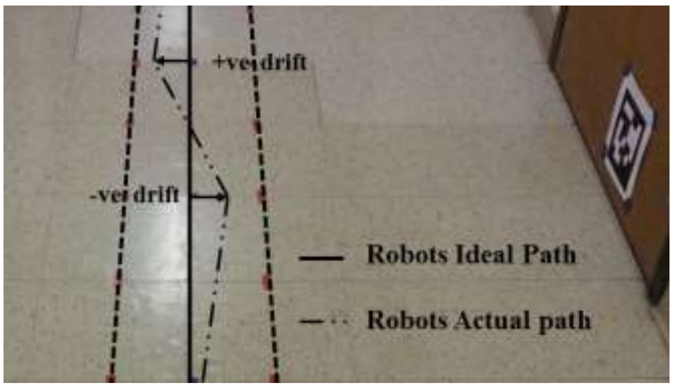


Figure 3.7 Negative (-ve) and positive (+ve) drift along the robot's path

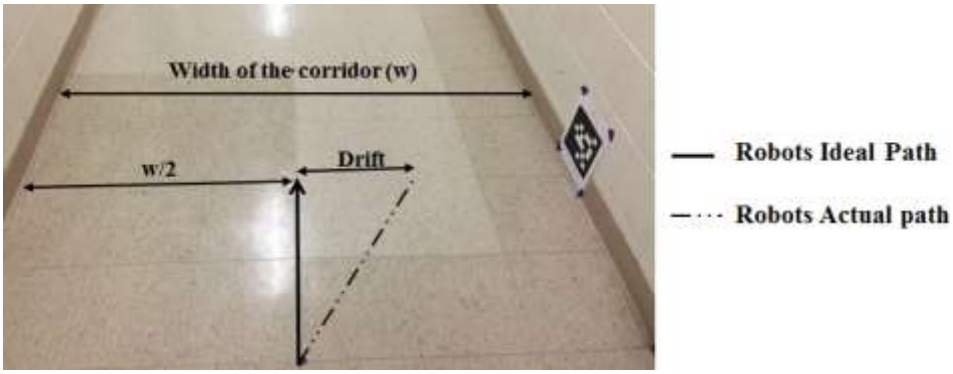


Figure 3.8 Maximum allowable drift of a robot in an indoor environment (corridor) considering the ideal robot's path to be the center line of the corridor.

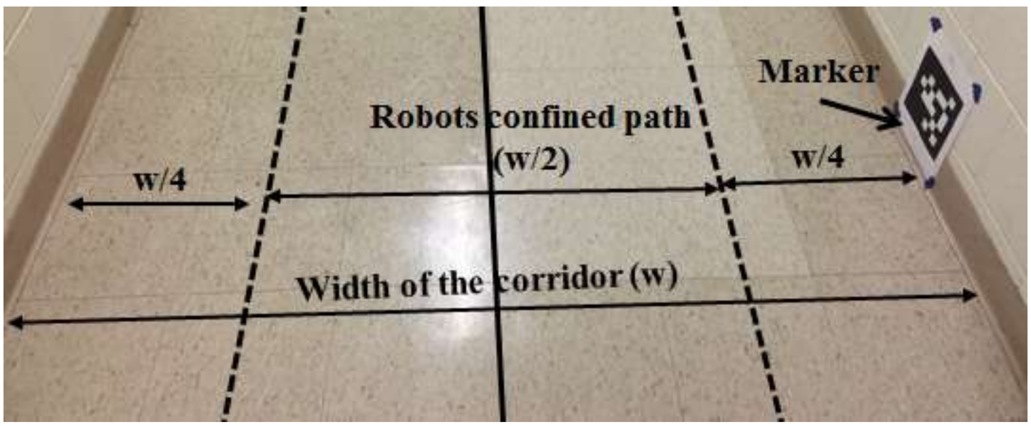


Figure 3.9 General consideration of robots confined path in an indoor environment (corridor) considering the ideal robot's path to be the center line of the corridor (dark black line in the figure).

As discussed, the robots drift needs to be corrected at every marker location to successfully reach the targeted destination location. The current drift correction algorithm developed is based on the relative pose information estimated with the help of the onboard camera, computer, the known marker's pose, and the known location in the fiducial markers network as per the algorithm discussed in the localization section. The algorithm returns 3D relative pose information (as shown in Eq. 1) in camera reference frame with respect to the marker. The entire matrix is referred to as homogenous transformation matrix in which R (3*3) denotes rotation matrix and T (3*1) denotes translation matrix. From that, the lateral distance information (distance between the camera and marker), d as shown in Figure 3.10, is extracted and the drift at each location (δdi) is estimated. After that, the pose correction (turning angle (αi) in this case) is calculated as shown in Eq. 2 and 3. A detailed description of various parameters in these equations is shown in Figure 3.10. Since the drift at each location might be different, the turning angle at each location also differs. Although Feng and Kamat (2012) used the relative pose estimate to determine navigational direction, estimating and correcting drift has not been addressed in prior related work.

$$\boldsymbol{H} = \begin{matrix} \boldsymbol{R11} & \boldsymbol{R12} & \boldsymbol{R13} & \boldsymbol{Tx} \\ \boldsymbol{R21} & \boldsymbol{R22} & \boldsymbol{R23} & \boldsymbol{Ty} \\ \boldsymbol{R31} & \boldsymbol{R32} & \boldsymbol{R33} & \boldsymbol{Tz} \end{matrix} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..\boldsymbol{Eq}(\boldsymbol{1})$$

Where:

H is the part of Homogeneous transform matrix returned by the localization algorithm

R (3*3): Rotation matrix

T (3*1): Translation matrix

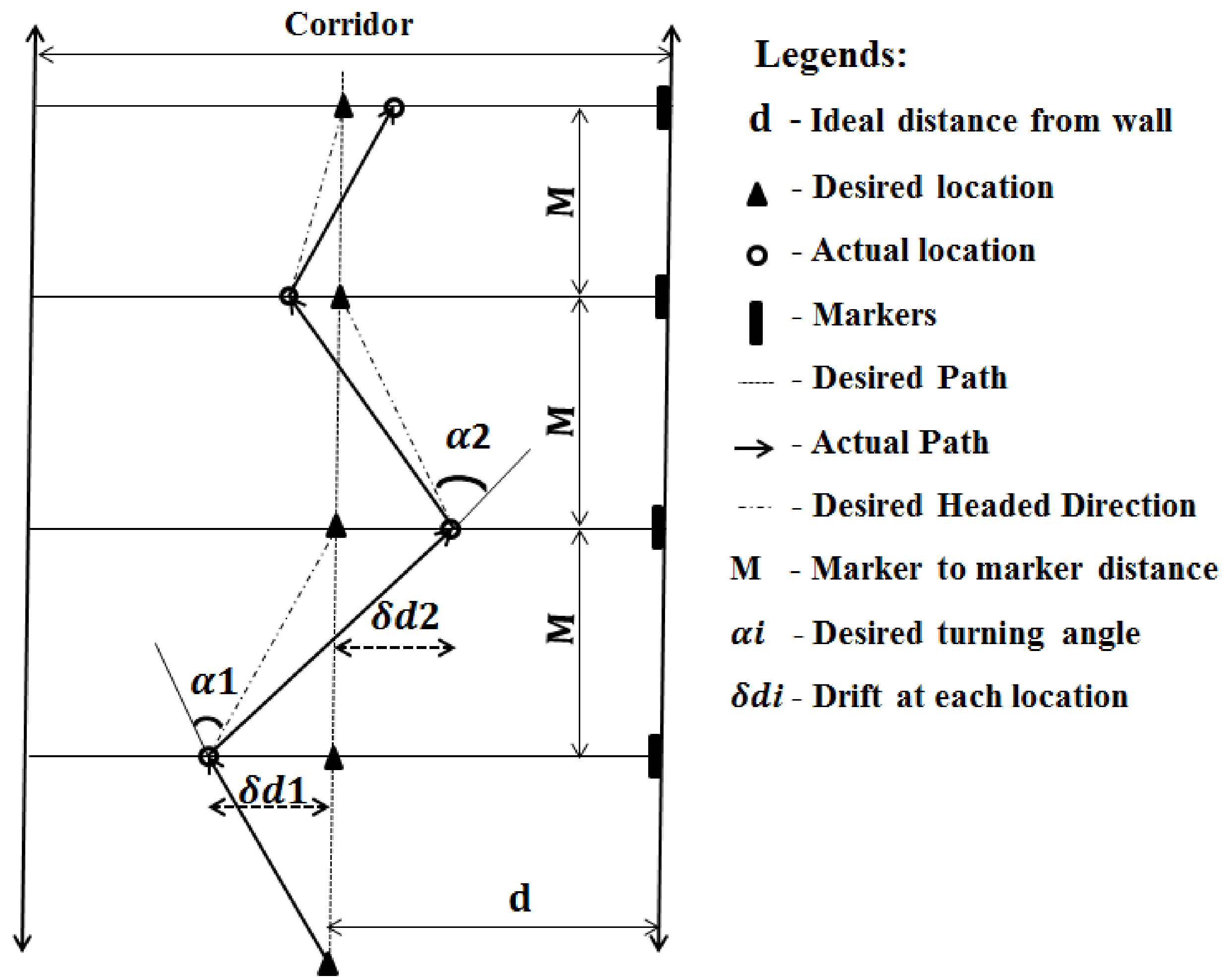


Figure 3.10 Detailed description of the development of drift correction equations

$$\boldsymbol{\alpha_1 = 2 * \tan^{-1}\left(\frac{\delta d}{M}\right)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \boldsymbol{Eq(2)}$$

$$\boldsymbol{\alpha_i = 180 - \tan^{-1}\left(\frac{\frac{M}{\delta d_i}}{1 + \left(\frac{\delta d_{(i-1)}}{\delta d_i}\right)}\right) - \tan^{-1}\left(\frac{M}{\delta d_i}\right) \ \forall\, i \geq 2} \ldots\ldots\ldots\ldots\ldots\ldots \boldsymbol{Eq(3)}$$

3.5.2.2. Dynamically configurable path mode

In this mode, the mobile robot's autonomous indoor navigation is based on user inputted coordinates which also allows the user to configure the robot's path dynamically. This mostly resembles and matches the criteria for real-world applications where the robot might have to traverse different paths during different times of the day.

The logic of the entire algorithm is the same as previously discussed and shown in Figure 3.6. However, the drift correction technique differs in this algorithm when compared to the previously discussed (Autonomous navigation (predefined path)) algorithm. The navigational instruction is estimated based on the user inputted coordinates by generating vectors based on adjacent pair of coordinates and calculating the angle between the pair of vectors. For example, if the list of user inputted coordinates (in global coordinate reference frame with origin being robots initial starting location) is L = { (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5)….., (x_n, y_n)}.,then the respective points represent P_1, P_2, P_3, P_4, P_5,…….. , Pn, the initial two vectors (V_1 and V_2) are generated with the help of Eq. 4 and Eq. 5 using P_1 P_2, and P_2 P_3 respectively.

Where

$$\overrightarrow{\boldsymbol{V_1}} = \begin{bmatrix} \boldsymbol{x_2 - x_1} \\ \boldsymbol{y_2 - y_1} \end{bmatrix} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \boldsymbol{Eq}(\boldsymbol{4})$$

$$\overrightarrow{\boldsymbol{V_2}} = \begin{bmatrix} \boldsymbol{x_3 - x_2} \\ \boldsymbol{y_3 - y_2} \end{bmatrix} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \boldsymbol{Eq}(\boldsymbol{5})$$

The directional angle (positive for anti-clockwise and negative for clockwise) between the vectors can be calculated with Eq. 6.

$$\boldsymbol{turning\ angle} = \boldsymbol{cos - 1}\left(\frac{\overrightarrow{\boldsymbol{V_1}} \cdot \overrightarrow{\boldsymbol{V_2}}}{\left\|\overrightarrow{\boldsymbol{V_1}}\right\| \ \left\|\overrightarrow{\boldsymbol{V_2}}\right\|}\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \boldsymbol{Eq}(\boldsymbol{6})$$

Where $\left\|\overrightarrow{\boldsymbol{V_1}}\right\|$ and $\left\|\overrightarrow{\boldsymbol{V_2}}\right\|$ denote the magnitude of the vector

The robot continues its last known navigational instruction until the marker at P_3 is recognized. The same methodology is applied to new set of points (P_2, P_3, and P_4), and the process continues until all the points are reached or all the markers are recognized. For example, if the user inputs the list of following coordinates { (0,0), (1,0), (1,1), (2,1), (2,2), (3,2), (3,3), (2,3), (1,2), (0,1) (0,0) }. Initially the robot goes straight until it sees the 1st marker placed at (1,0), then after the angle between the vectors (0,0) (1,0) and (1,0) (1,1) is calculated to be 90^0. The robot rotates 90^0 in anti-clockwise direction and the rest of the navigation logic continues in the same way. Also, when the robot reaches (2,3) an angle of 0^0 between (2,3) (1,3) and (2,3) (3,3) is calculated with the help of built-in algorithm and goes straight without turning. The intended path based on the user inputted coordinates for this example is shown in **Figure** 3.11.

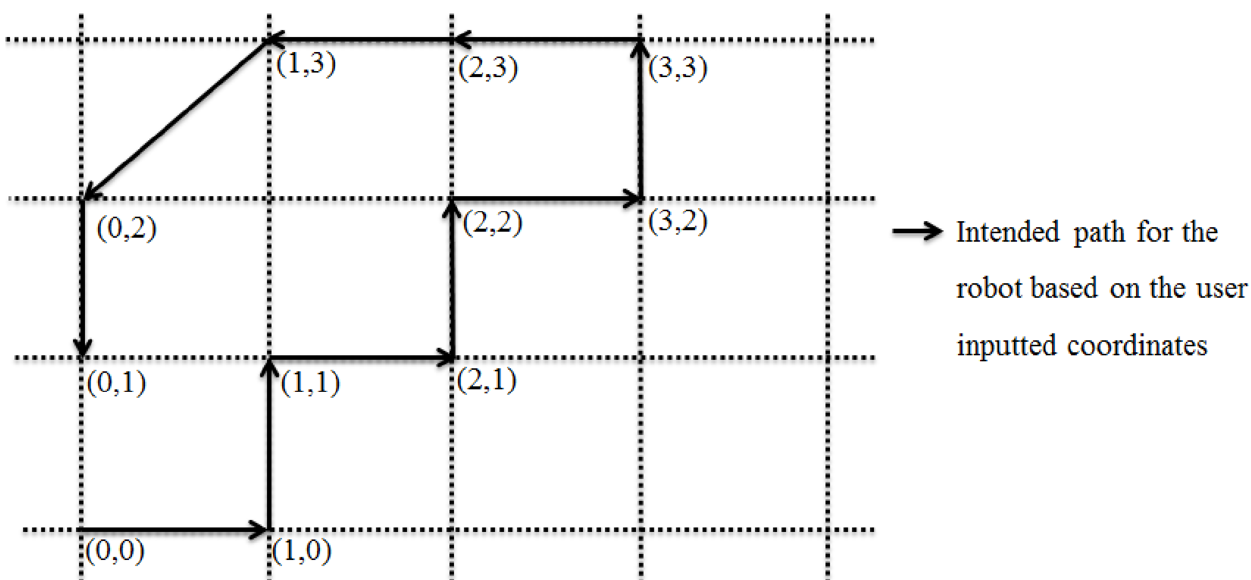


Figure 3.11 The path robot will traverse autonomously based on the user inputted coordinates.

Drift Correction (Dynamically configurable path): Similar to the drift correction algorithm discussed in the previous section, the current drift correction algorithm also works based on the relative pose information estimated with the help of the onboard camera, computer, the known marker's pose, and the known location in the fiducial markers network as per the algorithm discussed in the localization section. However, instead of calculating the lateral distance between the camera and the fiducial marker detected, this system estimates the relative pose angle between the plane of the marker and the plane of camera in cameras reference frame. The angle determined is termed as adjusted angle and it is added to the calculated angle in the navigation logic discussed above. Thus, a net corrected angle (based on the calculated angle and adjusted angle) is calculated and the robot instead of turning the entire calculated angle, rotates only the amount of corrected angle thereby, correcting its drift. The same process is continued at every marker location until the destination/ targeted location is reached.

3.5.2.3. Collision Prevention

Collision prevention is of significant importance due to the possibility of human obstacles and interferences in indoor buildings (Dieter et al. 1998). The primary goal is to prevent collision with any of the dynamic and static obstacles such as occupants and equipment. Thus, a collision prevention system is implemented on the mobile robot with the help of an onboard laser scanner. ROS has a package to integrate with the existing Turtlebot platform. The sensor outputs range values in every possible azimuth angle. At every time step, the minimum of the outputted range values are estimated to check for dynamic obstacles. If the value is less than a threshold (which can be arrived at based on the speed of the robot), the robot tries to go around (if possible) or stalls at its current position until the dynamic obstacle is cleared. However, if the obstruction is not cleared for an extended period of time or the robot is not able to continue the rest of the navigational path, then the robot reports the situation to the building/ facility manager for further assistance.

3.5.3 Data Collection and Geotagging

The immediate next step once the robot is cognizant of its surroundings and capable of navigating indoors is collecting data. Data collection is the process of gathering required information regarding various parameters of interest in a timely or systematic manner for testing a hypothesis or analyzing a research problem (NIU, 2016). The traditional way of collecting data is with the help of multiple sets of same sensors in the areas/ rooms/ locations of interest. Whereas, the proposed methodology uses single set of onboard sensors and a mobile platform to gather data at all the areas/ rooms/ locations of interest. In the context of this paper, parameters required to evaluate retrofit decisions such as temperature, humidity, indoor air quality, and light intensity are collected in buildings. The timely manner of data collection can be referred to as the frequency of data collection which is a function of the robot speed and required amount of data from a particular location. There are no restrictions on the amount of data that can be collected. However, as part of future work, the authors are researching efficient methods of collecting large data sets in buildings.

In the context of data collection, geotagging is the process of associating the information gathered (data) with the corresponding location such as latitude, longitude, specific name, or a specific location. In this study, geotagging is associating the ambient data collected along with its location information such as room number and floor number. To achieve this, a programmable interface is developed which bridges the communicated physical location information (as given by the fiducial markers) with the sensor data (obtained from the data collection). A python program is written which subscribes the published ROS data regarding the location of the robot (given by the fiducial marker as discussed in the localization section), concatenates it with the retrieved sensor data along with the time stamp, and exports the data to an excel file locally stored in the on-board netbook. This data then can be stored in a local server or can be updated to an online big data set interface for further real-time processing/ analysis or can be stored as data repository.

3.6 Verification

A case study was performed with a physical robot to validate all the four steps of methodology. Experiments were conducted in the Ross School of Business at the University of Michigan - Ann Arbor campus. The basement floor comprising of an open study lounge (monitored by four thermostats) and the open corridor (monitored by three thermostats) were chosen as the test bed for the case study experiments. Figure 3.12 shows the locations of the thermostats and/or the locations in the basement where the temperature readings are recorded by the BAS. Since the case study location chosen consists of public spaces (corridors and open student lounge) with dynamically changing occupancy levels, dynamically configurable path mode based navigation technique as discussed in the navigation section of the methodology is chosen for the robot data collection path.

3.6.1 Localization

The accuracy of determination of the relative pose of the robot with respect to the marker is critical for precisely estimating the drift correction. As already discussed, drift correction plays a key role in the robot successfully accomplishing the task and reaching the target destination. Thus, in this section, the performance results of the localization approach used for this study are tested and the corresponding results are shown in Figure 3.13. In order for the robot to precisely estimate its current location, it needs to be able to detect the fiducial marker first. Thus, considering various possibilities of indoor conditions, initial set of experiments were focused on the successful recognition of one of the family fiducial markers used for this study in various lighting conditions and camera angles (shown in Figure 3.13). The top three snapshots show the markers being detected at best and worst possible angles (with respect to the camera), and the bottom three show similar images in a poorly lighted indoor space. The two diagonals and the four edges drawn on fiducial markers (with different colors) show that the marker is detected successfully and “Tag 36” is the ID of the fiducial marker.

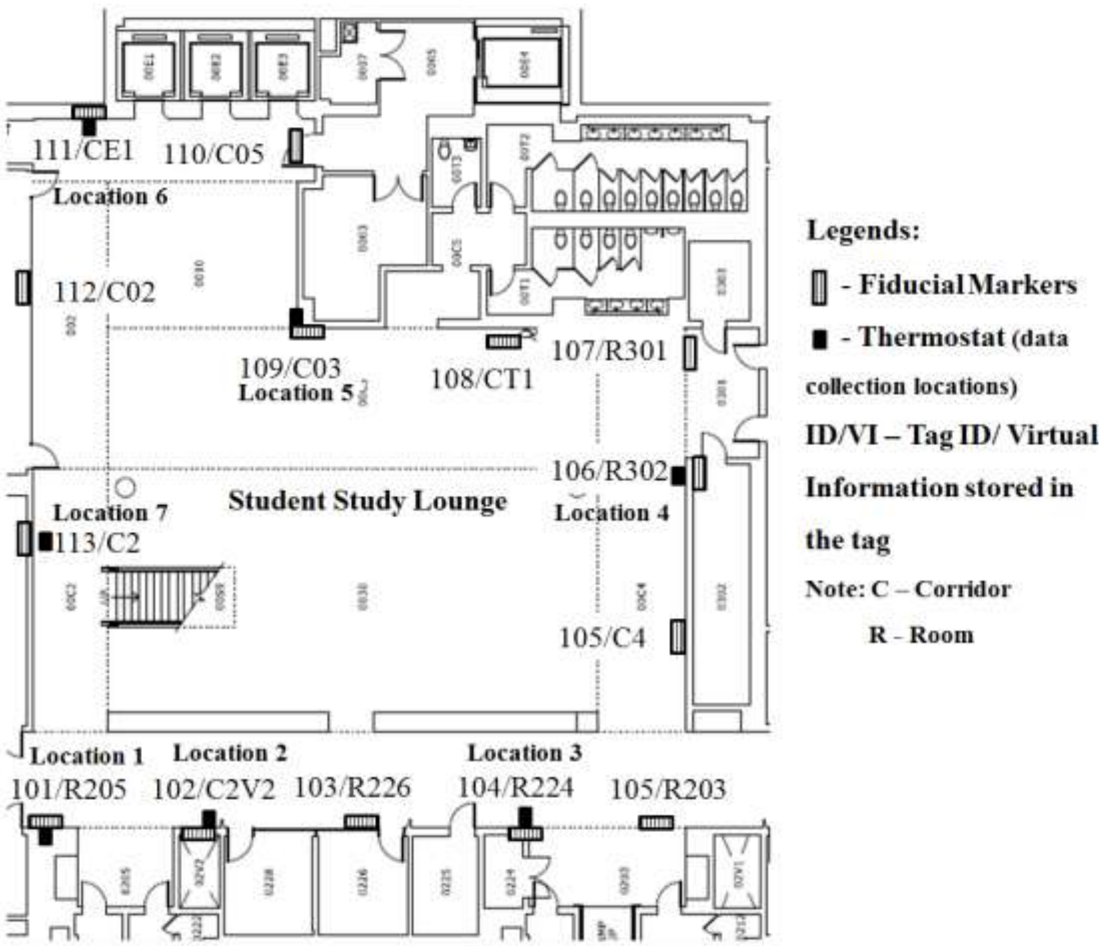


Figure 3.12 Fiducial marker network with the virtual information regarding the location stored in each of the markers along with the data collection locations in the basement floor of Ross School of Business, University of Michigan – Ann Arbor.

3.6.2 Navigation

A confined path which is safe for the robot navigation has to be determined based on the width of the corridor mainly because of the possible errors in estimation of robots current pose, accumulated drift, and turning angle. For this study, half the width of the corridor (w) was decided to be the robots confined path for the robots navigation in the corridor with equal distances on either side. Thus a maximum drift of w/4 can be estimated based on the center line of the robot and the robots confined path.

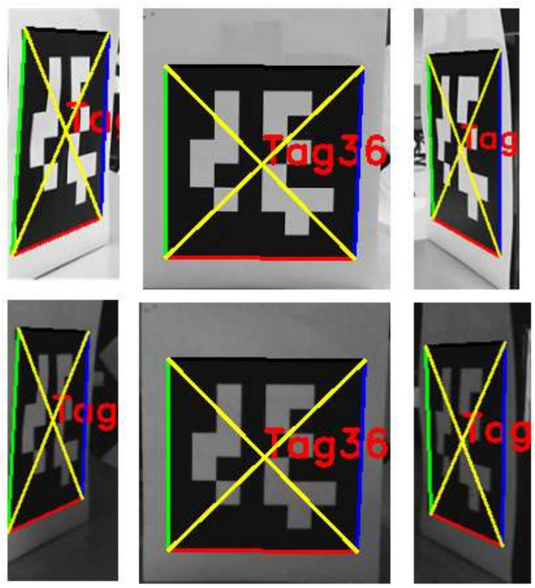


Figure 3.13 Successful recognition of one of the family of fiducial markers used for this study in various lighting conditions and camera angles.

First, the distance to distance between the markers needs to be estimated for setting up a fiducial marker network for robots navigation indoors. For that, several robot runs were performed to determine the maximum distance the robot will travel with a drift of 30cm (width of the corridor (180)/6). The results of the experiments are tabulated in Table 3.1. Each row in Table 3.1 represents a single trial of the experiment in the similar conditions. It is found that minimum of all distances travelled was 457 cm and thus marker to marker distance (M) was considered to be 400 cm for this particular case since there is a risk of robot colliding with the edges of the wall or losing sight of the marker. The general concept being the same, this distance might vary from each scenario (robot to robot and building to building) because the robot's drift may vary depending on several conditions such as type of the floor surface, friction between the tires and the floor surface, relative location of the camera on the bot and the marker on the wall, initial orientation of the robot, error in pose estimation, error in turning angle and robot's tendency to translate during the rotation commands. Now that the marker to marker distance is estimated, the immediate next step is to perform possible drift correction at every marker location along the navigation path.

Table 3.1 Experimental results to determine the marker to marker distance

SNO	Distance travelled (M in cm)	Drift (in cm)
1	627	30
2	647.5	30
3	911	30
4	1045.5	30
5	976	-30
6	854	30
7	549	30
8	554.5	30
9	1281	-30
10	518.5	30
11	732.5	30
12	1755	30
13	772.5	30
14	685	30
15	552	30
16	1111.5	-30
17	625	30
18	815	30
19	495.5	30
20	457	30

The validity of the drift correction algorithms discussed for autonomous mode (dynamically configurable path) is tested by conducting the following experiment. A four fiducial marker network was created along the corridor. The mobile robot autonomously navigates with the help of the aforementioned algorithm between the start and end locations. The experiment is repeated 10 times; each time the drift at every marker location was noted. The results shown in Table 3.2. indicate that the robot is always confined in the robots defined path and thereby concluding the robustness of the algorithm.

Table 3.2 Experimental results validating the drift correction algorithm developed.

SNO	**Drift at Marker 1 (in cm)**	**Drift at Marker 2 (in cm)**	**Drift at Marker 3 (in cm)**	**Drift at Marker 4 (in cm)**
1	-21	14	4	20
2	11	-26	1	-5
3	2	8	7	13
4	-13	16	2	14
5	-11	13	28	2
6	-3	2	20	23
7	-11	4	-14	15
8	13	-15	10	19
9	22	26	14	20
10	4	12	17	16

3.6.3 Data Collection

The data is collected in different locations (as shown in Figure 3.12) near the student study lounge, open corridors, and near elevators with the help of mobile robot in autonomous mode (dynamically configurable path). The readings were taken at all the locations with a time difference of 30 minutes similar to the pre-installed BAS samples. For example, the data is collected at location 1 at 10:00 AM, 10:30 AM, 11:00 AM at location 2 at 10:01 AM, 10:31 AM, 11:01 AM and similarly for the rest of the locations. CM-0199 COZIR® sensor along with the development kit was used for the ambient data collection. The accuracy of the sensor reading is +/- 1°C of the true value and the operating conditions of the sensor range from -25°C to 55°C (-13°F to 131°F). The sensor was calibrated every time before the start of the experiment. Zero point fresh air calibration was performed which means that the sensor was placed in fresh air environment for considerable amount of time, for the temperature to stabilize and for the fresh air to completely imbue into the sensor. Echo of a particular command is noted with the new zero point reading of the sensor. (COZIR, 2015) Data was collected at different hours of the day with varied occupancy levels from 06/15/2015 to 07/02/2015.

3.6.4 Geotagging

As discussed, geotagging means associating the location information with the data collected at that respective location. Thus, the data collected with the help of sensor needs to be associated with the current location of the robot. From left to right, information regarding the location id, date, time, humidity, temperature, and CO_2 values were recorded as shown in Figure 3.14.

location:0	Time:	6/18/2015	10:35:19	H	544	T	1211	Z	387	z	481
location:0	Time:	6/18/2015	11:03:51	H	548	T	1203	Z	360	z	360
location:0	Time:	6/18/2015	11:35:06	H	560	T	1209	Z	381	z	348
location:0	Time:	6/18/2015	12:05:45	H	574	T	1212	Z	391	z	422
location:0	Time:	6/18/2015	12:33:25	H	543	T	1215	Z	397	z	454
location:0	Time:	6/18/2015	13:00:47	H	530	T	1213	Z	377	z	328
location:0	Time:	6/18/2015	13:30:51	H	546	T	1213	Z	372	z	462
location:0	Time:	6/18/2015	14:02:52	H	573	T	1217	Z	395	z	312
location:0	Time:	6/18/2015	14:29:43	H	568	T	1217	Z	364	z	350
location:0	Time:	6/18/2015	14:58:23	H	537	T	1214	Z	401	z	414
location:0	Time:	6/18/2015	15:32:18	H	545	T	1215	Z	396	z	357
location:0	Time:	6/18/2015	16:00:04	H	583	T	1217	Z	346	z	334
location:0	Time:	6/18/2015	16:30:34	H	600	T	1213	Z	317	z	341
location:0	Time:	6/18/2015	17:01:03	H	566	T	1210	Z	336	z	358
location:0	Time:	6/18/2015	17:30:00	H	544	T	1210	Z	359	z	379
location:0	Time:	6/18/2015	17:59:23	H	565	T	1208	Z	368	z	418

Figure 3.14 A screenshot showing how the geotagged data is stored in an excel file in the local netbook.

3.7 Case Study

Once the respective ambient data is collected and geotagged in the building, the immediate next question that arises is how this data can be utilized. Energy simulation software packages such as EnergyPlus and eQuest are generally used to analyze the building energy performance and make informed retrofit decisions. There are three main phases of this decision making process 1) Model creation 2) Model Calibration 3) Simulation output analysis and decision making. In the first phase, an energy model of the existing building is created with the help of energy simulation software (e.g. EnergyPlus) and documents (e.g. floor plans and specifications). This is a representative energy model of the building at time = 0 years of the building (i.e. when the building was built). For the model to reflect on the current state (i.e. at time = t years) of the building, certain assumptions (e.g. current insulation properties of the materials) need to be incorporated because building systems deteriorate over time. In the second phase, the default energy simulation model parameters (e.g., set point temperature and occupancy schedule) are calibrated with the help of geotagged ambient data (e.g., air temperature data) collected with the help of the multi-sensor fused mobile robot. Finally, multiple scenarios (several combinations of retrofit options) are run and simulation output (e.g. total energy consumption) is compared to obtain the most optimal retrofit decision (such as inefficiencies in the building systems and degraded performance of building materials). A flowchart describing the entire process of retrofit decision making is shown in Figure 3.15 and each process is described in detail below.

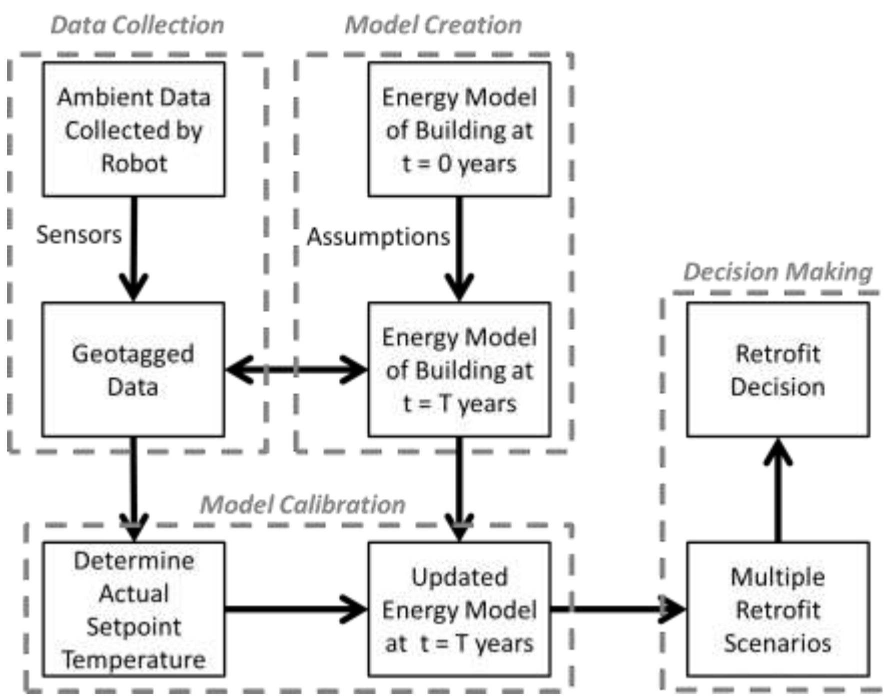


Figure 3.15 Building energy retrofits decision-making process.

A case study was performed to illustrate how the temperature data collected by the robotic platform can be utilized in the retrofit decision making process. Data collection was performed in an office room in the basement of G.G.Brown building at University of Michigan Ann Arbor. Ambient air temperature data was collected every 15 minutes for about 7 days (24 hours a day). The further process of model creation, model calibration, and simulation output analysis is described in detail in the following sections of this paper.

3.7.1 Model Creation

A base energy model is created with the help of EnergyPlus compatible models provided by Department of Energy (DOE) (US DOE 2015). The model chosen has similar basement dimensions, material properties and functions as that of the case study office room described previously where the data is collected. The model is an input data file (.idf file extension compatible with EnergyPlus) that can be edited as any normal text (.txt extension) file. Table 3.3. summarizes some of the major characteristics and simulation parameters of the chosen building model. The default model parameters (e.g. conductivity of the building materials) in the file do not take into account the building aging into consideration. However, it is not the case in reality since building systems and materials deteriorate with time and thus the performance gradually decreases.

Table 3.3 Characteristics of the case study building.

Description	**Inputs**
Location	Detroit, Michigan, USA
Latitude / Longitude	42.42^0/ -83.00^0
Shape	Rectangle
Building Length	73.11 m
Building Width	48.74 m
Type	Office building
No of floors	12 story plus basement
Area	46,320 m^2
Zones considered	Basement
Window – Wall Ratio	38% (equal distribution of windows)
Occupancy Density (Basement)	2.69/ 100 m^2
Simulation Time Period	September 28th – October 4^{th} 2016
Simulated Duration	168 hours

To account for this, in this case study, building material deterioration of the external envelope is considered and all the rest of the materials and systems are assumed to remain constant. That is, the effect of building aging on rest of the building systems and materials is not considered. The material performance (or insulating performance) is usually determined in terms of its R-value. R-value can be simply defined as the capacity of the insulating material to resist the heat flow (i.e. thermal resistance) (ORNL 2008; ASHRAE 2013). That is, R-value is proportional to the insulating power of the material. The lower the R-value, the lower the insulating capacity of the material and vice versa. EnergyPlus only has conductivity values of the building exterior materials and thus the conductivity has to be calculated with the help of R-value and the thickness of the material. The corresponding calculations of the R-values along with manufactures recommended service life for all the four building materials are shown in Table 3.4. It has to be noted that, for the purpose of this study, it is assumed that the R-value is same along the height of the wall.

A linear deterioration pattern is reasonably assumed for all the four external building envelope materials (Shohet et al. 2002). As proposed by Thomas et al. 2016, the building materials performance (R-value) is considered to reduce from 100% (at t = 0 years) to K% (at t = service life years). The value K signifies that the performance of the material degraded to such an extent that it has to be replaced. In case of linear deterioration, the material performance has a linear decrease from 100% to K% during the service life of the material as shown in Figure 3.16. The figure also shows a formula to calculate the R-value for any particular age (R_x) of the building given the initial R-value (R_0), K%, current age of the building (x), and service life (S). For the purpose of this case study, the value of K and the building age are assumed as 40% and 20 years respectively. The R-values estimated and the corresponding conductivity values are shown in Table 3.5. Though in this case, the numerical properties are numerically estimated, it is possible to estimate the R-Value of an assembly to determine current insulation properties (e.g. R-Value).

Table 3.4 R-value estimation for individual materials.

Item	**Conductivity (A)**	**Thickness (B)**	**U-Value (C) = (A)/(B)**	**R - Value (D) = 1/(C)**	**Material Age**
Units	W/m-K	m	W/m^2 -K	m^2 -K / W	in Years
1 inch Stucco - Layer 1	0.692	0.0253	27.34	0.037	30
8 inch Concrete - Layer 2	1.311	0.2032	6.45	0.155	50
Wall Insulation - Layer 3	0.049	0.0495	0.99	1.011	25
0.5 inch Gypsum - Layer 4	0.160	0.0127	12.60	0.079	25

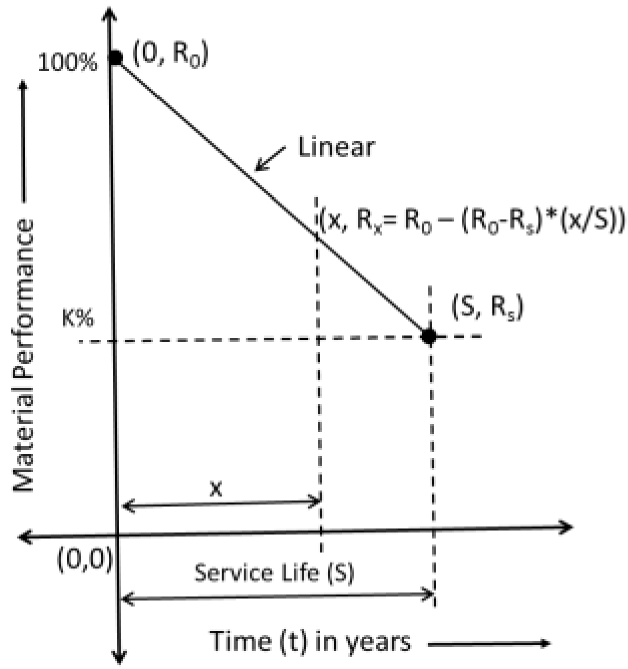


Figure 3.16 Linear deterioration pattern of the building material performance

Table 3.5 Conductivity value estimation for individual materials by assuming a building age of 20 years and K value as 40%.

Item	**Conductivity (A)**	**Thickness (B)**	**U-Value (C) = (A)/(B)**	**R - Value (D) = 1/(C)**	**Material Age**
Units	W/m-K	m	W/m^2 -K	m^2 -K / W	in Years
1 inch Stucco - Layer 1	1.1530	0.0253	45.57	0.022	30
8 inch Concrete - Layer 2	1.7250	0.2032	8.49	0.118	50
Wall Insulation - Layer 3	0.0942	0.0495	1.90	0.526	25
0.5 inch Gypsum -	0.3077	0.0127	24.23	0.041	25

Layer 4					

3.7.2 Model Calibration

This is an important step since the results and analysis rely on the efficiency of the calibrated model. The ambient air temperature data is collected every 15 minutes with the help of the robot. However, the ambient temperature cannot be directly used in EnergyPlus since ambient temperature is not a programmable input. EnergyPlus uses zone wise temperature set points to calculate energy demand and provide information about the resulting ambient temperature as an output. To address this issue, this approach calibrates the set point temperature (which is an input and can be changed in the energy model) in such a way that the corresponding output ambient temperature values match with the robot collected ambient temperature values.

A detailed step by step methodology of the proposed calibration procedure is as follows. First, run the model several times in EnergyPlus for a range of set point temperature values. Second, for each zone considered, generate and save the summary of output ambient air temperature values given by the EnergyPlus for the range of set point values determined in the previous step. Finally, determine the most appropriate set point temperature value by comparing the summary of robot collected ambient temperature data and output ambient temperatures (obtained by EnergyPlus simulations). The result of such a comparative analysis done on the current case study building is shown in Table 3.6. Based on visual inspection, a set point temperature of 24.1^{o}C is chosen as the most appropriate one (i.e. the best possible match with the robot collected data) since that is the closet match compared to others. To further strengthen the calibrated model, thermal cameras and other sensors can be used to verify the predicted R-values and/or thermal resistance as described by previous studies (Ahmad et al. 2014).

Table 3.6 Comparison of simulated (EnergyPlus) and actual (robot) ambient temperature values.

Ambient Temperature (in ^{o}C)	Set Point Temperatures (in ^{o}C)					Robot Collected Data
	23.0	24.0	24.1	24.2	24.3	NA
	Average number of hourly occurrences					
22.0 to 23.0	11.34	-	-	-	-	-
23.0 to 24.0	140.34	67.67	47	40.17	39.83	48.75
24.0 to 25.0	16.34	100.33	121	127.83	128	113.75
25.0 to 26.0	-	-	-	-	0.17	5.5
Total Simulated Hours	168	168	168	168	168	168

3.7.3 Simulation Output and Analysis

The aim of this step is to determine the best possible retrofit scenario to optimize the total energy consumption or annual energy demand of the building. Several combinations of retrofit scenarios are simulated and the total energy consumed (output) is generated. The option that gives the least energy consumption for the building is chosen as the final solution. The results for this case study are shown in Table 3.7 and as can be seen, third alternative (where layers 1, 2, and 3 are renovated) gives the most energy savings for both 7 days and 365 days simulation duration. The effective R-value of the assembly of materials is calculated based on series combination of materials as described in the sustainability workshop (Autodesk 2015). It can also be seen from Table 3.7 (layer 4 renovated with simulation duration of 7 days) that, renovating a particular material does not always guarantee energy savings. This might be possibly because of one or combination of reasons such as higher energy losses because of other building components (other than external wall assembly in this case) and varying external weather conditions

It has also been observed that renovating three of the layers gives more savings in energy compared to that of all the four layers. That is, a slight modification in R-values (20% to 45% improvement) has produced a 3% annual savings which might not be very significant. However, a similar analysis can be performed to study the effect of retrofitting other combination of building materials (e.g. doors, windows, and internal walls) using the same methodology to achieve more savings. In addition, ROI (Return Of Investment) analysis can also be performed to choose the most optimal solution. That is, for each retrofit alternative, the cost of retrofit along with the respective savings can be compared to determine the final solution instead of just comparing the most energy saving option. For instance, if the cost of renovating layer 4 is significantly higher, it is unreasonable to renovate only layer 4 because the energy savings is almost negligible (for simulation duration of 365 days).

Table 3.7 EnergyPlus output results for multiple retrofit scenarios

Total	**No Renovation**	**Layer 1 Renovated**	**Layers 1,2 Renovated**	**Layers 1,2,3 Renovated**	**Layers 1,2,3,4 Renovated**	**Layer 4 Renovated**	**Simulation Duration**
R-value (Assembly)	0.707	0.721	0.759	1.244	1.282	0.745	-

Total Site Energy (GJ)	504.45	504.36	503.42	498.88	501.97	505.66	7 days
% Savings	NA	0.02%	0.20%	**1.10%**	0.49%	-0.24%	
Total Site Energy (GJ)	30425.12	30378.43	30264.21	29512.08	29569.27	30424.31	365 days
% Savings	NA	0.153%	0.529%	**3.001%**	2.813%	0.003%	

3.8 Discussion, Conclusions And Limitations

The proposed methodology of mobile indoor robotic monitoring and data collection of ambient parameters discussed in this paper offer an effective and economical method as compared to the traditional state of the art (fixed stationary sensor network) data collection methods. This method involves using mobile indoor robots equipped with sensors to monitor and collect ambient data in buildings. It is particularly significant for old buildings that do not have an installed sensor network. With meager instrumentation of markers in buildings, the required data can be collected with the help of mobile robots. In addition, installing, calibrating, and removing sensor networks can be avoided if sensor fused autonomous robots are used in older buildings for collecting data and making retrofit decisions.

Furthermore, this study developed a generic framework that supports the data collected by a mobile robot to arrive at an optimal building retrofit decision (e.g., most economical and most energy saving). A case study was conducted with an objective to optimize the annual energy demand of the chosen space and determine the corresponding retrofit option. The results show the feasibility of this approach along with energy saving potential. The analysis illustrated how the model can be used to make informed retrofit decisions. The same approach adopted to investigate the effect of changing the external wall assembly can be used to propose several other retrofit options to the building to maximize energy saving potential from the retrofit

Some of the main characteristics of the traditional (stationary/fixed) sensor networks and mobile robotic data collection are described as follows. A) Upfront Costs: Fixed sensor networks need astronomical amount of sensors, while mobile based systems need only one set of sensors. B) Initial setup: Need to install and calibrate thousands of sensors in each room in a building. Though mobile based system requires installing markers in the environment, they are easily configurable and the calibration needs to be done only on one set of sensors. C) Operational and Maintenance Costs: For fixed systems, high manual and administrative costs are incurred for periodic battery replacement, maintenance, and calibration for each of these sensors. Locating these sensors may prove to be a challenge in complex buildings. On the other hand, it is comparatively a lot easier to perform the aforementioned tasks on one set of sensors. With regard to the runtime power consumption of the robot, it is envisioned that the robot will autonomously charge itself (similar to existing robotic platforms such as Roomba) when the battery is running low and resume the data collection. D) Frequency of data collection: Fixed systems are capable of collecting data at any frequency, but a single mobile platform is limited based on the area that needs to be monitored.

The proposed system requires instrumentation of markers in the environment. Careful consideration is required in this step for determining the placement of the markers, as they might suffer from occlusions. However, with minor alterations in the algorithm, it can be programmed to account for the occluded markers. These markers are very easy to deploy and configure as discussed in the technical approach section of the paper. Though collision prevention (doesn't collide with the dynamic or static obstacles such as occupants, walls, and objects) is considered currently, the robotic platform is not completely autonomous because of lack of obstacle avoidance (avoid obstacles to continue the navigation tasks) capabilities. However, with the help of onboard sensors and additional algorithms, obstacle avoidance capabilities can be achieved.

Regarding the privacy concerns, this method can be widely adopted in warehouses, data centers, museums, shopping centers and public spaces of academic and office facilities such as lounges and auditoriums. However, it has to be noted that in this study occupants did not find the robot disturbing during the case study data collection process. In addition, information regarding some of the data types such as appliance state and plug load (of a specific plug) cannot be collected without a fixed stationary sensor in each room because continuous monitoring is required for assessing the energy consumption and the appliance state throughout the day. Mobile robot will thus have some sensors (receiving nodes) that will receive information from the sensor fixed inside the room when the bot is nearby. The sensors fixed inside the rooms will be in a standby/ sleep state unless activated by the receiving nodes (in this case the ones placed on the mobile robot) (Tekdas et al. 2009).

Future research aims to incorporate robust obstacle avoidance and path re-planning methods in addition to the existing algorithms and achieve complete autonomous capabilities. Studies planned for future also include expanding the data collection to more data types and testing the robot in more complex indoor environments. In addition, compare and analyze the robotic and traditional sensor network data collection in terms of costs, sensor maintenance, and amount of data generated. There is also a strong need to further investigate the privacy issues with robotic data collection, especially in residential spaces.

Chapter 4

Usability Evaluation of a User Interface based on a Preferential Path Planning Algorithm for Assisting Individuals with Disabilities in Indoor Building Environments

4.1 Introduction

This paper presents a general framework for assisting individuals with disabilities (e.g. wheelchair and powerchair users) in indoor built environments with and without constraints considering user specific requirements (e.g. avoid stairs) and preferences (e.g. path with least number of turns instead of shortest distance). In addition, standard set of rules are established for generating an indoor graph network and marker map network which form a basis for automating the extraction of graph networks from floor plans. A visualization interface is developed with the help of proposed methodology and preliminary testing is conducted with individuals without disabilities in a pilot study. Case study results demonstrate the effectiveness of the proposed algorithm and provide insights to improve the design and functionality of the user interface.

Mobility is defined as the ability or quality of being able to move independently between locations of interest (Karimi et al. 2014). Several studies suggest that mobility impairment is one of the most prevalent concerns of the disabled population in the world (Kouroupetroglou 2013; Mirza et al., 2012; Sanchez et al., 2007; Tsetsos et al., 2006; Alm et al., 1998). In the past 30 years, the population of wheeled mobility users in United States (US) has grown 6 times (LaPlante, 2003). Several national surveys estimate that there are around 3.6 to 4.3 million mobility devices in the US (Boucher 2013, US Census Bureau 2010, Brault, 2008). This number is expected to grow from 5.2% to 8.8 % annually (Flagg, 2009). This is mainly attributed to the rapid increase in the US population's age and mobility related disabilities (King et al., 2013).

Wheelchairs are one of the most commonly used devices for mobility assistance. They enhance quality of life by allowing social inclusion and improving independence, self-esteem, and confidence (Oliveira 2016; Matthews et al., 2003). Even though there have been rapid technological advances in the last few decades, significant portion of the individuals with disabilities population still face several challenges to navigate in indoor environments (Kouroupetroglou 2013). As per the American Disability Act (ADA) 2010, 40% of the entrances might not even be accessible to the users of wheeled mobility devices. In addition, from any given location in the facility, there can only be a few accessible routes to egress (ADA 2010). This makes it not only challenging to find accessible route to desired locations of interest in unknown large complex buildings such as malls, airports, retail stores and hospitals but also time consuming (Tsetsos et al., 2005). Additionally, when there are multiple routes available, the desired route (e.g. path with least number of turns and path with the easiest trajectories) might not be readily known to the disabled individual.

Several assisted indoor navigation methods and systems for individuals with disabilities were suggested by previous researchers to address the aforementioned issues. Most of them are based on expensive and time consuming instrumentation (e.g. Bluetooth, Radio Frequency IDentification, and Wifi) of the indoor environment (Blattner et al., 2015; Chang et al., 2010; Garcia et al., 2010). These proposed methods are expensive due to the initial and operational costs of the beacons and time consuming because of the setting up, calibration, and regular maintenance requirements of the beacons. This is especially challenging in the buildings where such a system does not already exist. To that extent, vision based techniques that utilize fiducial markers (i.e. landmarks) which are cost effective, highly reconfigurable, and easy to install were explored (Oliveira et al., 2016; Neges et al., 2015). That is, visual cues (unique fiducial markers in this case) are placed at predefined locations along the navigational path in the indoor environment. These markers have the capability of storing virtual information (e.g. Room 1121) which helps determine the individual's location and thus the navigational direction. Further information regarding science, workability, and properties of the markers can be found in Mantha el al. 2016; Feng and Kamat 2012; Olson 2011.

The state of the art approach to determine the optimal path and guide individuals along the desired path using markers is by a) creating a graph-based network b) solving it using traditional algorithms such as Dijkstra (Dijkstra 1959; Dorigo et al., 2008) c) placing markers for turn by turn instructions. For e.g. Olivera et al., (2016) developed a marker based assisted navigation system dedicated to individuals with disabilities with the help of fiducial markers and a smart phone application. Similarly, Neges et al., (2015) utilized natural markers for assisting facility maintenance personnel navigate indoors. However, both the aforementioned methods did not articulate the procedure for creating the graph network, marker location selection (which locations in the buildings need markers and marker placement (where to place the markers in the specified location of the building). In addition, these approaches did not consider human preferred or human comfortable path planning but planned the shortest path considering distance. That is, planning a path which is most comfortable to the user taking the ease of navigation and preferences of the user in to consideration.

To summarize, none of the existing fiducial marker based methods explain the fundamental basis of the path planning and navigation process which is graph network and the corresponding marker network map creation. That is, the process of graph/marker network creation based on the indoor environment which involves defining a node, defining an edge, defining an edge attribute, defining a marker location, and its corresponding marker placement. To address these key research gaps, this study a) Established rules for creating and defining a node and an edge which form a basis for automating the indoor graph network creation b) Discussed and signified the attribute loaded (i.e. with edge weights other than distances) graph networks c) Developed a generic algorithm (which is not context specific) for determining the optimal path from the attributed loaded networks d) Established rules for marker location selection and thus the marker network map creation e) Presented a User Interface (UI) that demonstrates and integrates everything together f) Conducted a case study with the help of individuals without mobility impairments in a pilot study to show the feasibility of the proposed approach.

4.2 Background

Prior efforts have emphasized that individuals with disabilities are subjected to planning, significant cognitive burden and stress while navigating inside unfamiliar complex indoor spaces (Chen et al., 2011). In the context of disabled individual’s indoor navigation, planning refers to the ability of making higher-level decisions based on available information and cognition is the ability to continually process environmental information and take corresponding actions over time without external help. Several technological interventions (i.e. indoor navigation technologies) were suggested to reduce the planning and cognitive burden on the user.

The widely used outdoor navigation systems based on Global Positioning Systems (GPS) fail to work in indoor environments (Khoury and Kamat 2009) due to weak signal strength. That is, it fails to provide individuals with disabilities with a best possible route because of precision issues (Postolache et al., 2011; Deruwe and Wall, 2008). To address this issue, Bluetooth, Radio Frequency Identification (RFID), and Zigbee based navigation techniques were explored to assist persons with disabilities (Blattner et al., 2015; Garcia et al., 2010). Though these systems have good accuracy, they all require dense physical instrumentation (e.g. beacons) in the indoor environment. These not only have high initial costs but also require periodic maintenance and repair. On the other hand, laser scanners based navigation technologies eliminate the need of instrumentation of physical space (Montesano et al. 2010; Fernández-Madrigal et al. 2004). However, even laser scanners are expensive and require high computational capabilities. In addition, in the context of individuals with disabilities, a model of the building also needs to be created apriori by navigating all the corridors (Fernández-Madrigal et al. 2004). Such limitations prevent the adoption of such technologies for agile indoor navigation of individuals with disabilities.

To overcome the disadvantages faced by the other approaches, studies explored vision based navigation systems that use cameras and markers and provide turn by turn instructions (Oliveira et al., 2016; Kim et al., 2015; Gionata et al., 2014; Carlson and Demiris 2012). Markers are landmarks which help identify and localize the users in the indoor environment. Neges et al., (2015) created an indoor assisted navigation system using natural landmarks and a camera. However, some of the limitations of this approach are a) the start location has to be manually determined by the user; and b) the proposed method works only for navigation within a single floor and does not take into account the stair cases and elevators or need to navigate the wheelchair across different floors. These systems in general require prior training for determining the visual cues in the environment and thus are context specific and sensitive.

However, fiducial markers such as April tags developed by Olson (2011) are highly accurate, require relatively less computing capabilities, are cost-effective (they can be printed on paper) and are easy to install (Iwasaki and Fujinami, 2012). Though physical instrumentation of the space is required, it is cost effective and has less to no maintenance compared to other approaches where beacon/receiver installations are required. Despite these advantages, the current fiducial marker based systems still lack several features that are crucial to the disabled individual's navigation. For example, most of these approaches do not provide crucial indoor environment information such as access ramps, escalators, elevators, exits, and other significant locations (e.g. water fountains and restrooms).

Furthermore, none of the proposed solutions describe the general principles of marker network and graph network creation based on the indoor environment. This is crucial for successful assisted indoor navigation as the path computations are determined based on the networks formed. For instance, these approaches did not consider a situation where a marker is missed by the user. That is, if the user takes a wrong turn along the path, he/she will only be redirected at the next landmark (or marker in this case). This not only causes additional burden on the user but also has implications on the total journey time. In addition, markers are only placed at strategic locations making it difficult for the users to be cognizant of the immediate future event. That is, the users do not have any knowledge of their next action until a marker is scanned.

Proper navigation systems for individuals with disabilities must also take into account human preferences (Oliveira et al., 2016). However, most of the approaches considered only distance and determined the shortest path (Wu et. al., 2011; Teo and Cho 2016; Gecko 2017) with the help of standard algorithms such as Dijkstra's or A* (Dijkstra 1959; Dorigo et al., 2008) neglecting other factors such as accessibility and human preferences which are crucial for individuals with disabilities (Morales et al., 2015). For example, a preferred path can include a path with automatic caution doors along the way or a path which consists of least number of turns and not necessarily the shortest from the current location to the destination. In such cases, the path cannot be directly computed with the help of existing algorithms.

To address these key research gaps, a more flexible navigation system is required that is immune to the aforementioned limitations and has the capability of incorporating all the user preferences and constraints. Thus, the objectives of this study are to a) Describe in detail the attribute loaded graph and marker networks creation process b) Develop an algorithm to solve the attribute sensitive network based on the user preferences c) Investigate the functional requirements of a visualization interface and d) Perform a case study with the help of four scenarios to show the applicability of the proposed approach.

4.3 Methodology

The state of the art approach to determine the path in an indoor environment using fiducial markers is as follows a) create a graphical network based on the indoor environment b) use graphical traversal methods to determine the shortest distance between the current and the destination location c) place markers in the environment d) assist the individual to navigate along the determined path with the help of markers.

The methodology is divided into four major steps. First, the process and the significance of the attribute loaded graph networks based on the built environment are explained. Second, the network of markers required for the assisted navigation is established along with the taxonomy of markers. Third, the algorithmic process of the attribute sensitive path planning based on the user input preferences is described. Finally, functional requirements of the assisted navigation visualization interface are investigated. The following subsections describe these steps in the context of providing navigation instructions for individuals with disabilities.

4.3.1 Attribute Loaded Graph Network

In the current context, planning refers to estimating the most preferred way to navigate from the current location to the desired destination location. The most preferred way or path determined in this process is called the optimal route. Cognitive mapping refers to creation of a mental image of the building map (Fallah et al., 2013). Traditionally, the path planning exercise was performed by looking at the map/building directory of the facility, visually estimating the path that needs to be traversed, and perform cognitive mapping to complete the task (Fallah et al., 2013) or with the help of visual landmarks and signage (Villar et al., 2014). However, individuals with disabilities need additional information regarding features like staircases, possible risers, and hallways with automatic doors that physically limit their ability to traverse through these (Oliveira et al., 2016). Thus, even if they manage to locate themselves in the map, it may still be hard to orient them imposing significant cognitive burden particularly on the disabled (Villar et al. 2014).

The current state of the art practices to determine the optimal path for an individual in the building first create a mathematical representation (i.e. graph network) of the indoor built environment and then compute the necessary path with the help of graph network based algorithms. That is, first, the building plan or Computer Aided Design (CAD) model or Building Information Model (BIM) model is registered. Then, graphical networks are constructed with nodes representing distinct locations (e.g., rooms and elevators) and edges representing the node connections (e.g., corridors and passages). There can be various weight parameters (or attributes) for edges in the case of an indoor buildings such as distance, time, type (e.g. stairs and elevators), accessibility (e.g., based on physical dimensions of the corridor), and preferential (e.g. hallways with automatic doors) (Mantha et. al., 2017). The final graph network formed is the primary basis for the path planning algorithm computation. However, this step has received less attention and previous efforts did not establish guidelines to define and place a node, an edge, and the corresponding edge attributes. This step is important because of the following three main reasons.

First, improper placement of nodes can lead to incomplete network and this might lead to missing locations or paths in the building. Second, excessive placement of nodes can lead to network with redundant nodes which might lead to additional computational burden to estimate the optimal path. Third, networks with single edge attributes can cause additional burden on the individual. For example, consider a network with distance as an edge attribute. In this case, only shortest path between two nodes can be determined since there is no knowledge of other attributes such as possibility of stairs. This might lead to directing individuals along the paths with stairs causing additional burden, time, and effort on the user. Thus, to address these limitations, the proposed method clearly articulates the basic elements of the graph network formation (i.e. how to define nodes and edges) and proposes attribute loaded (i.e. multiple attributes for each edge) graph networks. In addition, it also categorizes the edge attributes of the network which form a key basis for the proposed planning algorithm described in the attribute sensitive path planning section of this paper.

A graph (G = (V, E)) is a combination of nodes (or vertices) (V) and edges (E) which denote the connection between the nodes. In the context of buildings, the authors comprehensively define the node with the following three basic rules based on the distinct characteristics of the floor plan layouts and node objects as described in Zhi et al., (2003): a) any notable location which might be of interest to a person. For example, any location of possible interest to a individuals with disabilities such as the entrance of any room located inside a building. But, a random location in a hallway where there is no room cannot be a node; b) any location with elbows (e.g. with only a left or right turn) and junctions (such as T, Y, 3-way, and 4-way); c) any location which assists the transfer of general public from one floor to another such as escalators, elevators, and stairs d) any location which signifies the end of the hallway or corridor. Figure 4.1(c) shows the locations of nodes (solid black circles) in floor plan. Each of these nodes follows one or more rules mentioned above. For example, nodes 3-6 follow the rule c and nodes 0-2 follow the rule b.

An edge in the current context is a connection that connects two locations in the buildings where a physical connection is possible. For example two nodes in the same hallway can be connected with an edge because it is possible to go from one to another with the help of the hallway. However, any random node in a particular floor cannot be connected with an edge to a node in another floor unless these two nodes represent a physical link such as elevators or stairs. An example floor plan along with the corresponding graph network is shown in Figure 4.1(b).

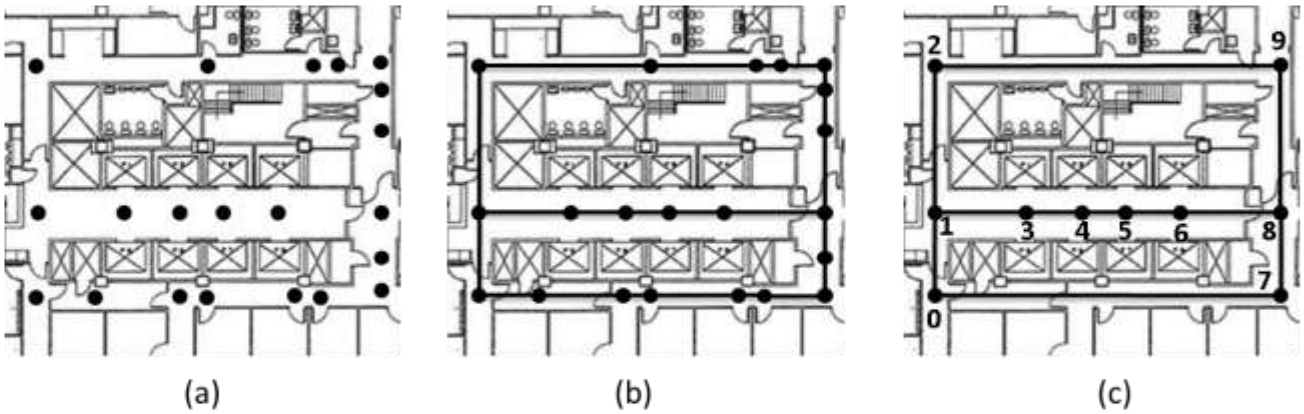


Figure 4.1 The process of optimized (reduced) node network generation (c) from graph network (b) and identified nodes (a).

With increase in number of nodes, the number of possible paths increase drastically and thus the computational complexity increases to determine the optimal path (Selim and Zhan 2016). Therefore, size of the network plays a significant role in the path planning algorithmic process. Hence, it is important to optimize the size of the network. This can be done by removing the redundant nodes in the network. Based on the definition of the nodes described previously, a redundant node is a node which follows only rule a in the definition of a node. For example, nodes 3-6 are not redundant nodes because they follow both the rules a and c. Similarly, the rest of the nodes 0-2 and 7-9 are not redundant nodes because they follow only rule b. The optimized node network after removing the redundant nodes is shown in Figure 4.1(c).

Each edge can have multiple attributes such as distance, time, and ease of accessibility. With the help of these multiple attributes, there can be multiple graph networks of the same plan of the building, each graph representing a different edge attribute. Mantha et al. 2017 describes the significance of multi-layered graphs or the possibility of multiple graph networks. Figure 4.2a and 2b show an example of a multilayered graph with distance and possibility of automatic caution doors as edge attributes respectively. Where ‘N’ means the corresponding edge contains a door which is not an automatic caution door.

The standard procedure to determine the shortest path using traditional graph traversal based algorithms such as Djkstra’s or A* as mentioned previously and distance as edge attribute shown in Figure 4.2a. However, in case of additional constraints or preferences such as a combination of Figure 4.2a and 2b, Oliveira et al., (2016) suggests updating the network first and then determine the shortest path. That is, in the case of Figure 4.2b, edges with an attribute ‘Y’ are removed and the network is updated first and then the shortest path is determined accordingly. However, this method fails if there are more than two edge attributes. Furthermore, Morales et al., (2016) developed a human comfortable path planner (HCoPP) for autonomous wheelchairs with three attributes namely length (distance), comfort, and visibility. Even this approach is not generic and does not work if other edge attributes are to be considered. Thus, this study proposes a generic framework to determine the optimal path for any multi-layered graph network which is more flexible and complete compared to the existing methods. To achieve this, authors categorize the types of attributes and subsequently explain its significance along with step by step procedure of the framework developed in the attribute sensitive path planning section.

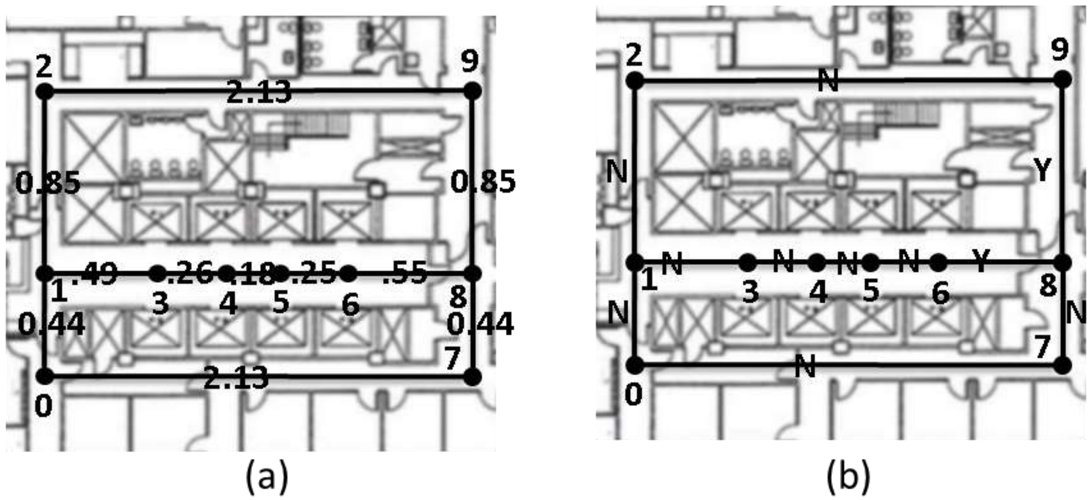


Figure 4.2 Graph networks based on same floor plan with edge attributes as distance (a) and edges with manual doors (b)

The attributes are broadly categorized into three types based on the quantitative and qualitative measures namely a) Measured b) Binary and c) Subjective. Any quantitative value that can be attributed for a particular edge comes under measure type. The most common example for a measured type attribute can be the distance which represents the physical distance between the nodes representing the edge. Another example can be time that requires traversing a particular edge. As the name suggests, binary is composed of or involves two possibilities Yes or No. For example, accessibility of an edge for a disabled individual is a binary type attribute. This is because an edge is either accessible (Yes) or not accessible (No) i.e. 1 or 0 in algorithmic terms respectively. Lastly, all those attributes such as ease of maneuvering, terrain difficulty, visibility, and others can be termed as subjective types. This is due to the fact that ease of maneuvering depends on the type of the wheelchair or powerchair and specific characteristics of a particular user. Similarly, terrain difficulty also varies between a powerchair user and a manually operated wheelchair user. In addition, it is also subjective to the physical abilities of a particular user. Depending on the user's capabilities or preferences, a score ranging from 1 to 5 might be assigned with 1 being the least and 5 being the most difficult.

4.3.2 Attribute Sensitive Path Planning

The goal of this step is to determine the optimal path for the user based on the preferential user inputs and/or constraints. First, an attributed loaded graph network is created based on the indoor built environment as explained in the previous section and the corresponding mathematical representation is described later in this section. Second, the network is updated based on the possible binary edge attributes. Third, all possible simple paths are determined with the help of a recursive algorithm and their corresponding costs are estimated. Finally, the optimal path is determined based on the computed costs.

Let a graph (G) consists of n nodes (or vertices V), m edges (E), and l attributes (A) based on any built environment. Then, the value of an edge attribute between the nodes $V_i$and $V_j$in $V = \{V_1, V_2, \dots, V_n\}$ can be represented as $A_{m_x}\{V_i, V_j\}$ or $A_{b_y}\{V_i, V_j\}$, or $A_{s_z}\{V_i, V_j\}$ where A_m is measurement type, $A_b$is binary type, $A_s$is subjective type, $i \epsilon [0, n], j \epsilon [0, m]$, $x, y, z \epsilon [0, l]$, and $x + y + z = l$. That is, x is number of measurement type attributes, y is number of number of binary type attributes, and z is number of subjective type attributes. For example, the edge attributes values for the edge between nodes 0 and 7 in the graph shown in Figures 2a and 2b will be {{2.13},{0},{}}. Here $x = 1, y = 1, z = 0$, and $l = 2$. That is, 2.13 represents the distance and '0' represents there is no manual door. At this stage, G represents the attribute loaded graph network.

4.3.2.1 Network Update

In this step, the network is updated based on the binary attribute type. That is, edge filter(s) need to be applied and all the corresponding edges in the network will be removed. For example, consider a situation where the disabled individual prefers paths that consist of automatic door openers. Figure 4.3a shows a graph network superimposed on the building's floor plan with edge attributes representing the possibility of a manual door opener. That is, a value of 1 means there is a manual door on that particular edge and 0 either means there is no door or there is an automatic caution door. In the figure shown, the doors present on the edges between nodes 6, 8 and 8, 9 are not automatic caution doors. The updated graph network based on this particular user preference will be as shown in Figure 4.3b. That is, after filtering out the edges between nodes 6, 8 and 8, 9. It has to be noted that after the network update step, there will be only two attribute types left namely, measurement and subjective attribute types since all the binary attribute types are utilized to update the network.

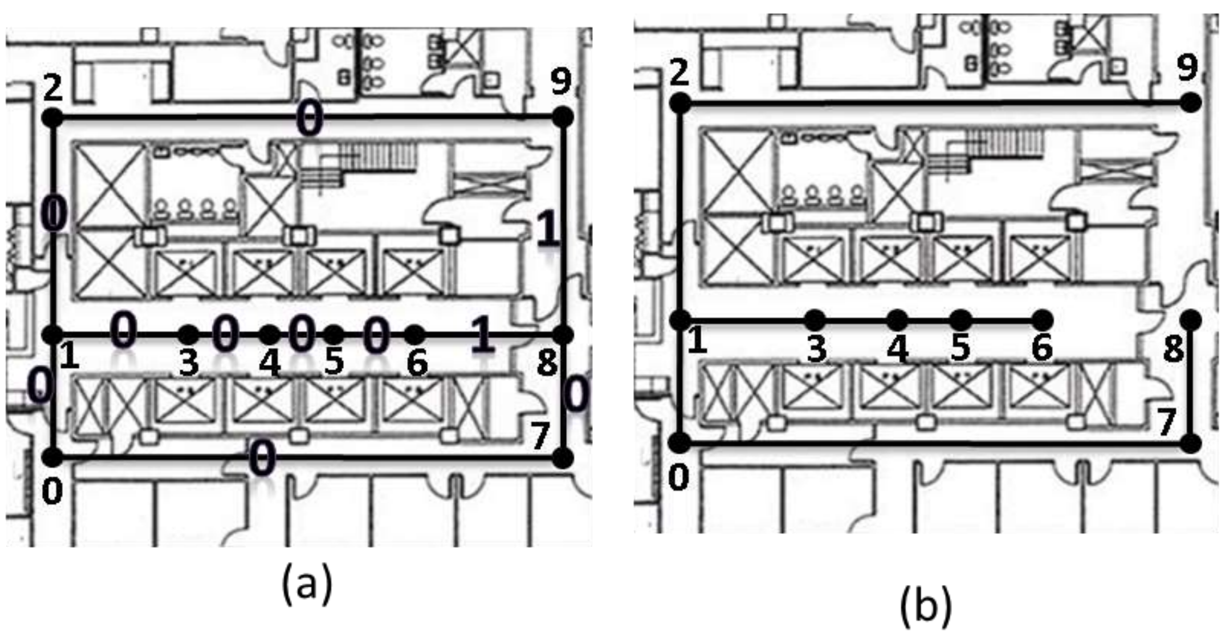


Figure 4.3 Attribute loaded graph network update example based on a specific binary attribute

4.3.2.2 Optimal Path Determination

This updated graph network is used to determine all simple paths between the start node and the destination node. Simple paths are obtained by avoiding repeating nodes in the path. That is to find all non-cyclical paths between the start and the destination nodes. The National Institute of Standards and Technology (NIST) online Dictionary of Algorithms and Data Structures describes finding all simple paths as a classic search algorithm and recommends using Depth First Search to find the solution (Black, 2017). It has to be noted that there can be infinite number of possible paths that contain cycles between two nodes in an undirected graph network. Thus, it is important to find all possible simple paths. A recursive algorithm was implemented to determine all simple paths between the start and destination nodes. The overarching idea is to start the search from the start node, store visited vertices in an array, and obtain the path once the destination node is reached. It has to be noted that the current vertices need to be marked as visited for the search to not go in cycles. Further details about the implementation can be found in (Gupta 2017).

The next step is to determine the cost for each of these possible simple paths determined. A generic weighted cost function can be represented as shown in the Eq(1)

$$C_p = \alpha_1 \Sigma A_{m_1} + \alpha_2 \Sigma A_{m_2} .. + \alpha_x \Sigma A_{m_x} + \beta_1 \Sigma A_{s_1} + \beta_2 \Sigma A_{s_2} + \cdots + \beta_z \Sigma A_{s_z}$$

Where $\alpha_1, \alpha_2, ... \alpha_x, \beta_1, \beta_2, ... \beta_z$ represent the corresponding weights for each of the edge attributes. Finally, the optimal path is chosen based on the costs estimated for each of these possible paths. Consider an example network shown in the Figure 4.4 where a represents the distance and b represents the ease of maneuvering in the corridor on a scale of 1-5 (1 being easy and 5 being the most difficult). If the start node is 0, destination node is 1, user prefers distance and ease of maneuverability equally then $\alpha = \beta = 0.5$. All possible simple paths between 0 and 2 would be 0 → 1 → 2 and 0 → 3 → 2. The corresponding costs for each of these paths would be C_1= 0.5(0.8+1.3) + 0.5(1+1) = 2.05 and C_2 = 0.5(0.8+1.3) + 0.5(2+2) = 3.05 respectively. Thus based on the chosen preferences, path 0 → 3 → 2 is ideal. On the other hand, if an $\alpha = 1$ is considered, then it is equivalent to finding the shortest path between nodes 0 and 2. That is, depending on the preferences of the users, the weights can be modified. Finally, users are guided along the determined optimal path with the help of markers along the way. Details regarding how the markers are placed and how the users are guided from one marker to another are provided in the fiducial marker network and communication section of this paper.

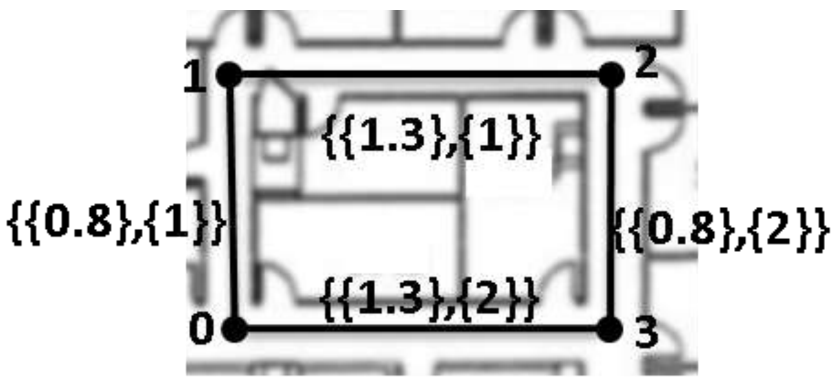


Figure 4.4 Graph network with multiple edge attributes

4.3.3 Fiducial Marker Network

At this stage, optimal path from the user's current location to the desired destination location is determined. As discussed earlier, markers (i.e. landmarks) need to be placed in the environment to localize and guide the user till the destination. To address the limitations of the fiducial marker based methods discussed in the background section of this paper, a methodology is proposed that explores the incorporation of verification markers (to verify and confirm that the users have taken the correct turn) and caution markers (to caution the users regarding the impending event). In addition, none of the previous studies explain the process of location selection for markers in the environment. That is, the process of selecting locations for installing markers in the building. Thus, this study categorizes the types of markers and explains the installation procedure of each of these types in conjunction with the created graph network. The proposed methodology also cautiously takes into consideration the placement of markers (e.g. on the walls or on the ceilings) with respect to the individuals with disabilities (i.e. camera on the wheelchair) which is a key aspect in the detection of the markers.

For the purpose of this study, authors categorize the markers into five different types namely general, turn, caution, verification, and level change based on the characteristics and limitations of the marker based navigation methods. This categorization also helps in identifying the locations in the building that needs a marker for marker guided navigation. The following are rules for deciding the location of each of the types of the markers. General markers (G) are all those type of markers which signify a notable and end of hallway locations in the building such as meeting room, information kiosk, restroom, and office rooms. That is, these markers are placed at the nodes which follow the rules ‘a’ and ‘d’. Turn markers (T) signify all the locations in the building where there is a possibility to turn. These are located at the nodes which follow rule ‘b’ of the node definition. The importance of caution markers (C) is to signify an approaching turn that the user might have to take. This type of markers usually precedes the turn markers described above. On the other hand, verification markers (V) are followed by turn markers. The importance of this type of markers is to ascertain the user regarding the turn they have just taken. For example, if the user takes a wrong turn (left instead of right or right instead of left), the corresponding verification marker directs the user in the correct direction. That is, the algorithm checks the verification markers and then determines that the user is headed in the wrong direction and redirects them in the right direction. As the name suggests, the level change markers (L) are located at places where the user can navigate from the current level or floor to a different floor. Depending on the situation, a location might be categorized into multiple different types and need multiple different types of markers. For example, consider node location 2. If that location also had stairs or elevators, that location might need turn marker and level change marker both since both the conditions are satisfied.

4.3.3.1 Marker Placement

Marker placement refers to the installation position of the markers such as ceilings, walls, and floors at the required locations in the building. Röhrig et al. (2012) showed significant promise in robot navigation using floor mounted markers in an industrial warehouse setting. However, this approach cannot be feasible in case of residential or commercial built spaces because of wear and tear caused by the frequent occupant and equipment (e.g. carts in a retail store) movement. Shneier and Bostelman (2015) described the range based wall mounted systems for indoor navigation. This approach suffers from occlusions (i.e. occluding camera detecting the markers) from frequent occupant movement especially in case of commercial built spaces. Additionally, at least two markers are required for defining each location resulting in comparatively more markers to be installed in the building. . For example, to define a particular room in a building's hallway, two markers are required, one on each wall of the hallway. This is because the camera will always be pointed to one particular direction.

Carlson and Demiris (2012) explored a ceiling mounted system for collaborative control of a robotic wheelchair. It was suggested that the markers are installed on the ceiling and the camera points directly above (i.e. perpendicular to the plane of the marker or ceiling) to detect the marker. However, this approach suffers from illumination sensitivity which might make the marker detection difficult sometimes. Based on the review of the aforementioned approaches, a combination of ceiling and wall mounted systems work the best for commercial and residential built environmental applications. Since, the current approach is designed for individuals with disabilities in buildings like hospitals, universities, and office spaces; a ceiling mounted system is preferred. This is because the markers are less likely to get damaged due to the occupancy movement indoors and consistent visibility for the camera without occlusions.

4.3.3.2 Marker Guided Navigation

This section describes the underlying procedure in guiding the users from the current location to the destination location with the help of markers. The overall logic of the marker guided navigation is shown as flowchart in Figure 4.5. First, fiducial markers are detected with the help of the onboard RGB camera and images are captured at a very high rate and are analyzed for the presence of the marker which is called segmenting. Second, the computer decodes the information from the markers in the form of 1's and 0's and determines the ID of the marker by cross referencing (matching) with the database of markers. Third, the current location, the relative pose and orientation information is extracted and interpreted (Olson 2011; Feng and Kamat, 2012). The location is determined with the help of the predefined locations of the markers. The orientation and pose estimation is done with the help of homogenous transformation matrix returned by the marker detection algorithm as shown in Eq(2) in which R (3*3) denotes rotation matrix and T (3*1) denotes translation matrix. A rotation matrix is representation of the relative rotation between the camera's and marker's coordinate systems. Thus, with the help of rotation matrix, the camera's orientation can be determined relative to the marker. Then, the directional arrow to be presented is determined based on the current marker position and the path to the destination location. This process is repeated until the destination is reached.

$$\boldsymbol{H} = \begin{matrix} \boldsymbol{R11} & \boldsymbol{R12} & \boldsymbol{R13} & \boldsymbol{Tx} \\ \boldsymbol{R21} & \boldsymbol{R22} & \boldsymbol{R23} & \boldsymbol{Ty} \\ \boldsymbol{R31} & \boldsymbol{R32} & \boldsymbol{R33} & \boldsymbol{Tz} \end{matrix} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.. \boldsymbol{Eq}(\boldsymbol{2})$$

Where:

H is the part of Homogeneous transform matrix returned by the localization algorithm

R (3*3): Rotation matrix

T (3*1): Translation matrix

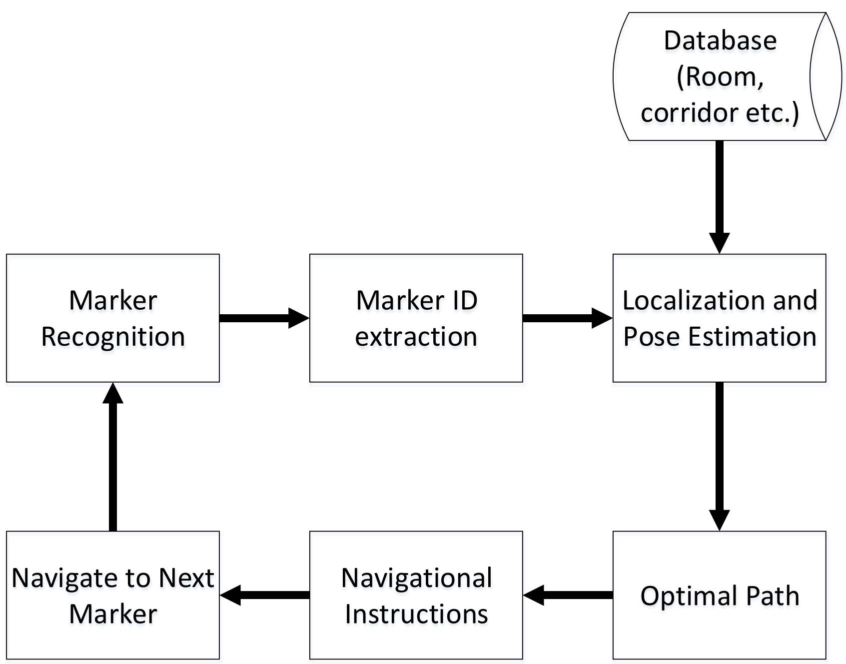


Figure 4.5 Overview of the marker guided navigation process

The overall logical flow of the algorithmic process described till now is shown as a flow chart in Figure 4.6. One of the possible scenarios that can happen during the navigation process is blocked paths (or edges to be more precise) due to several reasons such as maintenance activity (e.g. air conditioning repair), housekeeping (e.g. floor cleaning), reserved events (e.g. conferences), and emergency situations (e.g. fire). To address this issue, detour management is envisioned which routes the user through a different route to the destination considering all the inputs previously chosen by the user. The underlying principle behind this is the same where a temporary network update step is performed and is used for further steps with the rest of the algorithmic logic intact.

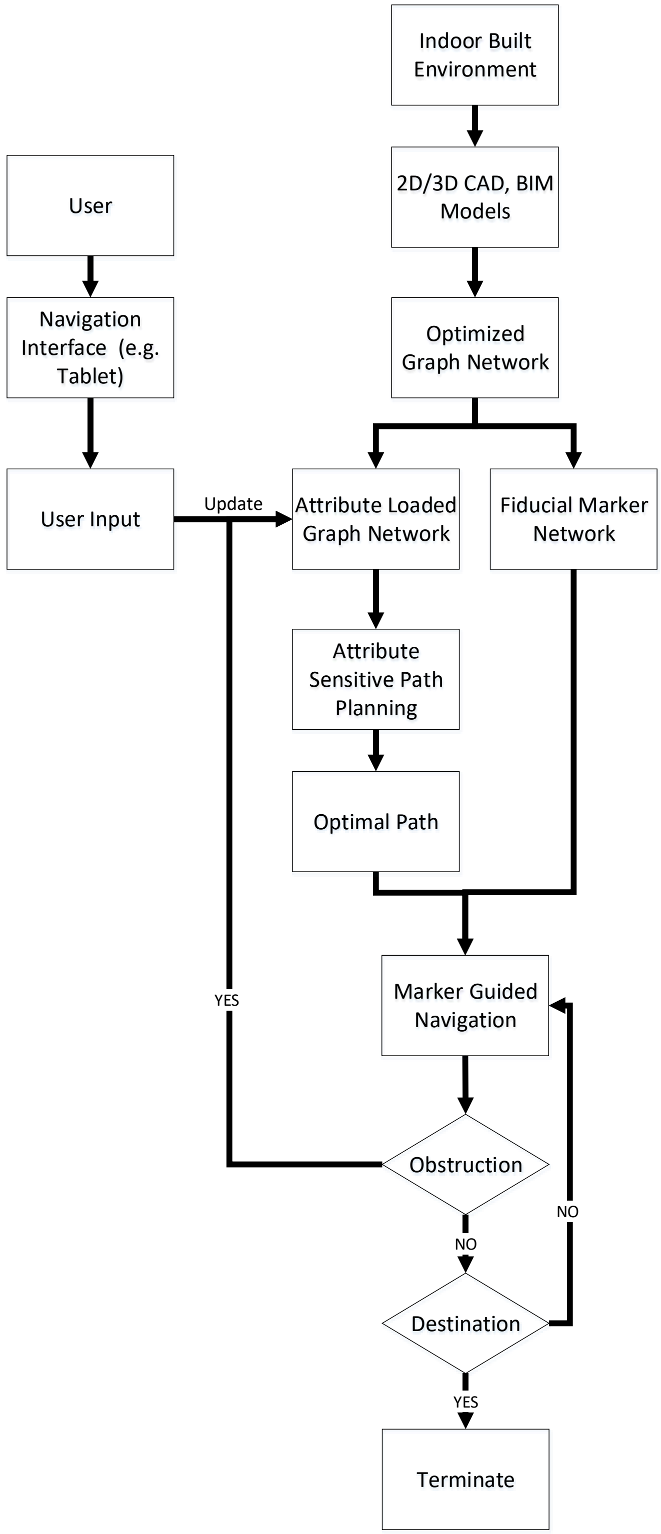


Figure 4.6 Algorithmic overview of the proposed general path planning framework for multi-layered graphs based on based on user preferences and constraints.

4.3.4 Communication

Several studies have explored using smartphone, tablet, and handheld device based applications/ interfaces for assisting pedestrians in indoor environments. For example, Oliveira et al., (2016) and Feng and Kamat (2012) proposed a smartphone based application which provides turn by turn instructions at strategic locations to the users with directional arrows on the smartphone screen. However, Li. et al (2013) suggest that spatial information regarding current location and navigational directions displayed on the map is most beneficial for navigation purposes. Table 4.1 shows the summary of several studies in the past decade that suggested the functional requirements of such an assisted indoor navigation interface.

None of the studies developed a system that incorporates all the functional requirements mentioned in the table. Specifically, the following three functional requirements a) Information regarding critical waypoints ahead of time (Li et al. 2013; Ziefle et al. 2007) – That is, the user is prompted or cautioned just ahead of time regarding the approaching turn, elevator or destination; b) Path summary with waypoints (Rehrl et al. 2007) – That is, information regarding the summary of determined path is provided to the user before the start of the navigation; and c) Preferential constraints (Rehrl et al. 2007) (as discussed in sections 4.3.1 and 4.3.2) were found to receive the least amount of attention. In an aim to address these critical research gaps, the authors developed a navigational interface dedicated to individuals with disabilities that encompasses all the aforementioned functional requirements. The interface developed is platform independent and allows for accessibility, type, and preferential constraints.

Table 4.1 Functional requirements of an assistive navigation interface for individuals with disabilities

Source	Description	Wang et al. 1997	Fernández -Madrigal et al. 2004	Montesano et al. 2010	Oliveira et al. 2016	Neges et al. 2015	Lie et al. 2013
Li et al. 2013; Ziefle et al. 2007	Information regarding critical waypoints ahead of time	x	x	x	x	x	x
Rehrl et al. 2007	Path summary with waypoints	x	x	x	x	x	x
	Preferential constraints	x	x	x	x	x	x
	Orientation feedback	x	√	√	x	√	√
	Position feedback	x	√	√	x	√	√
Wen et al. 2013	Forward up map	x	x	x	x	√	√
Li et al. 2013	Spatial information in the map	x	√	x	x	√	√
	Detour management	√	x	√	x	√	x
Li et al. 2013; Arning et al. 2012	Enriched map with landmark information	x	x	x	x	√	x

Human machine interface is a critical factor in determining the user's acceptance of the interface (Adlam 2003). Prior efforts have developed either android or iOS based application but time and effort has to be put in to develop each of them separately. However, HTML based applications can be instantly used (does not require installation) and work on several browsers across all devices. Thus, an interface is developed based on HTML for the user to visualize the map, input the data, and view the output (e.g. navigation instructions). The home screen of such an interface is shown with an example floor plan view in Figure 4.7. The current layout shows the entire floor plan in the center with user input toolbar in the top right corner with a black background. An enlarged user input controls is shown in Figure 4.8. The floor plan shown is based on the marker scanned when entered the building. The default floor plan however is the first floor of the building. Details regarding each of the components of the user interface are described in the following sections of this paper.

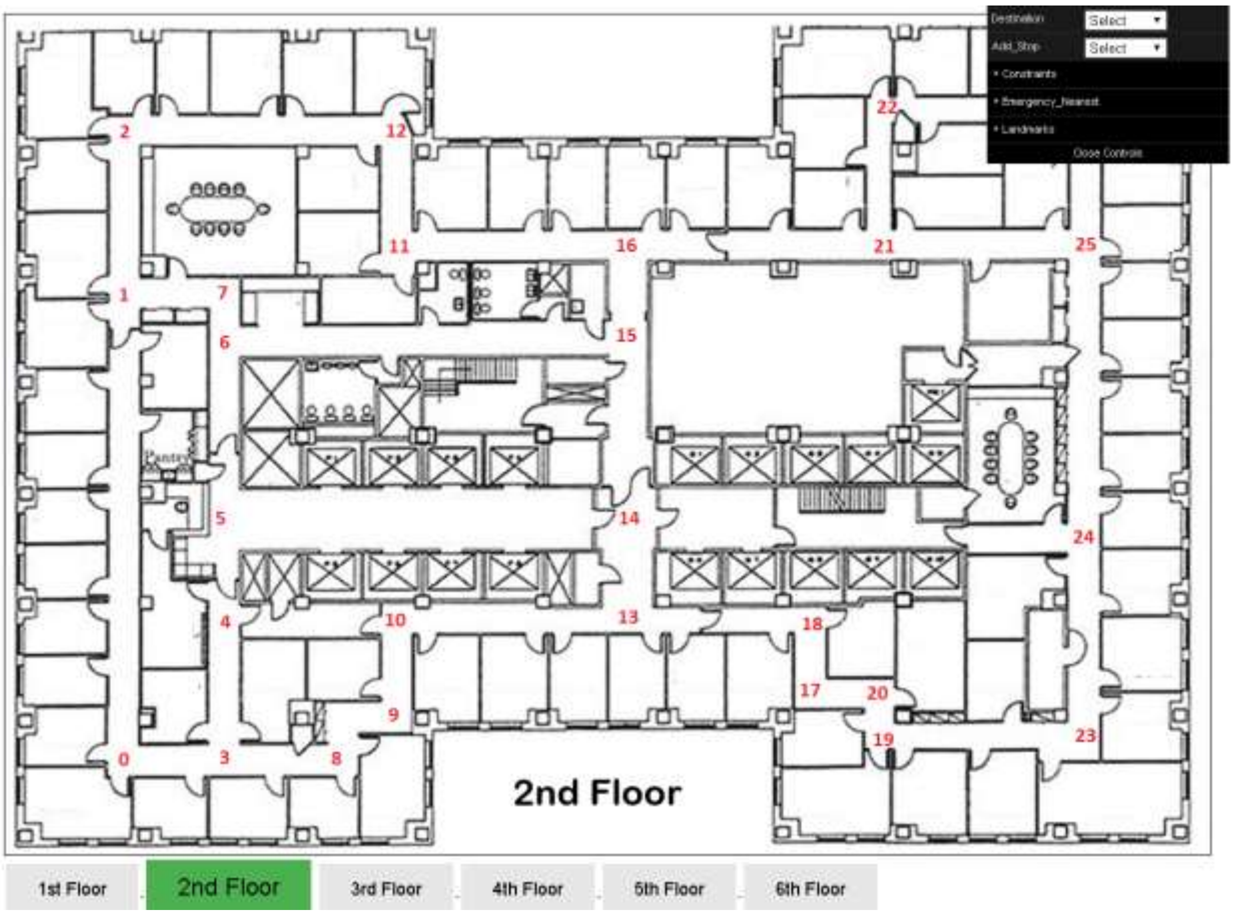


Figure 4.7 Assisted navigation interface layout with an example floor plan and user controls

4.3.4.1 User input

The user input controls are divided into four layers each layer corresponds to a specific category. The first layer consists of primary inputs regarding the user intended destination with an option for adding a stop in between. Rehrl et al., 2007 emphasizes the significance of a personalized route calculation that takes user's specific preferences into consideration. Thus, the second layer consists of possible mobility constraints and preferences such as avoiding stairs and least number of turns. The third is an emergency mode layer which aids the user in reaching the nearest exit or emergency room in case of emergent situations such as fire and tornado. Studies suggest that enriched landmark information is beneficial to the users of high and low spatial ability (Li et al. 2013; Arning et al. 2012). That is, providing such information will better assist users in their wayfinding abilities and enable them to achieve higher navigational performance. Therefore, the last input layer contains information regarding the notable landmarks in the current floor. For example, specific landmarks (e.g. vending machines) will help the user in orienting themselves and taking turns at complex junctions in the building. On the other hand, it also helps the user to view the nearest information desk in a shopping mall or checkin counter in a hospital. An example user input control with all the layers is shown in Figure 4.8.

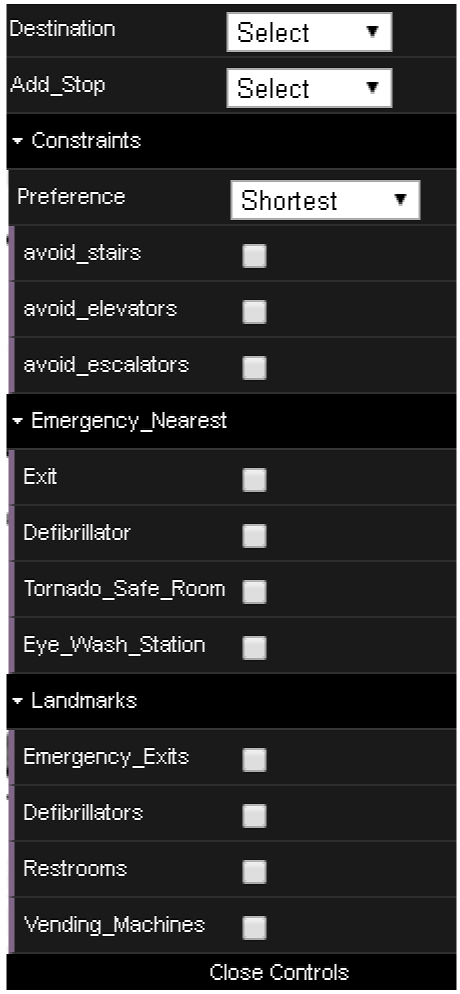


Figure 4.8 Example user input controls for a certain facility with all the input layers

4.3.4.2 Path summary

Before starting the trip, i.e. during the pre-trip phase, information regarding the path summary helps the user to better visualize and prepare for the trip. For this purpose, the current visualization provides a) path overview – overview of the path including current location, destination location, and critical waypoints on the floor plan of the building; b) total distance and time – the total distance of the trip in miles and expected duration of the trip; c) total turns – since a turn to be taken is a significant event for a individuals with disabilities especially indoors, the total number of turns till the destination is also included in the summary.

4.3.4.3 Position and orientation feedback

Positioning is the prime basis for indoor wayfinding or assisted navigation (Rehrl et al., 2007). As discussed earlier, the position of the user is determined with the help of an on-board camera and known locations of fiducial markers in the indoor environment. This study extends the wall and ceiling mounted systems discussed in marker placement section to provide visual feedback of the orientation to the user as previously discussed in the marker guided navigation section. For example, the position and orientation of the user is shown with a solid green arrow as shown in Figure 4.9.

4.3.4.4 Real time path

Li et al. 2013 suggests that spatial information displayed on the map is the most advantageous for indoor wayfinding. In order to provide better contextual information to the user, in addition to the current position and orientation, the navigational path information is also provided. This includes the path traversed till now and path to the destination form the current location. To differentiate between the aforementioned paths, the path traversed till now is shown as lightly shaded green and the path to the destination is shown in dark green from the current location to the destination as shown in Figure 4.9. Furthermore, it has to be noted that the destination is highlighted with a red blob on the floor plan and the current position with an arrow pointed in the user’s headed direction as discussed.

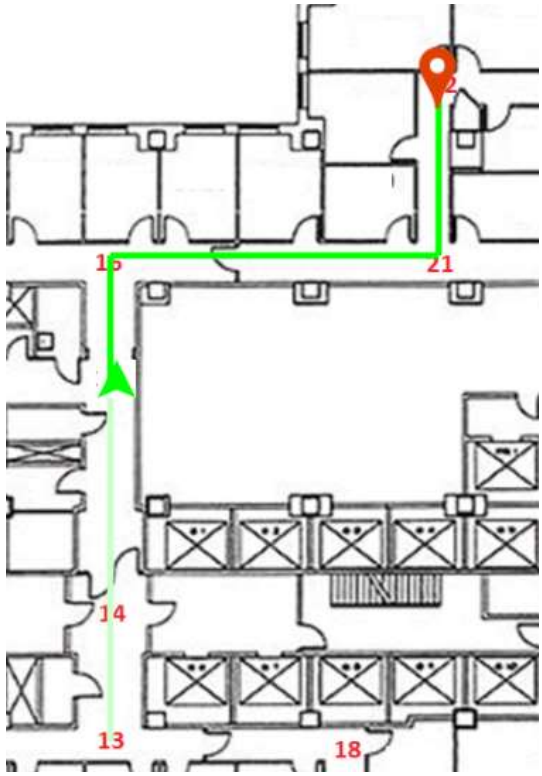


Figure 4.9 A trip with current location (green triangle), destination location (red pin), path traversed till now, and path to the destination.

4.3.4.5 Caution

Several studies suggested providing the user with enough caution during critical waypoints when providing turn by turn instruction (Li et al. 2013; Ziefle et al. 2007). In addition, Rehrl et al. 2007 suggested that, providing turn by turn instructions with observable physical natural landmarks in the environment instead of simple turn by turn instructions. That is, instead of “walk straight for 10 meters and turn right” the instructions can be as follows “walk up till the end of corridor and turn right towards the water fountain”. Thus, this study employs both and provides users with instructions regarding specific landmark information. These are specific instructions associated with the respective markers located in the physical environment. This will not only enhance the user’s navigational experience but also helps better interact and relate with the physical environment. In addition, to enhance the visual experience of the user, the map is scaled (i.e. zoomed in) to better show the approaching turn as shown in Figure 4.10.

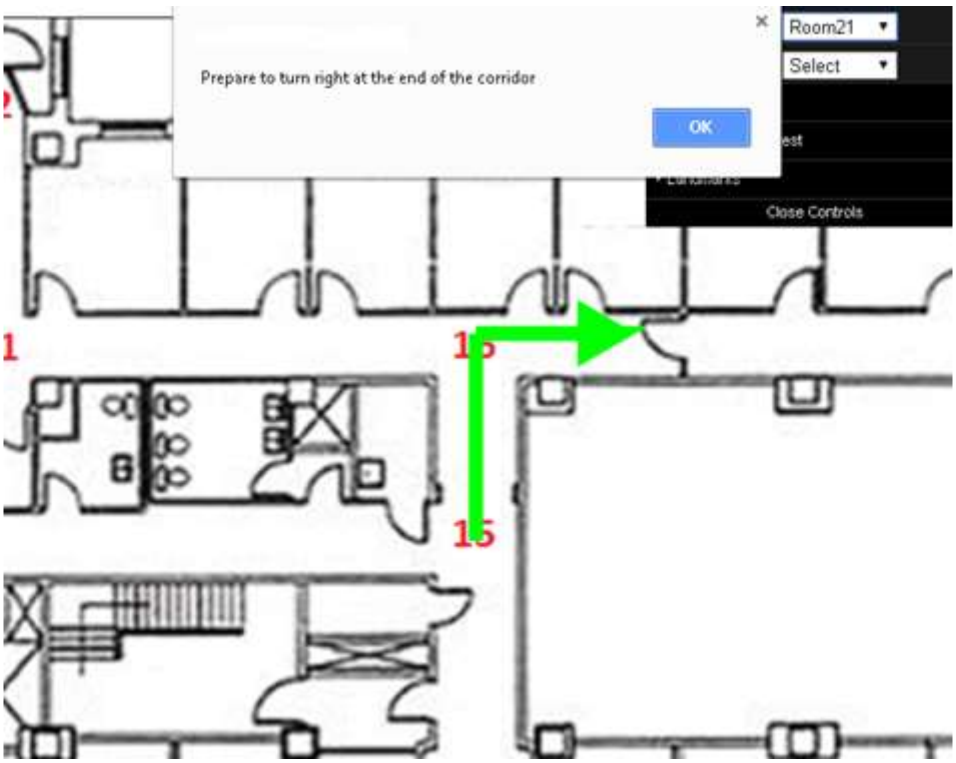


Figure 4.10 User is cautioned regarding the approaching turn with text instructions (dialogue box on the top) and also by zooming into the map and enhancing the view.

4.3.4.6 Verification

While following turn by turn instructions, if the user misses a turn or takes a wrong turn, the user is unaware of the mistakenly performed action until the next marker is scanned. This is one of the major drawbacks of the marker based systems and none of the previous studies considered such a situation and addressed it. As discussed earlier, this study addresses this gap by including a verification marker, which prompts the user with text instructions (e.g. alert message as shown in Figure 4.10) confirming the previous action performed. For example, consider a situation where the user turned left and was supposed to turn right. The alert message in this case can be “You have taken a wrong turn, please turn around and proceed in the other hallway”. Thus, verification markers are located at the beginning of the turns (i.e. branches or junctions) inside the building. Figure 4.11 shows an example representation of placement of verification markers (shown as black solid squares) in the physical space with four markers at the four way junction (i.e. one in each branch or hallway) and three markers at the T-shaped junction. It has to be noted that fiducial markers (April tags) are small in size and can be installed in indoor environments in a non-intrusive way with least amount of retrofit and require very less to no periodic maintenance.

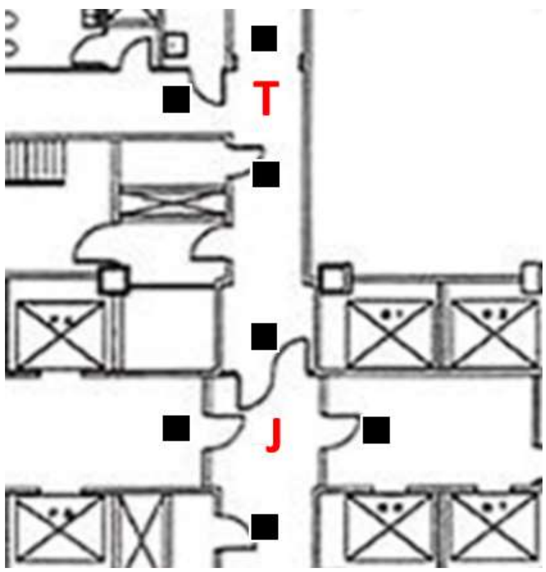


Figure 4.11 Verification marker (shown as solid black squares) placement at two junctions with possible turns in the indoor built environment where J represents four way junction and T represents T- shaped junction.

4.3.4.7 Viewing notable landmarks in the facility

Studies suggest that an enriched map with landmark information can not only improve the visibility of the navigation information but also help address the disorientation issues faced by many of the users in the environment (Li et al. 2013; Arning et al. 2012). This study provides the user with optional information regarding some of the significant landmarks in the environment such as emergency exits, information desks, food courts, restrooms, and water fountains. One or multiple of these landmarks can be added to the map view by selecting and deselecting the respective landmark in the user input layer. For example, if the restrooms option is selected in the landmarks layers, the restrooms accessible to the individuals with disabilities located in the floor are shown with red blobs as shown in Figure 4.12.

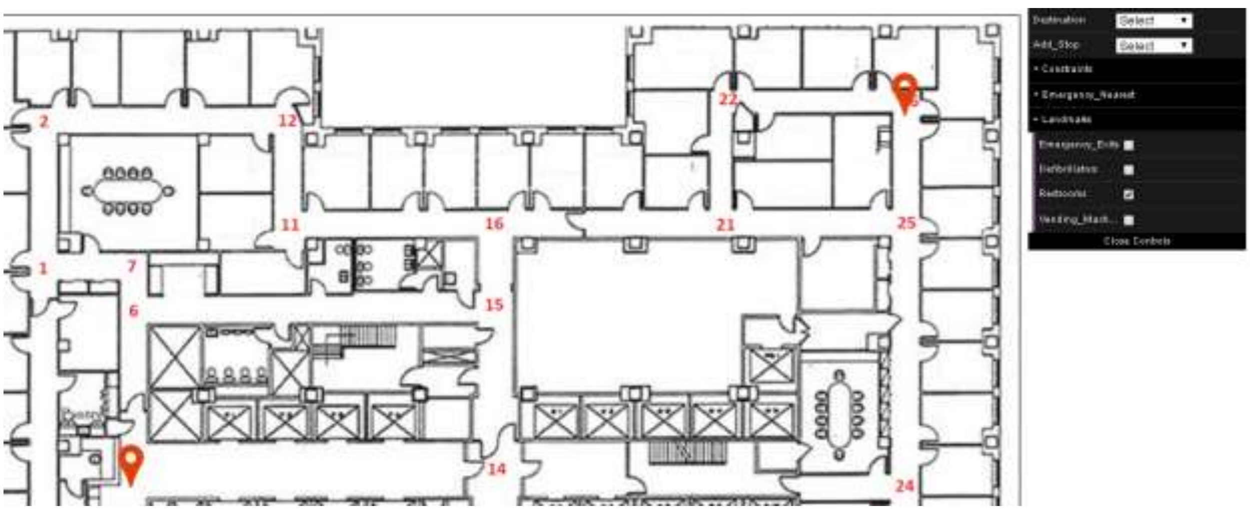


Figure 4.12 Example map view with enriched landmark information where the red blob represents the respective landmark chosen.

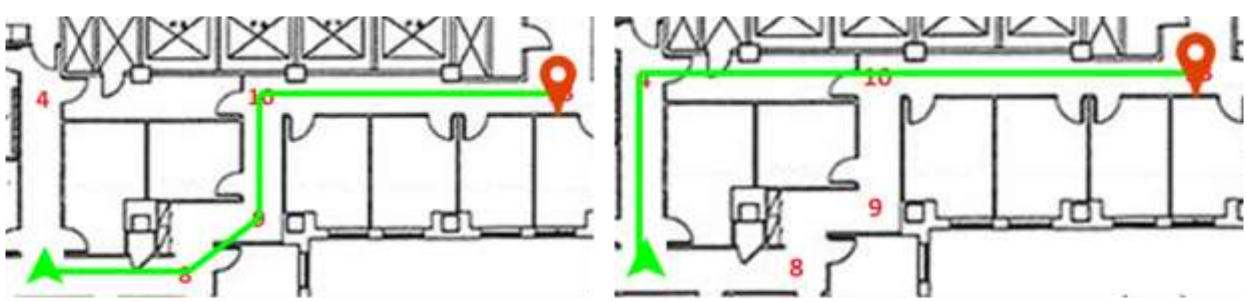
In addition, the user is also provided with the option of selecting the ideal landmark based on the current location, user chosen preferences, and constraints. That is, at any instance, the user can choose to view or route to any specific landmark with the chosen preferential constraints. For example, anytime during the navigation the user can choose to show the route to the nearest exit in case of emergency or optimal route to the check in counter in case of a hospital visit.

4.4 Case Study

The floor plan of a large building is chosen for simulation experiments, and is shown in Figure 4.7 The graphical representation of the reduced node network consists of 27 nodes and 33 undirected edges or arcs. Four different scenarios are chosen to show the workability of the proposed framework and are described in following subsections.

4.4.1 Scenario 1: Preferential Constraints (shortest vs least number of turns)

The goal of this scenario is to determine the shortest path between two locations (or nodes), path with least number of turns, and compare them. For example, consider node 3 as the current location and node 13 as the destination location as shown in Figure 4.7 . The path shown in Figure 4.13a is the shortest as determined by the algorithm with the help of measured (distance) type attribute. The path in Figure 4.13b is based on least number of turns, but is not the shortest path to the destination. It can be observed that based on the current orientation of the user at the current location, the user has to take four turns to reach the destination in case a compared to that one turn in case b. This situation might be especially helpful to users with manual wheelchair compared to that of powerchair users because of the additional physical effort involved. That is, the user has to turn and orient along the path in the physical environment at every turn.



(a) (b)

Figure 4.13 Comparison of shortest distance (a) and path with least number of turns (b) between start (arrow) and destination (red blob) nodes.

4.4.2 Scenario 2: Add Stop

Consider a situation where the user needs to make a quick detour to the nearest restroom or water fountain on their way to a meeting room. This scenario compares and evaluates the results of a specific example. For example, consider the start, destination, and intermediate (mid stop) locations as 5, 11, and 15 respectively. In Figure 4.14a, the shortest path from 5 to 11 is computed for the individuals with disabilities. On the other hand, Figure 4.14b shows the shortest path based on the intermediate passing location 15 with start and destination locations being the same. The algorithm initially estimates the shortest path from the current location to the intermediate destination and then to the final destination. It is coincidental that the number of turns in case of Figure 4.14b (three) is lesser compared to that of Figure 4.14a (four) and it need not be the case always.

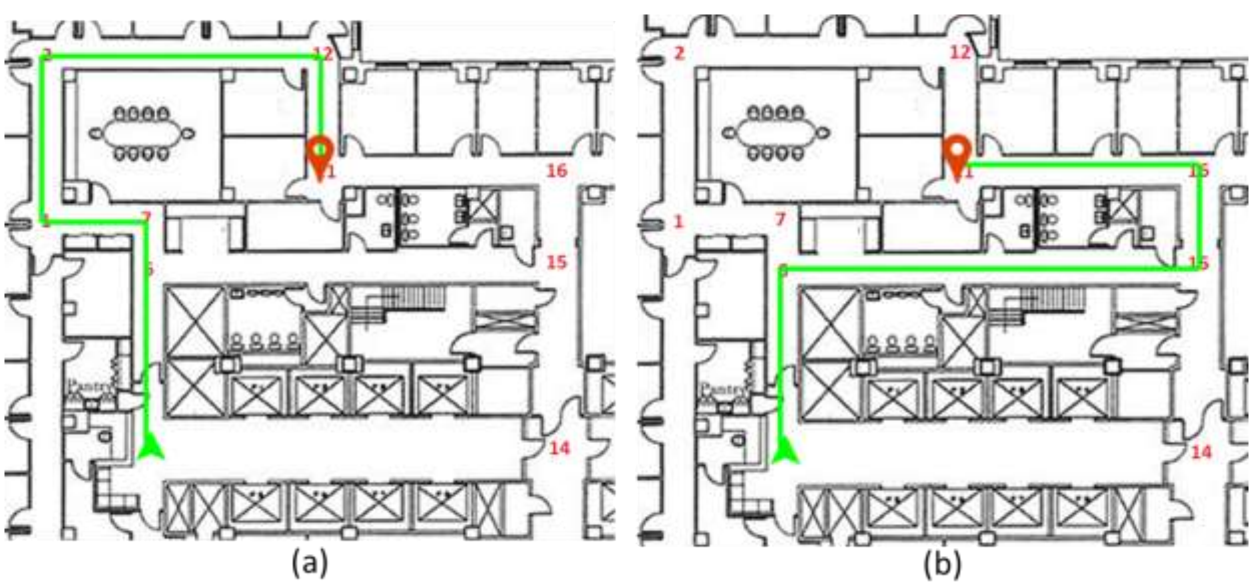


Figure 4.14 Comparing paths with (b) and without (a) intermediate stop (node 15) where arrow represents start location/orientation and red blob represents the destination.

4.4.3 Scenario 3: Detour Management

Detour Management is yet another feature of the algorithm, which enables individuals with disabilities to reach their destination in case of any obstruction in the pre-selected route. As discussed previously, the network update step will omit the particular edge and plan the rest of the part from the current location to the desired destination location in accordance with the user preferences. Figure 4.15 shows the results of an example scenario of a real time obstruction in a building. Figure 4.15a shows the shortest path from location 5 (arrow) to location 2 (red blob). Consider an obstruction between locations 1 and 7, which temporarily prohibit any movement through that particular edge or hallway or corridor. In such a case, the algorithm re-routes the path through locations 6-15-16-11-12-2 in real time, as shown in Figure 4.15b. It has to be noted that this may not be the shortest path from the intended start and destination locations.

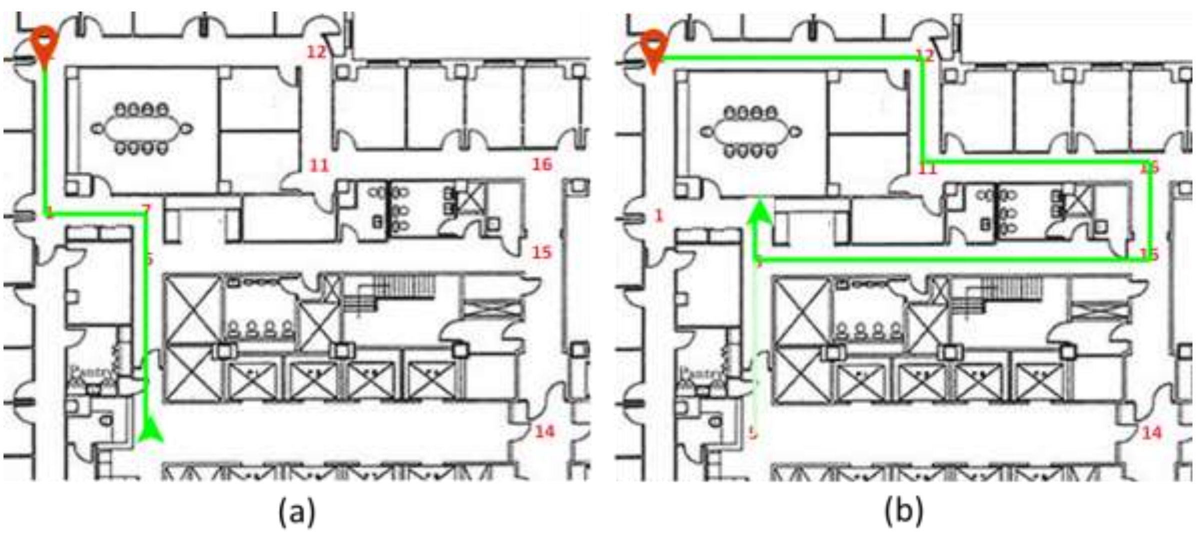


Figure 4.15 Comparison of path results with and without real-time obstructions

4.4.4 Scenario 4: Missed Marker

The goal of this scenario is to compare the results of paths with and without verification markers. That is to evaluate and investigate the differences between ideal and actual path traversed by the users in case of missed markers. For example, consider a situation where the users intended destination location is 21 and the start location is 24. The shortest path returned by the algorithm is superimposed on the plan and is shown in Figure 4.16a. If due to any reason, the user misses to take the turn at location 25, based on the traditional marker based systems, the user is not redirected until the next marker is scanned (i.e. at location 26). Once the marker at this location is scanned, the user is redirected to the intended destination location from the current location (location 26 in this case) through 22 to 21. As discussed earlier and from the current example it can be observed that the total distance of the trip increased from 10.8 units to 15.2 units as shown in Figures 16a and 16b. In addition, it also increases travel time and physical burden on the disabled individual. However, with the help of verification marker (proposed methodology) this can be avoided as the user is immediately prompted (with the help of an alert pop up window as discussed previously) after crossing the junction location 25 as shown in Figure 4.16c. In this case, the user is prompted of the missed turn and is redirected to the intended path. Thus, there are no additional travel time, physical burden, and distance implications on the user. Though the proposed approach also suffers from a similar limitation of missing the verification marker, it atleast adds an additional layer of safety and assists the user better compared to the traditional marker based navigation systems.

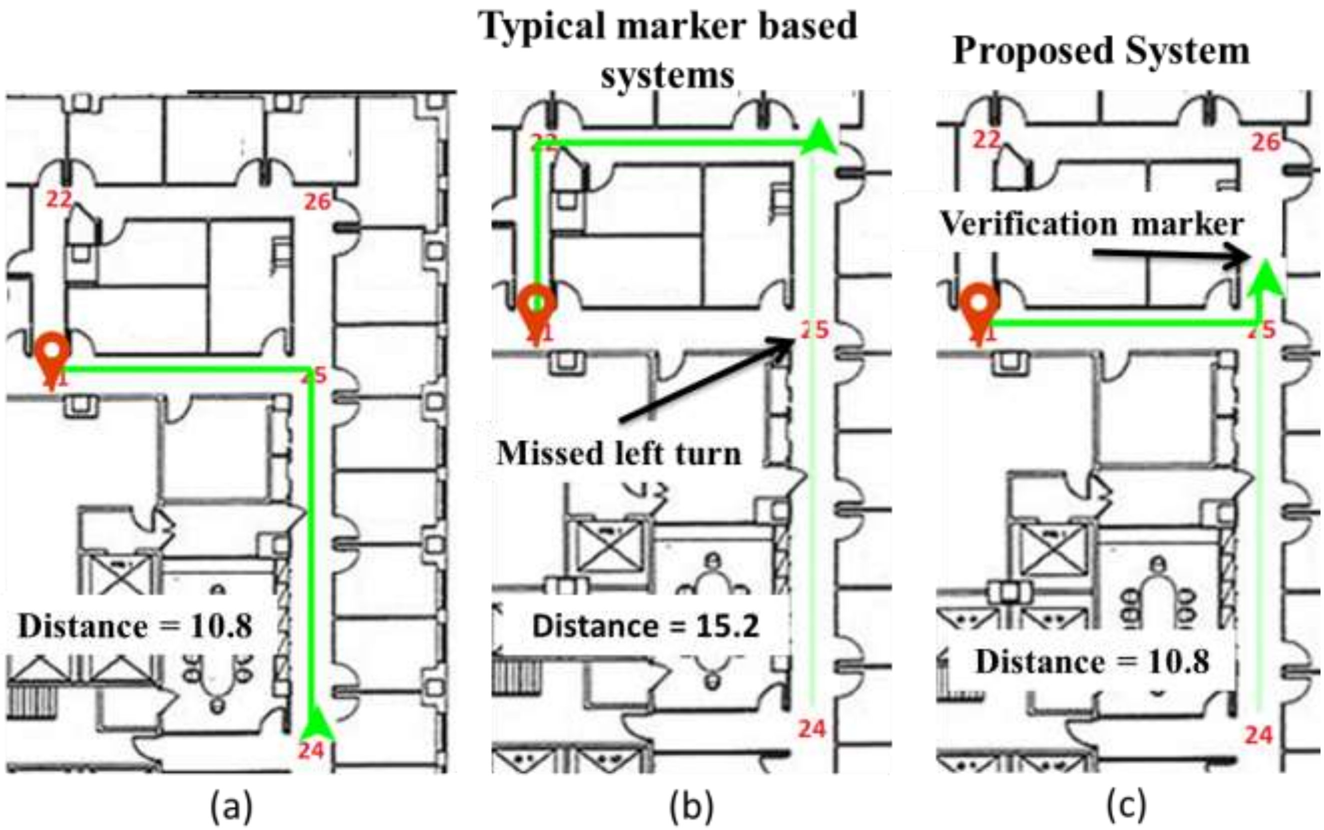


Figure 4.16 Implication of missing a marker during the navigation. A comparison of results between the traditional marker based systems (b) and the proposed approach (c).

4.5 Pilot Testing Of The Navigation Application

The objective of this step is to evaluate the usability of the prototype navigation application developed for individuals with disabilities. As part of clinical testing process, the preliminary step is to test the application with the help of individuals without disabilities. This is done to understand the application interface's design needs, concerns, priorities, and draw insights on the performance and usability of the application before testing it on people with mobility impairments (Yanco 2001; Yanco and Gips 1998). Pilot testing experiments were conducted in the George Granger Brown Building at the University of Michigan - Ann Arbor campus with the help of a manual wheelchair as shown in Figure 4.17. The adjustable tray shown acts as a rest for the tablet and can be easily shifted forward or backward depending on the convenience of the participant. The figure also shows the tablet interface used by a participant and the camera used for the purpose of localization.

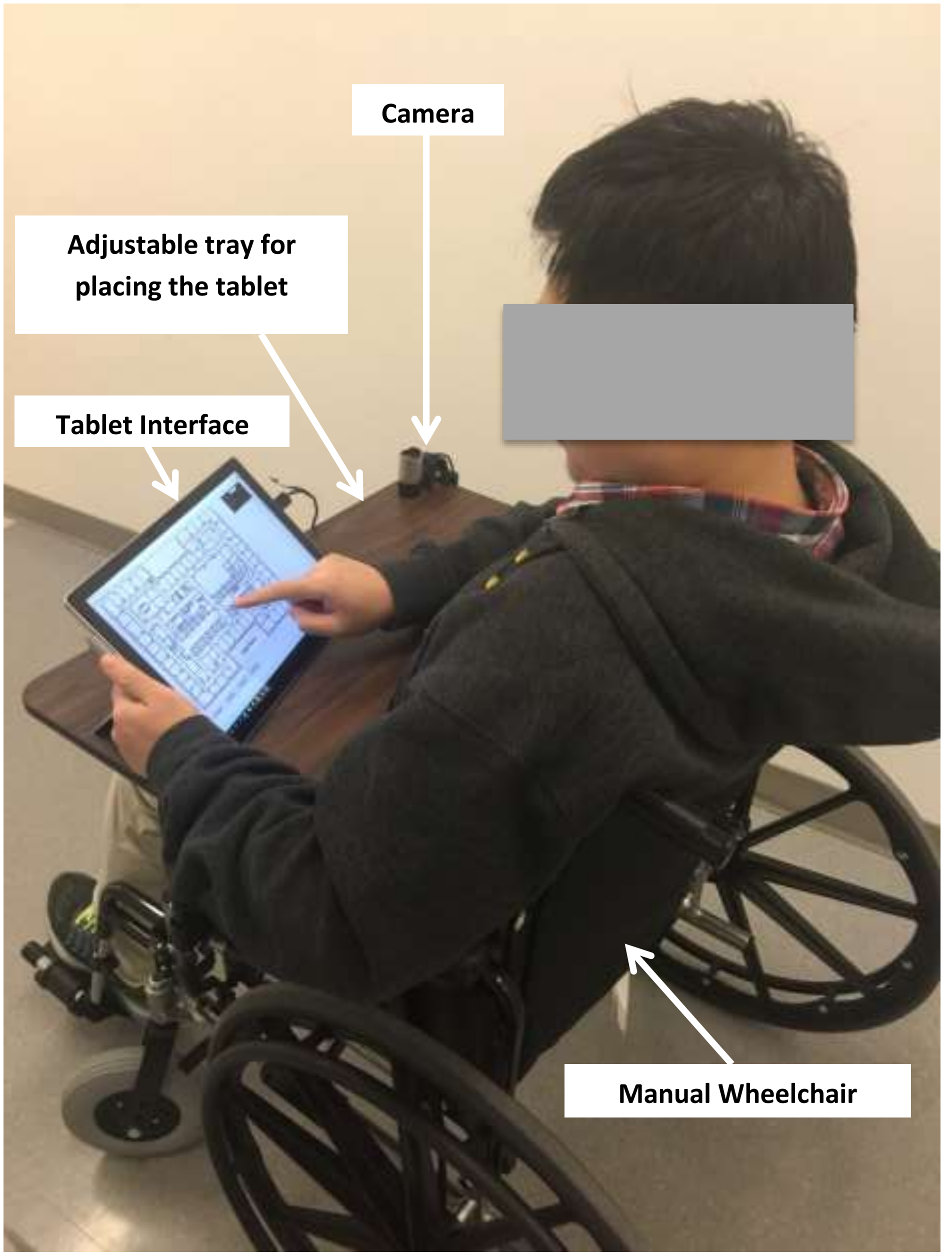


Figure 4.17 Manual wheelchair for testing the indoor navigation application

The experiments comprised of the following three phases: Phase I - Training with the wheelchair and application interface, Phase II - Actual navigation tasks, and Phase III - Basic

Questionnaires and Post-trial interview. The study was approved by the Institute Review Board (IRB) (HUM00142295) at the University of Michigan. All the participants were made aware of the procedural details of the experiments and were notified of the possibility of publishing the results in scientific journals.

4.5.1 Phase I: Training with the wheelchair and application interface

After obtaining the participants consent to proceed with the study, the participant was seated in a manual wheelchair in a comfortable posture and a quick demo of the navigation application was given. Then, the participant was provided with instructions on the basic operations of wheelchair along with its safety features. After this, the participant was given time to get comfortable with the wheelchair and the navigation device.

4.5.2 Phase II: Actual navigation tasks

In this step, two to three target destination locations were provided to the participant starting from room 1340 as shown in Figure 4.18. For example, the participant starts at 1340 GGBrown and the first destination location provided was 1324. Once the participant successfully navigates to 1324, the second targeted destination location of 1368 was provided. Data regarding successful target completions and possible missed or wrong turns taken were gathered during the navigation tasks.

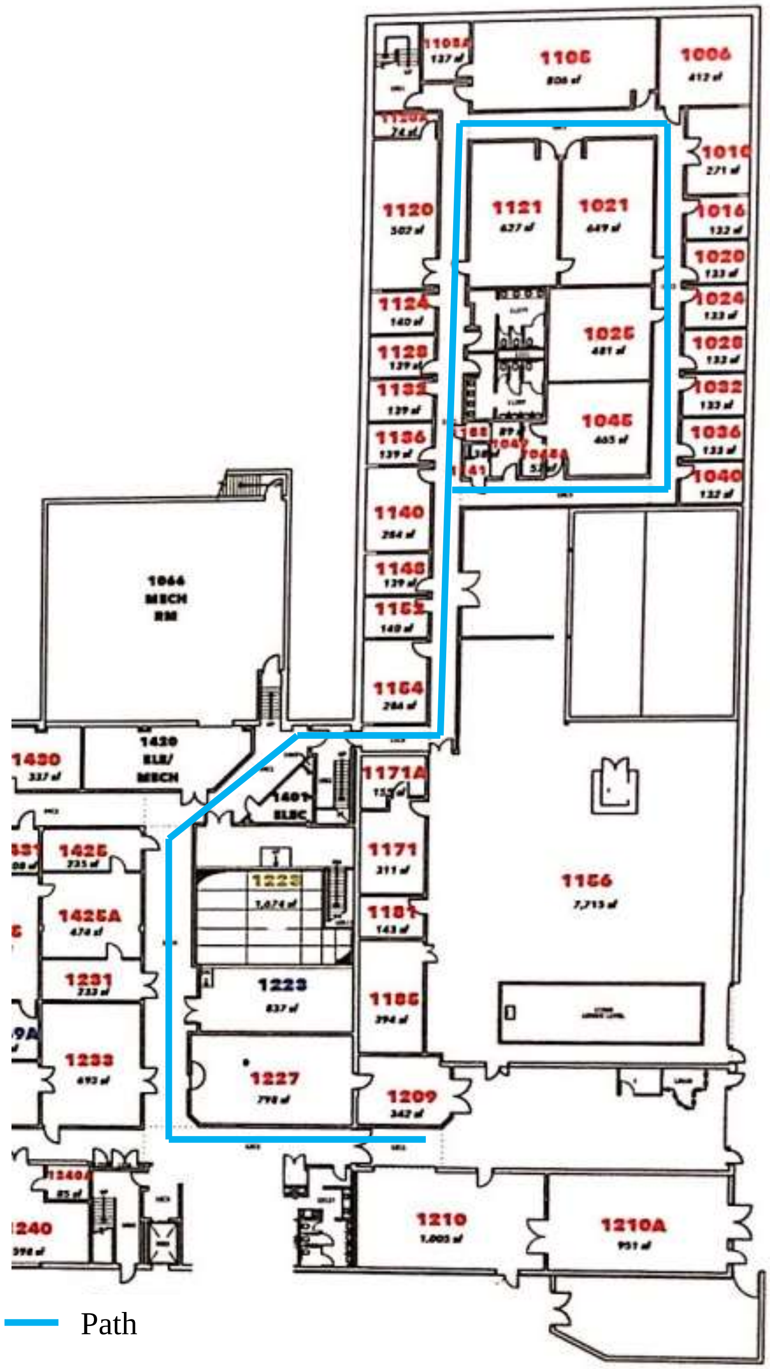
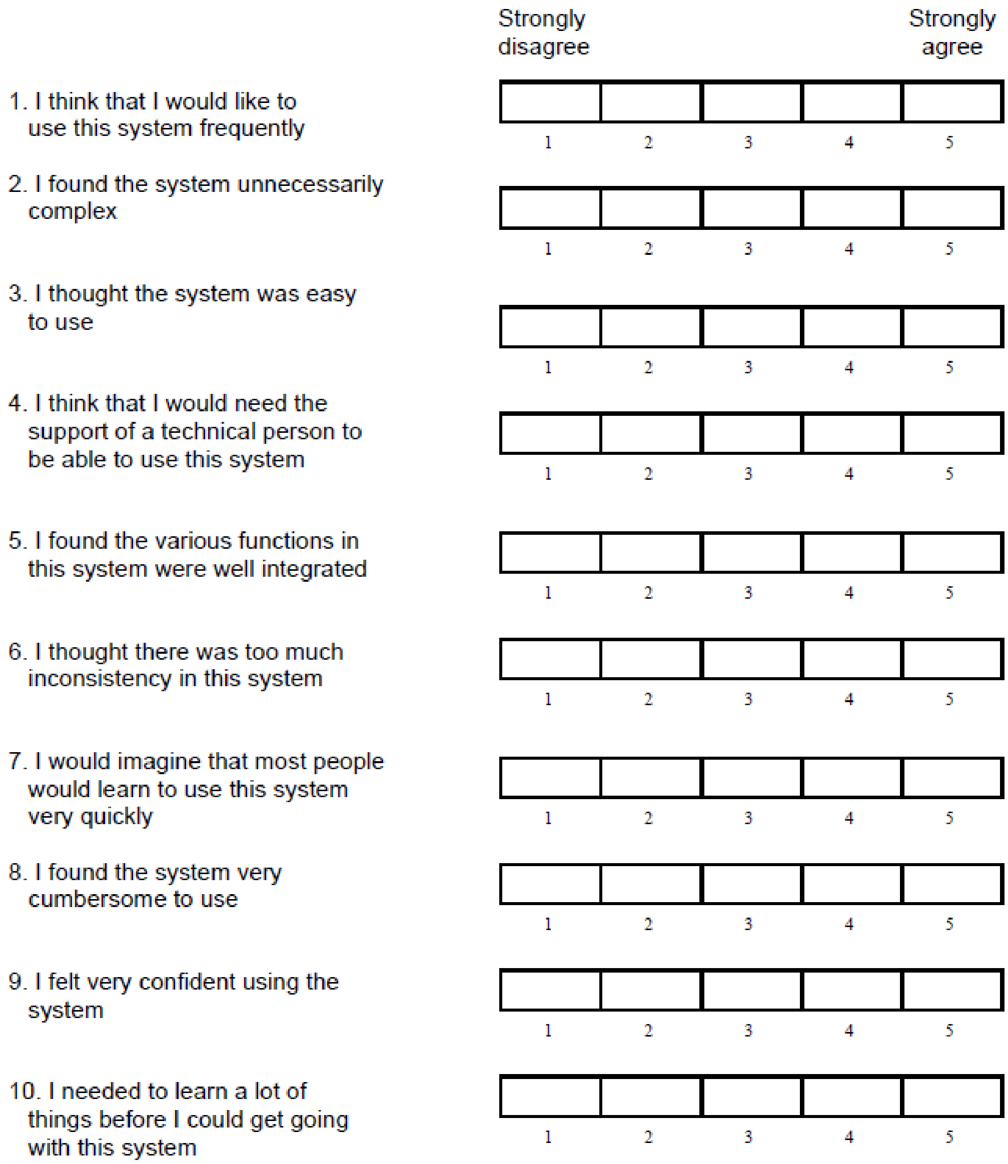


Figure 4.18 Plan view of the 1st Floor, GGBrown at University of Michigan, Ann Arbor Campus

4.5.3 Phase III: Basic Survey and Post-trial interview

First, key demographic information (e.g., age, gender) and experience in using smart phone or tablet based applications was obtained from the participants from a paper-based survey. Second, an in-person semi-structured interview was conducted to get feedback on the overall performance of the navigation application along with any challenges faced. A sample set of the questions posed were as follows "*What is your overall opinion on the performance of the application?", "Was the home screen easy to comprehend?", "Did you find it difficult to use the application?", "Was the application helpful in giving you a sense of your current location, the path, and the destination location?", "Would it be helpful if audio instructions were included along with visual prompts?"*. The aim of this step is to investigate the impact of the applications aid in successful task accomplishment, the performance variation between males and females, the role of hand dominance with the orientation of the tablet interface and thus the comfort of the user. The participants were also asked to fill the System Usability Scale (shown in Figure 4.19) as part of the post-trial interview questionnaire for deriving insights on the ease of usability of the interface application (Brooke 1996).



	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Figure 4.19 System Usability Scale, questions and response scales (Source: Brooke, 1996)

4.6 Results

There were a total of 10 participants (8 males and 2 females) who volunteered for the study. The ages of the participants varied between 22 and 35. Almost all of them had experience using smartphone or tablet based applications. Though some of the participants were a little familiar with the building, the building was recently renovated and thus, none of them were aware of the exact location and the path to the destination. A total of 23 destinations and 26 turns were involved.

All the participants successfully navigated to their target destinations showing the workability of the proposed approach. However, one of the participants had trouble with orientation (i.e. confusion between left and right) and took a wrong turn but the verification marker helped route the participant back to the intended path. In addition, most of the participants felt that the application was helpful and easy to use. The following discussion summarizes the feedback of the participants regarding the navigation interface. Most of the participants replied it was easy to comprehend the home screen of the application but one of the participants suggested adding an introduction or brief description of different elements on the home screen. Almost all of the participants agreed that the application gave them the sense of their current location, the path and the destination location. However, one of the participants felt having a legend would make it easier and another participant preferred to have a separate indicating blob for the start location as well. In addition, a participant suggested showing an estimated time and distance to the destination along with the path to give a better sense of the journey ahead.

While majority of them liked visual prompts (70%), they would also like audio prompts with a few exceptions of no prompts at all. Especially some of the participants commented that they wouldn’t prefer audio prompts due to privacy concerns. 90% of the participants preferred to have an option to view the notable landmarks in the facility but 10% wanted all the information on one map with unique symbols for each of the landmarks. In addition, few of the participants wanted to have an option to navigate to the nearest landmark with the corresponding availability information. For example, occupancy status of the nearest restroom and the current wait time of the nearest coffee shop.

Finally, the SUS score was estimated based on the participants responses and the procedure detailed in Brooke 1996. SUS is chosen because it is reliable and robust to evaluate the usability of any kind of system. The score ranges anywhere between 0 and 100. It has to be noted that it is not a percentage and the higher the number the better the usability of a system. The average score of all the participants' responses based on the individual responses shown in Figure 4.20 was estimated to be 85 with a least score of 70 and highest score of 95. Based on the industry standards, it can be said that the navigation interface is reliable and the participants would recommend the usage of this application to other users (Sauro 2011). Starting from top left each of the charts consists of responses for two of the questions. For example, the top left chart consists of responses for questions 1 and 2. Similarly, top right consists of 3 and 4, middle left consists of 5 and 6, middle right consists of 7 and 8 and finally the bottom consists of 9 and 10. Each of these charts also consists of mean average responses as shown. The following are some of the observations based on the question-wise responses of all the participants a) Almost all the participants replied they would use such a system frequently (question 1) with a few exceptions. The exceptions might be because all the participants in the study were graduate students without mobility impairments b) Though significant portion of the participants felt confident using the interface (question 9), some of them suggested a demo or legend explaining different features of the application will be helpful c) Few of them opined that the system was cumbersome to use (question 8) due to the font size. In addition, one of the participants did not feel confident at all using the system (question 9). The authors believe this might be because of the misinterpretation of the question or the response scale since the participant successfully navigated to three destinations using the interface without any errors (e.g. missed turns) or prompts (e.g. researcher helping the participant).

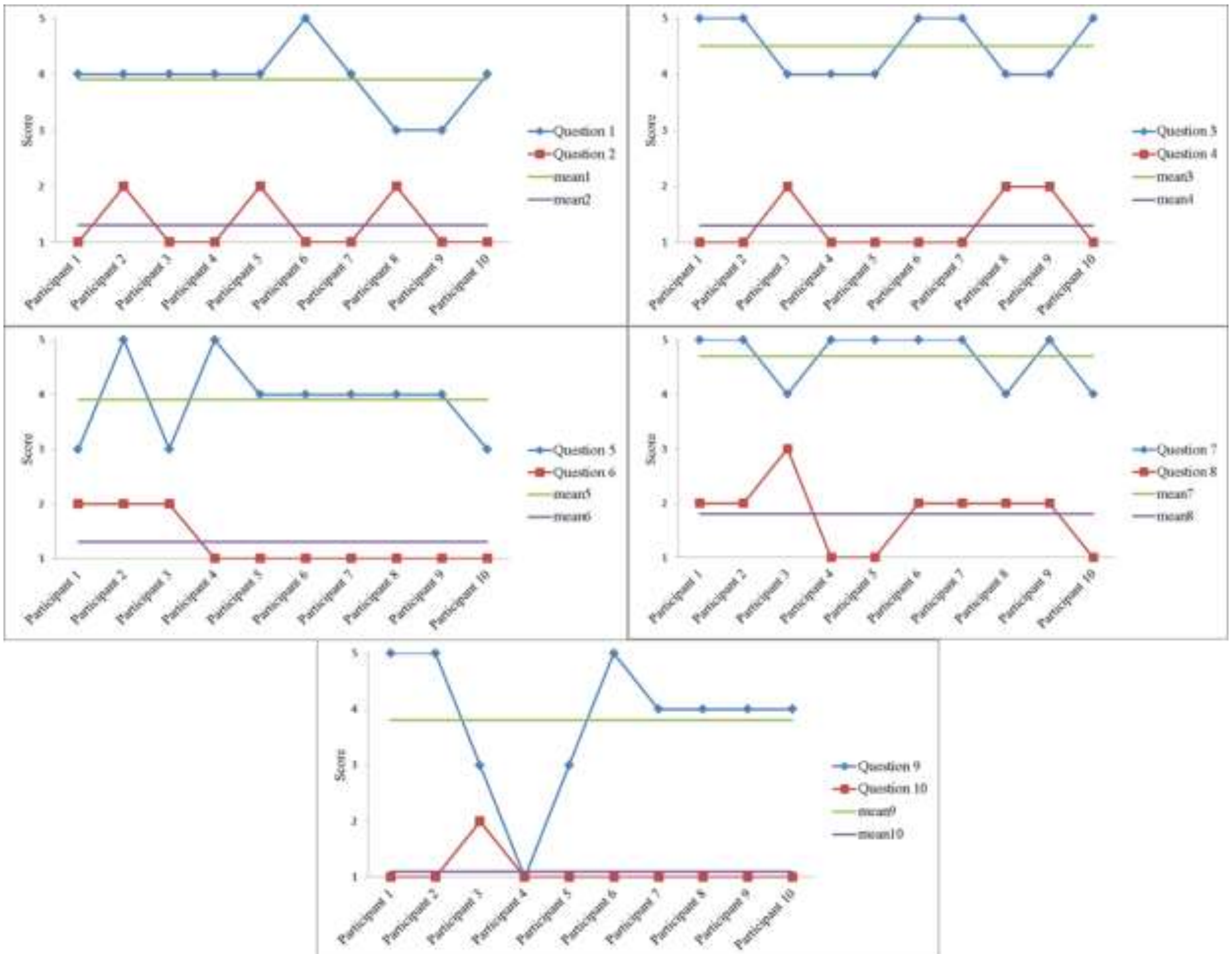


Figure 4.20 System Usability Scale (SUS) responses for all the questions answered by the participants.

4.7 Conclusions

This study proposes a generic navigation algorithmic approach for assisting individuals with disabilities in complex indoor built environments. Some of the key contributions include basic rules for creating an indoor graph network, establishing the significance of attribute loaded graph networks, developing a generic algorithm for determining the optimal path (instead of shortest path only) from the attributed loaded networks which is not context specific, rules for marker network map creation, and development of a User Interface (UI) that demonstrates and integrates everything together. A scenario analysis was also conducted to show the feasibility of the proposed algorithm with and without constraints such as preferential (e.g. least number of turns), detour, and missed markers.

In addition, pilot test experiments were conducted with the help of ten individuals without disabilities to determine the design and functional efficiency of the developed interface application. Results showed significant promise in successfully completing several navigation tasks. 70% of them preferred to have visual prompts for navigation and 90% of them favored viewing the notable landmarks in the facility. Based on all the combined responses of participants, an average SUS score of 85 was estimated showing the workability and reliability of the developed system. Furthermore, some of the main recommendations of the participants regarding the application are as follows a) Brief instructions regarding the usage of the application on the home screen b) Auto complete search box to search for the possible destinations c) Display the estimated time and distance to the destination along with the path d) Option to view the start location during the course of navigation along with legends that show the current and destination location e) Additional information (e.g. estimated wait time at the nearest security check in at an airport) regarding the nearest landmarks in the facility.

Chapter 5

Conclusions and Future Work

Recent advances in computing and robotics have enabled the possibility of widespread deployment of service robots for several indoor building environment applications. Potential examples include inventory management, hospitality, security, and surveillance to name a few. This dissertation is focused on two main applications building operation and utilization. That is, investigate the potential of semi and completely autonomous robotic systems to efficiently monitor, operate, and use the co-inhabited existing buildings.

For example, consider group of robots that need to perform several tasks (e.g. deliver equipment in hospitals) i.e. visit different locations in an indoor building environment. Any of the aforementioned task-oriented robots require four key technical capabilities a) Task Allocation – optimally distribute the locations among themselves b) Tour/Path Planning – plan their respective paths i.e. determine the order in which each of these locations needs to be visited c) Localization – have the ability to locate itself in the indoor environment and d) Navigation – navigate itself to the desired locations of interest. Most of the existing robotic systems that perform these key functions are either context or application specific and lack a general and scalable framework especially for indoor building environment applications. To address these key research gaps, this dissertation develops general and rapidly deployable framework and algorithms related to navigation, tour/path planning, and task allocation for mobile robotic applications in indoor building environments. The main methods adopted for developing these algorithms and frameworks include combinatorial optimization, graph traversal based path planning, fiducial marker based localization, landmark-based navigation, and building energy simulation.

Thrust 1 of this research developed an efficient iterative algorithm that assists single or group of robots to optimally divide tasks between themselves and plan their respective tours in complex indoor building environments with and without constraints such as feasibility, accessibility, and occlusions. A scenario analysis was conducted to compare the performance of existing algorithms with the developed approach. It was found that the proposed method performs better compared with other algorithms in almost all the cases considered. Some of the key findings include a) some of the existing algorithms (e.g., k-means clustering) do not work as well for indoor environments as they do for outdoor environments. This is because k-means clustering does not take into account the network topology (information regarding the edges in the network) b) informed initialization can reduce the number of iterations to determine the optimal path c) distance restrictions might have to be imposed for avoiding unequal division of labor d) frequency of tasks visited was significantly affected by the location of base node and the size of network. The results demonstrate the feasibility of the proposed approach in a range of applications involving constraints on both the environment (e.g., path obstructions) and robot capabilities (e.g., maximum travel distance on single charge).

In thrust 2, basic rules to create indoor graph networks and marker network maps which form a basis for indoor path planning, marker-based localization, and navigation were established. In addition, developed a preferential path planning algorithm based on attribute loaded indoor graph networks that takes preferences (e.g. least number of turns) and pragmatic constraints (e.g. real-time obstructions such as inaccessible corridors due to temporary maintenance activity) into consideration. The algorithm was validated with the help of a representative floor plan and scenario analysis. Results suggest that paths significantly vary from the traditional shortest paths. However, it has to be noted that the preferences needs to be preloaded in the algorithm. Then, the proposed algorithm was used to assist people with disabilities indoors as part of the case study application. This was conducted to show the general applicability of the proposed algorithm for indoor building environments. Though the application involves assisting people with disabilities indoors, this can be used for general population or robots (e.g. semi-autonomous or autonomous navigation of the robotic wheelchair) for that matter. A navigation interface was developed and the pilot testing of the application was conducted with the help of individuals without disabilities navigating on the wheelchairs. Ten participants completed different navigation tasks using the application and all of them successfully reached their destinations without any errors. The results from the survey and semi-structured interviews conducted after the experiments suggest that the navigation application was reliable and almost all the participants recommend the use of this application. Some of the key recommendations and preferences of the participants are as follows a) audio prompts along with visual prompts b) option to go to the nearest landmarks c) introduction/demo of the home screen and the usage of the application. To take this further, the application can be improved based on the feedback by the participants and then test it with the help of individuals with disabilities in a real physical environment.

At this stage, the robots have the capability to determine what do to (task allocation) and how to do it (path planning). To further this research, thrust 3 developed fiducial marker (april tag) based navigation and drift correction algorithms for achieving autonomous indoor robot navigation. The objective of this thrust was to achieve successful indoor robot navigation using inexpensive, highly reconfigurable, and rapidly deployable sparse localization technique based on markers. The developed algorithms were validated by testing it on a turtlebot in a real building. Results suggest that marker to marker distance and marker placement play a key role in the successful navigation task accomplishment. As part of the case study application, the algorithms were also tested to gather ambient data using autonomous multi-sensor fused indoor mobile robots. In addition, a case study was performed to show how the data collected (by the mobile robot) can be processed, analyzed and subsequently used for building energy retrofit decisions as a proof of concept

The following sections provide a summary of the future research directions that can be considered as continuations to this study.

5.1 Future Directions of Research

5.1.1 Multi-Base Depot Optimization for Indoor Robotic Service Networks

The proposed methodology of multi-robotic task allocation and route planning uses predefined location selection (randomly choosing) of the base node and only a single base station (start and end node location) was considered for the optimization algorithm. To address these limitations, further methods need to be investigated for optimizing the initial start location(s) [or the depot(s)] instead of random selection. In addition, the existing framework can be extended to multiple start and destination depots with and without feasibility, accessibility, and occlusion constraints, i.e., different robots starting at different start depots and all ending at their respective nearest destinations. Lastly, these algorithms can be tested on actual robotic systems in several social settings to further investigate privacy concerns.

5.1.2 Sensor Suite Design for Robotic Data Collection

Information regarding some of the data types such as appliance state and plug load cannot be collected without a fixed sensor in each room because continuous monitoring is required for assessing the energy consumption and the appliance state throughout the day. Thus, it could be interesting to explore installing some sensors (receiving nodes) on the mobile robot that will receive information from the sensors fixed inside the room when the robot is nearby. The sensors fixed inside the rooms will be in a standby/ sleep state unless activated by the receiving nodes (in this case the ones placed on the mobile robot), i.e., more energy efficient than having fixed sensors in the room for transmitting the same data.

5.1.3 Data Fusion, Aggregation, and Visualization of Data Collected by Robots

This research validated the use of single and multiple robots for monitoring and collecting ambient data in buildings. The data gathered by different robotic platforms need to be synchronously aggregated for subsequent validation, data storage, visualization and/or real-time analysis (online simulation). Thus, several data fusion and visualization techniques need to be investigated to manipulate, summarize, and visually represent the asynchronous location specific floor and room level granular data. Furthermore, statistical validation of the aggregated data collected with the help of multiple robots (swarms) and the Building Automation Systems data collected with the traditional stationary networks will have to be performed.

APPENDICES

Appendix 1General instructions and the complete code for performing the iterative algorithm

1. Download the code which is publicly made available at the following link
2. Please note that all the user inputs are in the _main_ function at the end of the document. It is a python code and the file can be executed in windows, linux, or mac platforms without errors on python 2.7 or above. No additional dependencies needs to be installed.
3. The algorithm in its current form works without alterations for undirected graph networks. If desired, modifications can be made in the user input section to the variable "listOfEdges" for directed graphs. This can be done by modifying the graph connectivity details in the list of lists. Please note that the first two values of each list are the node numbers and the third value is edge attribute.
4. Please note that the numbering of the vertices starts at 0 and not 1.
5. The readers are requested to cite this study if benefitted from this algorithm.

Appendix 2 Informed consent for the assisted navigation interface usability evaluation experiments

Informed Consent

Consent to Participate in a Research Study

Title of the Project: Usability Testing of Assisted Indoor Navigation Application for Disabled Individuals with the Help of Nondisabled Individuals

Principal Investigator: Carol C. Menassa, Ph.D., Associate Professor, Civil and Environmental Engineering, University of Michigan

Co-investigators: Vineet R. Kamat, Ph.D., Professor, Civil and Environmental Engineering, Clive D'Souza, Ph.D., Assistant Professor, Industrial and Operations Engineering, University of Michigan

Invitation to Participate in a Research Study

We invite you to be part of a research study about evaluating the indoor navigation interface on a wheelchair. Insights from the study along with the proposed technology can greatly improve independence in disabled users while also reducing associated costs on wheelchair attendants or caregivers.

Description of Your Involvement

The experiment protocol will require approximately 60 – 75 minutes of your time (not counting the informed consent process). If you agree to participate in the study we will ask you:

1) To take the General Survey at the start of the study (approximately 2-5 minutes)

2) You will be asked to sit in a wheelchair in a comfortable posture while we will give a quick demo of the navigation application. We will also give you instructions on how to operate the wheelchair along with the safety features. You will be given ample time to get comfortable with the wheelchair and the navigation device. This step will approximately take 10-15 minutes.

3) Once the training phase is finished, we will ask you for your consent to proceed with the actual navigation tasks. Two to three target destination locations will be provided to you starting from the lab. We will be present at all times with you to ensure safety of the participant. During the process, we will also obtain data regarding successful target completions and possible missed or wrong turns taken.

4) Following completion of the navigation tasks, we will conduct a semi-structured interview to obtain feedback on the overall performance of the navigation application along with any challenges faced. You will also be asked to fill the paper-based NASA Task load Index and System Usability Scale as part of the post-trial interview questionnaire.

Benefits of Participation

Although you may not directly benefit from being in this study, others may benefit because the results will help us develop future research on designing indoor wayfinding applications for disabled individuals that guarantee their independence.

Risks and Discomforts of Participation

We don't believe that there are any risks or discomforts from participating in this research.

Confidentiality

Your data will be stored and referenced using a randomly generated id. No sensitive personal information (e.g., name, address, phone number) is collected in this experiment.

We plan to publish the results of this study. Since the data is already anonymized, there will be no identifying information in the published work.

It is possible that other people may need to see the information you give us as part of the study, such as organizations responsible for making sure the research is done safely and properly like the University of Michigan and government offices.

Storage and Future Use of

We will store your data for future research studies in an M Box account that is only accessible to the research team working on this project. This account is password protected. Since the data is transmitted anonymously to our database, there will be no identifying information associated with the data.

Your name and any other identifying information will not be stored anywhere.

Only the research team will have access to this signed consent form. This consent form will be securely stored for the duration of the study in the Department of Civil and Environmental Engineering at the University of Michigan.

Voluntary Nature of the Study

Participating in this study is completely voluntary. Even if you decide to participate now, you may change your mind and stop at any time. You do not have to answer a question you do not want to answer/share. If you decide to withdraw before this study is completed, **your data will be completely deleted and no record of it will be kept.**

Contact Information for the Study Team

If you have questions about this research, you may contact the PI's Ph.D. student Bharadwaj Mantha at baddu@umich.edu. You can also contact the PI at menassa@umich.edu.

Contact Information for Questions about Your Rights as a Research Participant

If you have questions about your rights as a research participant, or wish to obtain information, ask questions or discuss any concerns about this study with someone other than the researcher(s), please contact the: University of Michigan Health Sciences and Behavioral Sciences Institutional Review Board 2800 Plymouth Road Building 520, Room 1169 Ann Arbor, MI 48109-2800 Phone: (734) 936-0933 or toll free, (866) 936-0933 Email: irbhsbs@umich.edu

Consent

By signing this document, you are willingly agreeing to participate in this study. The research team will give you a copy of this document for your records. The research team will keep one copy of this document securely stored as indicated above.

At this time, we don't expect to receive any financial benefits from this research. However, since we are always looking for new ways to improve personal environments, it's possible that the data from this research could one day lead to a product or device from which the University and our researchers could benefit.

I agree to participate in the study

____

Printed Name

____

Signature   Date

BIBLIOGRAPHY

ADA (2010), “ADA standard for accessible design”, https://www.ada.gov/regs2010/2010ADAStandards/2010ADAStandards_prt.pdf [Accessed 18th September, 2017]

Ahmad, A., Maslehuddin, M. and Al-Hadhrami, L.M., 2014. “In situ measurement of thermal transmittance and thermal resistance of hollow reinforced precast concrete walls.” *Energy and Buildings*, *84*, pp.132-141.

Akyildiz, I. F., and Kasimoglu, I. H. (2004). “Wireless sensor and actor networks: research challenges.” *Ad hoc networks*, *2*(4), 351-367.

Ali, A. I., and Kennington, J. L. (1986). “The asymmetric M-travelling salesmen problem: A duality based branch-and-bound algorithm.” Discrete Applied Mathematics, 13(2-3), 259-276.

Alm N, Arnott JL, Murray IR, Buchanan I. (1998) “Virtual reality for putting people with disabilities in control.” *IEEE International Conference on Systems, Man, and Cybernetics*; 1998; San Diego, USA. San Diego: IEEE; 1998. p. 1174-9.

ASHRAE. (2013). “Energy standard for buildings except low-rise residential buildings.” *ASHRAE/IESNA 90.1*, Atlanta, GA, USA.

Autodesk Sustainability Workshop. (2015). “Total R-values and thermal bridging.” < http://sustainabilityworkshop.autodesk.com/buildings/total-r%20-values-and-thermal-bridging> [Accessed 20th September, 2016]

Balta, H., Bedkowski, J., Govindaraj, S., Majek, K., Musialik, P., Serrano, D., Kostas A., Roland S., and Cubber, G. (2016). "Integrated Data Management for a Fleet of Search-and-rescue Robots." Journal of Field Robotics.

Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2004). Estimation with applications to tracking and navigation: theory algorithms and software. John Wiley and Sons.

Bellman, R. (1962). "Dynamic programming treatment of the travelling salesman problem." Journal of the ACM (JACM), 9(1), 61-63.

Bellmore, M., and Nemhauser, G. L. (1968). "The traveling salesman problem: a survey." Operations Research, 16(3), 538-558.

Bhadauria, D., Tekdas, O., and Isler, V. (2011). "Robotic data mules for collecting data over sparse sensor fields." *Journal of Field Robotics*, 28(3), 388-404.

Biswas, J., and Veloso, M. (2010). "Wifi localization and navigation for autonomous indoor mobile robots." *In Robotics and Automation (ICRA), 2010 IEEE International Conference on* (pp. 4379-4384). IEEE.

Blattner, A., Vasilev, Y., & Harriehausen-Mühlbauer, B. (2015). "Mobile indoor navigation assistance for mobility impaired people." *Procedia Manufacturing*, *3*, 51-58.

Boltr (2014), "SaviOne robot butler starts room service deliveries this week" online http://www.savioke.com (May 16, 2016).

Boucher, Patrice, Amin Atrash, Sousso Kelouwani, Wormser Honoré, Hai Nguyen, Julien Villemure, François Routhier, Paul Cohen, Louise Demers, and Robert Forget. (2013). "Design and Validation of an Intelligent Wheelchair towards a Clinically-Functional Outcome." Journal of Neuroengineering and Rehabilitation 10 (1): 1.

Brault, Matthew W. 2012. "Americans with Disabilities: 2010." Current Population Reports 7:

0–131.

Brooke, J. (1996). “SUS-A quick and dirty usability scale.” *Usability evaluation in industry*, 189(194), 4-7.

Brumitt, B. L., and Stentz, A. (1996), “Dynamic mission planning for multiple mobile robots.” In IEEE International Conference on Robotics and Automation (Vol. 3, pp. 2396-2401).

Burgard, W., Cremers, A. B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., and Thrun, S. (1998). “The interactive museum tour-guide robot.” In 15th national conference on Artificial Intelligence, Madison, Wisconsin, USA (pp. 11-18).

Burgard, W., Cremers, A. B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., and

Carlson, T., and Demiris, Y. (2012). Collaborative control for a robotic wheelchair: evaluation of performance, attention, and workload. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42*(3), 876-888.

Chen, Q., and Luh, J. Y. S. (1994), “Coordination and control of a group of small mobile robots.” In IEEE International Conference on Robotics and Automation (pp. 2315-2320).

Christensen, H. I., Batzinger, T., Bekris, K., Bohringer, K., Bordogna, J., Bradski, G., Brock, O., Burnstein, J., Fuhlbrigge, Y.,and Eastman, R. (2009). “A roadmap for us robotics: from internet to robotics.” *Computing Community Consortium*.

Christofides, N. (1976), “Worst-case analysis of a new heuristic for the travelling salesman problem.” (No. RR-388). Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group.

Coyle (2014), “Green Building 101: Why is energy efficiency important?”, < http://www.usgbc.org/articles/green-building-101-why-energy-efficiency-important> [Accessed, 12th April 2015].

COZIR, (2015). “GSS Sensor User’s Manual” < http://www.co2meters.com/Documentation/Manuals/Manual-GSS-Sensors.pdf> [Accessed 20th September, 2015].

Dermibas, M. (2005). “Wireless sensor networks for monitoring of large public buildings.” University at Buffalo, Tech. Rep.

Deruwe G, and Wall R. (2008) “Pedestrian navigation and integration with distributed smart signal traffic controls.” *IECON 2008: Proceedings of the 34th Annual Conference of IEEE - Industrial Electronics;* 2008 Nov 10-13; Orlando, USA. Orlando: IEEE; 2008. p. 2923-8. http://dx.doi.org/10.1109/iecon.2008.4758424.

Dias, M. B., and Stentz, A. (2001). “A market approach to multirobot coordination” (No. CMU-RI-TR-01-26). Carnegie-Mellon Univ Pittsburgh Pa Robotics Inst.

Dieter F., Burgard W., and Thrun S., (1998). "Active markov localization for mobile robots." *Robotics and Autonomous Systems* 25, no. 3 (1998): 195-207.

Dijkstra, E. W. (1959). "A note on two problems in connection with graphs". Numerische Mathematik. 1: 269–271. DOI:10.1007/BF01386390

Ding, H., and Castañón, D. (2016). “Fast algorithms for UAV tasking and routing.” In Control Applications (CCA), 2016 IEEE Conference on (pp. 368-373). IEEE.

DOE (2017), “Building Performance Database”, https://energy.gov/eere/buildings/building-performance-database [Accessed: April’16th, 2017]

Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., and Winfield, A. (Eds.). (2008). “Ant Colony Optimization and Swarm Intelligence” ANTS 6th International Conference , Brussels, Belgium, September 22-24, 2008, (Vol. 5217). Springer.

Evangelisti, L., Guattari, C. and Gori, P., (2015). “Energy retrofit strategies for residential building envelopes: An Italian case study of an early-50s building.” *Sustainability*, *7*(8), pp.10445-10460.

Fallah, N., Apostolopoulos, I., Bekris, K., and Folmer, E. (2013). Indoor human navigation systems: A survey. *Interacting with Computers*, *25*(1), 21-33.

Feng, C., and Kamat, V. R. (2012). “Augmented reality markers as spatial indices for indoor mobile AECFM applications.” *In Proceedings of 12th international conference on construction applications of virtual reality (CONVR 2012)* (pp. 235-24).

Feng, C., Azenkot, S., and Cakmak, M. (2015). “Designing a robot guide for blind people in indoor environments.” In Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts (pp. 107-108). ACM.

Fernández-Madrigal, J. A., Galindo, C., and González, J. (2004). “Assistive navigation of a robotic wheelchair using a multihierarchical model of the environment.” *Integrated Computer-Aided Engineering*, *11*(4), 309-322.

Flagg, J. (2009). Chapter 1: Wheeled Mobility Demographics. In S. Bauer, M. E. Buning (Eds) The Industry Profile on Wheeled Mobility, Rehabilitation Engineering Research Center on Technology Transfer, University at Buffalo, Buffalo, NY.

Florin, C., Akhil, S., Shubham, S., Vivek, J., and Rachit, J. "A data-driven approach for solving route & fleet optimization problems." < http://www.karvyanalytics.com/A-data-driven-approach-for-solving.pdf>, Karvy analytics, [Accessed on 11th Jan, 2017]

García, J. C., Amaral, P. F., Marrñn, M., Mazo, M., and Bastos Filho, T. F. (2010). “Proposal for an Ambient Assisted Wheelchair (A2W).” In *Industrial Electronics (ISIE), 2010 IEEE International Symposium on* (pp. 2325-2330). IEEE.

Gionata, C., Francesco, F., Alessandro, F., Sabrina, I., and Andrea, M. (2014). An inertial and

QR code landmarks-based navigation system for impaired wheelchair users. In *Ambient Assisted Living*(pp. 205-214). Springer, Cham.

Golden, B., Levy, L., and Dahl, R., "Two generalizations of the traveling salesman problem." Omega 1981;9(4):439–41.

Goldstein M. (1990), "Self-organizing feature maps for the multiple traveling salesmen problem. In: Proceedings of the IEEE international conference on neural network. San Diego, CA, 1990. p. 258–61.

Gorbenko, A., Mornev, M., and Popov, V. (2011), "Planning a typical working day for indoor service robots." IAENG International Journal of Computer Science, 38(3), 176-182.

Granderson, J. (2013). "Energy information handbook: Applications for energy-efficient building operations." *Environmental Energy Technologies Division, Presented at Lawrence Berkeley National Laboratory, Berkeley, CA* < http://escholarship.org/uc/item/03z8k1v3> [Accessed: Jan'16th, 2015]

Granzer, W., Praus, F., and Kastner, W. (2010). Security in building automation systems. Industrial Electronics, IEEE Transactions on, 57(11), 3622-3630.

Gromicho J, Paixão J, Branco I. (1992) "Exact solution of multiple traveling salesman problems." In: Mustafa Akgül, et al., editors. Combinatorial optimization. NATO ASI Series, vol. F82. Berlin: Springer; 1992. p. 291–92.

Grondzik, W. T. (2009). " Principles of Building Commissioning". John Wiley and Sons, Inc..

Gupta (2017) "Find paths given source destination" < http://www.geeksforgeeks.org/find-paths-given-source-destination/> [Accessed 16th June, 2017]

Habib, M. K. (2007). "Robot Mapping and Navigation by Fusing Sensory Information." INTECH Open Access Publisher.

Heo, Y., Choudhary, R., and Augenbroe, G. A. (2012). "Calibration of building energy models for retrofit analysis under uncertainty." *Energy and Buildings*, 47, 550-560.

Hoffman, K. L., Padberg, M., and Rinaldi, G. (2013). "Traveling salesman problem." In Encyclopedia of Operations Research and Management Science(pp. 1573-1578). Springer US.

Horan, B., Najdovski, Z., Black, T., Nahavandi, S., and Crothers, P. (2011). "OzTug mobile robot for manufacturing transportation." In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on* (pp. 3554-3560). IEEE.

IEA (2012), "ECBCS Annex 53: Total energy use in buildings – analysis and evaluation methods", 2009-2012.

iRobot (2016), "Clean less, Save more" < www.irobot.com> (May 16, 2016)

Iwasaki, M., and Fujinami, K. (2012). "Recognition of Pointing and Calling for Industrial Safety Management." *In Proc. of the 2012 First ICT International Senior Project Conference and IEEE Thailand Senior Project Contest* (pp. 50-53).

Jaggs, M., and Palmer, J. (2000). "Energy performance indoor environmental quality retrofit—a European diagnosis and decision making method for building refurbishment." *Energy and buildings, 31*(2), 97-101.

Jose, K., and Pratihar, D. K. (2016). "Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods." Robotics and Autonomous Systems, 80, 34-42.

Kara, I., and Bektas, T. (2006), "Integer linear programming formulations of multiple salesman problems and its variations." European Journal of Operational Research, 174(3), 1449-1458.

Karimi H., A., Zhang L., and Benner J., G., (2014) "Personalized accessibility map (PAM): a novel assisted wayfinding approach for people with disabilities", *Annals of GIS*, 20:2, 99-108, DOI: 10.1080/19475683.2014.904438

Karp, R. M. (1977). Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane. Mathematics of operations research, 2(3), 209-224.

Khoury, H. M., and Kamat, V. R. (2009). Evaluation of position tracking technologies for user localization in indoor construction environments. *Automation in Construction*, *18*(4), 444-457.

Kim, B. K., Tanaka, H., and Sumi, Y. (2015). Robotic wheelchair using a high accuracy visual marker lentibar and its application to door crossing navigation. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (pp. 4478-4483). IEEE.

Kim, J., and Lynch, J. P. (2011). "Autonomous decentralized system identification by Markov parameter estimation using distributed smart wireless sensor networks." *Journal of Engineering Mechanics*, 138(5), 478-490.

King DE, Matheson E, Chirina S, Shankar A, Broman-Fulks J. (2013). The Status of Baby Boomers' Health in the United States: The Healthiest Generation?. JAMA Intern Med. 2013;173(5):385-386. doi:10.1001/jamainternmed.2013.2006.

Király, A., and Abonyi, J. (2015). "Redesign of the supply of mobile mechanics based on a novel genetic optimization algorithm using Google Maps API." Engineering Applications of Artificial Intelligence, 38, 122-130.

Kouroupetroglou G. (2013) "Disability informatics and web accessibility for motor limitations." In: *Advances in medical technologies and clinical*. Hershey: IGI Global; 2013.

LaPlante, Mitchell P, and Disability Statistics Center. (2003). "Demographics of Wheeled

Mobility Device Users." In *Conference on Space Requirements for Wheeled Mobility*. Center for Inclusive Design and Environmental Access Buffalo, New York.

Laporte G. (1992), "The traveling salesman problem: an overview of exact and approximate algorithms." European Journal of Operational Research 1992;59:231–47.

LaValle, S. M., and Kuffner, J. J. (2001). "Randomized kinodynamic planning." The International Journal of Robotics Research, 20(5), 378-400.

Lawler, E L., Lenstra, J K, Rinnooy Kann AHG, and Shmoys DB. "The traveling salesman problem." Chichester: Wiley; 1985.

Lee, J., Hyun, C.H. and Park, M., (2013). "A vision-based automated guided vehicle system with marker recognition for indoor use." *Sensors*, *13*(8), pp.10052-10073.

Levitt, T. S., and Lawton, D. T. (1990). "Qualitative navigation for mobile robots." *Artificial intelligence*, *44*(3), 305-360.

Lin, H.W., (2006). "A VaFALCON neuro-fuzzy system for mining of incomplete construction databases". *Automation in construction*, *15*(1), pp.20-32.

Linner, T., Güttler, J., Bock, T., and Georgoulas, C., (2015). "Assistive robotic micro-rooms for independent living." *Automation in Construction*, 51 (2015) pp. 8–22.

Ma, Z., Cooper, P., Daly, D., and Ledo, L. (2012). "Existing building retrofits: Methodology and state-of-the-art." *Energy and Buildings*, 55, 889-902.

Mantha, B. R., Menassa, C. C., and Kamat, V. R. (2016). "Ambient Data Collection in Indoor Building Environments Using Mobile Robots." In ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction (Vol. 33, p. 1).Auburn, Alabama, USA.

Mantha, B., Feng, C., Meanassa, C., and Kamat, V, (2015a) "Real-time Building Energy And

Comfort Parameter Data Collection Using Mobile Indoor Robots”, *International Symposium on Automation and Robotics in Construction (ISARC) 2015*, Oulu, Finland.

Mantha, B., Meanassa, C., and Kamat, V, (2015b) “A Taxonomy of Data Types and Data Collection Methods for Building Energy Monitoring and Performance Simulation”, *Advances in Building energy research, Taylor and Francis*. Permlink < http://www.tandfonline.com/doi/full/10.1080/17512549.2015.1103665>

Matthews, H., L. Beale, P. Picton, and D. Briggs. 2003. “Modelling Access with GIS in Urban Systems (Magus): Capturing the Experiences of Wheelchair Users.” Area 35 (1): 34–45. doi:10.1111/1475-4762.00108.

Mirza R, Tehseen A, Kumar AVJ.(2012) “An indoor navigation approach to aid the physically disabled people.” *ICCEET 2012: Proceedings of the International Conference on Computing, Electronics and Electrical Technologies*; 2012 March 21-22; Nagercoil, India. Kumaracoil: IEEE; 2012. p. 979-83. http://dx.doi.org/10.1109/icceet.2012.6203860.

Modares A, Somhom S, and Enkawa T. (1999), “A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems.” International Transactions in Operational Research, Vol 6: PP 591–606.

Montañés, J. A. P., Rodríguez, A. M., and Prieto, I. S. (2013). “Smart indoor positioning/location and navigation: A lightweight approach.” *IJIMAI*, 2(2), 43-50.

Montesano, L., Díaz, M., Bhaskar, S., & Minguez, J. (2010). Towards an intelligent wheelchair system for users with cerebral palsy. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *18*(2), 193-202.

Morales, Y., Watanabe, A., Ferreri, F., Even, J., Ikeda, T., Shinozawa, K., ... & Hagita, N. (2015, May). Including human factors for planning comfortable paths. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*(pp. 6153-6159). IEEE.

Nallusamy, R., Duraiswamy, K., Dhanalaksmi, R., and Parthiban, P. (2009), "Optimization of Non-Linear Multiple Traveling Salesman Problem Using K-Means Clustering, Shrink Wrap Algorithm and Metaheuristics." International Journal of Nonlinear Science, 8(4), 480-487.

Neges, M., Koch, C., König, M., and Abramovici, M. (2017). "Combining visual natural markers and IMU for improved AR based indoor navigation." *Advanced Engineering Informatics*, *31*, 18-31.

NIST (2017), " All Simple Paths" , < https://xlinux.nist.gov/dads//HTML/allSimplePaths.html> [Accessed June 16th, 2017]

NIU (2016) "" < https://ori.hhs.gov/education/products/n_illinois_u/dfront.html> [Accessed April 25th, 2016]

Nourollah, S., V. (2009). "Automated building monitoring using a wireless sensor network." *Doctoral dissertation*, Concordia University.

Oliveira, L. C. D., Soares, A. B., Cardoso, A., Andrade, A. D. O., and Lamounier Júnior, E. A. (2016). Mobile Augmented Reality enhances indoor navigation for wheelchair users. *Research on Biomedical Engineering*, *32*(2), 111-122.

Olson, E., (2011). "AprilTag: A robust and flexible visual fiducial system." Proceedings of IEEE International Conference on Robotics and Automation, pp. 3400-3407.

ORNL (Oak Ridge National Laboratory) (2008), "Which kind of insulation is best?", http://web.ornl.gov/sci/buildings/docs/factSheets/Insulation-FactSheet-2008.pdf [Accessed 20th September, 2016]

Pandoricweb (2011) “Simple k-means clustering implementation”, http://pandoricweb.tumblr.com/post/8646701677/python-implementation-of-the-k-means-clustering , [Accessed September 10th, 2016]

Papadimitriou, C. H. (1977). “The Euclidean travelling salesman problem is NP-complete.” Theoretical Computer Science, 4(3), 237-244.

Park, T. J., and Hong, S. H. (2009). “Experimental case study of a bacnet-based lighting control system.” *Automation Science and Engineering, IEEE Transactions* on, 6(2), 322-333.

Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. (2013), “A review of dynamic vehicle routing problems.” European Journal of Operational Research, 225(1), 1-11.

Pinto, J. (2016). “Intelligent robots will be everywhere”, < http://www.automation.com> (May 16, 2016).

Ponraj, R., and Amalanathan, G. (2014), ”Optimizing Multiple Travelling Salesman problem considering the Road Capacity.” Journal of computer science, 10(4), 680.

Postolache O, Silva PG, Pinheiro E, Pereira MD, Madeira R, Mendes J, Cunha M, Postolache G, Moura CM. (2011) “Pervasive sensing and computing for wheelchairs users health assessment.” *Proceedings of the 1st Portuguese Meeting in Bioengineering;* 2011 Mar 1-4; Lisbon, Portugal. Lisbon: IEEE; 2011. p. 1-4. http://dx.doi.org/10.1109/enbeng.2011.6026070.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., and Ng, A. Y. (2009).” ROS: an open-source Robot Operating System.” In *ICRA workshop on open source software* (Vol. 3, No. 3.2, p. 5).

Raffler, S., Bichlmair, S., and Kilian, R. (2015). Mounting of sensors on surfaces in historic buildings. *Energy and Buildings*, *95*, 92-97.

Raftery, P., Keane, M., and Costa, A. (2011). “Calibrating whole building energy models: Detailed case study using hourly measured data.” *Energy and Buildings*, 43(12), 3666-3679.

Raghavan, A. N., Ananthapadmanaban, H., Sivamurugan, M. S., and Ravindran, B. (2010). “Accurate mobile robot localization in indoor environments using Bluetooth.” In *Robotics and Automation (ICRA), 2010 IEEE International Conference on* (pp. 4391-4396). IEEE.

Rao, J., and Biswas, S. (2008), “Joint routing and navigation protocols for data harvesting in sensor networks.” In 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (pp. 143-152)

Robotics, V. O. (2013) "A roadmap for US robotics: from internet to robotics." *Robotics Virtual Organization* (2013).

Röhrig, C., Kirsch, C., Lategahn, J., Müller, M., and Telle, L. (2012). “Localization of Autonomous Mobile Robots in a Cellular Transport System”.*Engineering Letters, 20*(2).

Ryan, J. L., Bailey, T. G., Moore, J. T., and Carlton, W. B. (1998), “Reactive tabu search in unmanned aerial reconnaissance simulations.” In Proceedings of the 30th conference on Winter simulation (pp. 873-880). IEEE Computer Society Press.

Sacks, R., Navon, R., Brodetskaia, I., and Shapira, A. (2005). “Feasibility of automated monitoring of lifting equipment in support of project control.” Journal of Construction Engineering and Management, 131(5), 604-614.

Sanchez JH, Aguayo FA, Hassler TM. (2007) “Independent outdoor mobility for the blind.” *Proceedings of the Virtual Rehabilitation 2007,* Sep 27-29; Venice, Italy. Venice: IEEE; 2007. p. 114-20. http://dx.doi.org/10.1109/icvr.2007.4362150.

Sathyan, A., Boone, N., and Cohen, K. (2015). "Comparison of Approximate Approaches to Solving the Travelling Salesman Problem and its Application to UAV Swarming." International Journal of Unmanned Systems Engineering., 3(1), 1.

Sauro J. (2011), "Measuring Usability With The System Usability Scale (SUS)" < https://measuringu.com/sus/> [Accessed 10th March, 2018]

SDR (2016), "Tactical Surveillance Robots" <http://www.sdrtactical.com/Surveillance-Robots/> (May 16, 2016).

Selim, H., and Zhan, J. (2016). "Towards shortest path identification on large networks." *Journal of Big Data*, *3*(1), 10.

Shneier M. and Bostelman R (2015). "Literature Review of Mobile Robots for Manufacturing", Intelligent Systems Division Engineering Laboratory, NIST < http://nvlpubs.nist.gov/nistpubs/ir/2015/NIST.IR.8022.pdf> [Accessed March 23rd, 2016].

Shohet, I. M., Puterman, M., and Gilboa, E. (2002). "Deterioration patterns of building cladding components for maintenance management." Constr. Manage. Econ., 20(4), 305–314.

Singh, S., and Lodhi, E. A. (2014). "Comparison Study of Multiple Traveling Salesmen Problem using Genetic Algorithm." International Journal of Computer Science and Network Security (IJCSNS), 14(7), 107.

Sofge, D., Schultz, A., and De Jong, K. (2002). "Evolutionary computational approaches to solving the multiple traveling salesman problem using a neighborhood attractor schema." In Workshops on Applications of Evolutionary Computation (pp. 153-162). Springer Berlin Heidelberg.

Song C., Lee, K., and Lee, W., D., "Extended simulated annealing for augmented TSP and multi-salesmen TSP." Proceedings of the international joint conference on neural networks, vol. 3, 2003. p. 2340–43.

Swartz, R. A., Zimmerman, A. T., Lynch, J. P., Rosario, J., Brady, T., Salvino, L., and Law, K. H. (2012). "Hybrid wireless hull monitoring system for naval combat vessels." Structure and *Infrastructure Engineering*, 8(7), 621-638.

T. Adlam, "Technology, autonomy and cognitive disability," in *Ubi- Health 2003: 2nd Int.Workshop Ubiquitous Comput. Pervasive Healthcare Appl.*, 2003.

Tekdas, O., Isler, V., Lim, J. H., and Terzis, A. (2009). "Using mobile robots to harvest data from sensor fields." *IEEE Wireless Communications*, 16(1), 22.

Teo, T. A., and Cho, K. H. (2016). "BIM-oriented indoor network model for indoor and outdoor combined route planning." *Advanced Engineering Informatics*, *30*(3), 268-282.

Thomas, A., Menassa, C. C., and Kamat, V. R. (2016). "System Dynamics Framework to Study the Effect of Material Performance on a Building's Lifecycle Energy Requirements." *Journal of Computing in Civil Engineering*, 04016034.

Thrun, S. (1998). The interactive museum tour-guide robot. *In Proc. of the National Conference on Artificial Intelligence (AAAI), Madison, Winsconsin, USA* (pp. 11-18).

Thrun, S., Bennewitz, M., Burgard, W., Cremers, A., Dellaert, F., Fox, D., Hähnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (1999). "MINERVA: A second generation mobile tour-guide robot." *In Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Detroit, Michigan, USA, pp. 3136-3141. 1999.

Thrun, S., Burgard, W., and Fox. D., Probabilistic Robotics. MIT Press, 2005. - -

Tilden M.W. "Robotics can - and will - change our lives in the near future", < www.theguardian.com> (May 16, 2016).

Torres-Solis, J., Falk, T. H., and Chau, T. (2010). "A review of indoor localization technologies: towards navigational assistance for topographical disorientation." *Ambient Intelligence*, pp. 51–84, 2010.

Tsetsos V, Anagnostopoulos C, Kikiras P, Hadjiefthymiades S. (2006) "Semantically enriched navigation for indoor environments." *International Journal of Web and Grid Services*. 2006; 2(4):453-78. http://dx.doi.org/10.1504/IJWGS.2006.011714.

U.S. DOE. (2015). "Reference buildings by building type: Large office."http://energy.gov/eere/downloads/reference-buildings-building-type-large-office-1 [Accessed 20th January, 2015].

UN (2004), "Accessibility for the Disabled - A Design Manual for a Barrier Free Environment" http://www.un.org/esa/socdev/enable/designm/AD3-01.htm [Accessed 18th September, 2017]

US DOE/EIA (2014), "Annual Energy Outlook 2014 with projections to 2040", <http://www.eia.gov/forecasts/aeo/pdf/0383(2014).pdf> [Accessed, 23rd March 2015]

Vilar, E., Rebelo, F., and Noriega, P. (2014). Indoor human wayfinding performance using vertical and horizontal signage in virtual reality. *Human Factors and Ergonomics in Manufacturing & Service Industries*, *24*(6), 601-615.

Vlissidis, A., Charakopoulos, S., Kolokotsa, D., and Boian, I. (2008). "Wireless Sensor Networks Building Monitoring Application." *Bulletin of the Transilvania University of Brasov*, Series I: Engineering Sciences, 1(50).

Wacholder E, Han J, and Mann RC (1989). "A neural network algorithm for the multiple traveling salesmen problem." Biology in Cybernetics Vol 61, PP 11–9.

Wang, F., Liu, J., and Sun, L. (2010b). "Ambient data collection with wireless sensor networks." EURASIP Journal on Wireless Communications and Networking, 2010, 4.

Wang, W., Wang, N., Jafer, E., Hayes, M., O'Flynn, B., and O'Mathuna, C. (2010a). "Autonomous wireless sensor network based building energy and environment monitoring system design." *In Environmental Science and Information Application Technology (ESIAT), 2010 International Conference on IEEE*, Vol. 3, pp. 367-372.

Wu, B. F., Jen, C. L., & Huang, T. W. (2011). "Intelligent radio based positioning and fuzzy based navigation for robotic wheelchair with wireless local area networks." In *Robot, Vision and Signal Processing (RVSP), 2011 First International Conference on* (pp. 61-64). IEEE.

Xu, M., Hijazi, I., Mebarki, A., Meouche, R. E., and Abune'meh, M. (2016). "Indoor guided evacuation: TIN for graph generation and crowd evacuation." Geomatics, Natural Hazards and Risk, 1-10.

Xu, W., and McCloskey, S. (2011) "2D Barcode localization and motion deblurring using a flutter shutter camera." In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on* (pp. 159-165). IEEE.

Yu, Z., Jinhai, L., Guochang, G., Rubo, Z., and Haiyan, Y. (2002), "An implementation of evolutionary computation for path planning of cooperative mobile robots." In Proceedings of the 4th World Congress on Intelligent Control and Automation, (Vol. 3, pp. 1798-1802).

Zhang T, GruverWA, and Smith MH (1999). "Team scheduling by genetic search." Proceedings of the second international conference on intelligent processing and manufacturing of materials, vol. 2, 1999. p. 839–44.

Zhi, G. S., Lo, S. M., and Fang, Z. (2003). "A graph-based algorithm for extracting units and loops from architectural floor plans for a building evacuation model." *Computer-Aided Design*, *35*(1), 1-14.