



Accessibility Assessment Via Workspace Estimation

Jing Yang

Technical Report CSE-2008-01

January 10, 2008

Department of Computer Science and Engineering
4700 Keele Street Toronto, Ontario M3J 1P3 Canada

Abstract

Accessibility is an important factor to enable people with disabilities to live a good and independent life. For an environment to be suited for wheelchair use it must be sufficiently clear of obstacles so that the wheelchair can navigate it. The process of evaluating a built environment for accessibility is known as “accessibility assessment”. Determining accessibility is closely related to the problem of determining possible motions of a specific kinematic structure – given an environment and a mobile device, how much of the environment is accessible? Given these similarities, the accessibility assessment process is reformulated as a kinematic planning problem and a motion planner based on the Probabilistic Roadmap Planner (PRM) is used to assess accessibility. Rather than treating each of the joint angles ‘equally’ within the PRM, we explore a hierarchical characteristic of all the joint angles when constructing the roadmap. This PRM variant allows efficient and effective planning in the accessibility assessment domain. An additional component of the methodology for accessibility assessment, an environmental planner, evaluates potential environmental modifications by invoking the motion planner. The developed system can be a useful tool for clinicians to assess accessibility.

Contents

Abstract	iv
1 Introduction	2
1.1 From accessibility assessment to robot workspace estimation	4
1.2 Thesis contributions	6
1.3 Structure of the thesis	6
2 Related Work	7
2.1 Accessibility assessment	7
2.1.1 Traditional accessibility assessment	8
2.1.2 Advanced tools to support accessibility assessment	10
2.2 Motion Planning	23
2.2.1 Definition and application	23
2.2.2 Traditional motion planning approaches	25
2.2.3 Randomized planning algorithm	28
2.3 Summary	33
3 Accessibility Assessment	35
3.1 Formal statement of the problem	35
3.1.1 Environment	36
3.1.2 Reachable workspace	36
3.1.3 Accessibility assessment	37
3.2 Motion planner: hierarchical PRM	38
3.2.1 Observations	38
3.2.2 Hierarchical probabilistic roadmaps	41
3.2.3 Construction of the hierarchical roadmap \mathcal{R}	45
3.2.4 Roadmap Enhancement	49
3.2.5 Example	50
3.3 Environmental planner	53
3.3.1 A greedy algorithm	53
3.3.2 Example	57
3.4 Summary	60
4 Assessing accessibility	63
4.1 Experimental setup	63
4.1.1 Kinematic model	63
4.1.2 Environment	63

4.1.3	Implementation details	67
4.2	Reachable workspace estimation	68
4.3	Environmental modifications	71
4.4	Summary	74
5	Discussion and future work	75
5.1	Discussion	75
5.2	Future work	76
5.2.1	Extending the motion planner	76
5.2.2	Extending the environmental planner	77
5.2.3	Extending the accessibility assessment research	77

Chapter 1

Introduction

“I like that word, ‘accessibility.’ It should be a real part in the life of every paral. The paral, like anyone else, should feel that all people and places, thoughts and actions, arts and sciences, are open to him.¹”

An estimated 100-150 million people with physical disabilities worldwide require the use of a wheelchair [17]. From 1997 to 2002 in the U.S., the number of people age 15 and over who use a wheelchair increased from 2.2 million to 2.7 million and the number of those who used an ambulatory aid such as a cane, crutches, or a walker increased from 6.4 million to 9.1 million [44, 64]. The causes of disability among mobility support device users are various (see Figure 1.1), with older adults being the largest population group requiring mobility support devices. In Canada, about 7.2 percent of the population 85 years and older use a wheelchair and 31.7 percent use some other type of mobility device [61]. Given this large social need, there is a long history of research on wheelchair-related problems from a range of different perspectives including Health and Rehabilitation, Law, Civil Engineering and Computer Science and Engineering. Research into wheelchair-related problems include developing legislation for equal rights for wheelchair users, the search for lighter materials for wheelchair production, the development of robotic wheelchairs, and the design of wheelchair accessible spaces.

Accessibility is an important factor for all people with disabilities in order for them to live and work independently and to minimize the cost of personal care services. For an environment to be well suited for wheelchair use not only must the ground plane be flat and approximately horizontal, it must also be the case that the space be sufficiently clear of obstacles so that the wheelchair can navigate the environment. The process of evaluating an environment for accessibility is known as “accessibility assessment” and the task is sketched in Figure 1.2. Typically this process is performed manually, resulting in an error-prone and time consuming task that must be accomplished by trained assessors. Although manual assessment can be successful, the shortage of trained assessors, especially in rural areas, can introduce significant delays in assessing specific environments. Even when assessors are available, the lack of advanced tools can introduce delays in the process of assessment, and at the very least introduces a level of subjectivity in the process that is undesirable.

¹The desire for greater access can be traced back to 1934 in an article that appeared in the Polio Chronicle, that is the House Journal of Roosevelt Warm Springs Institute for Rehabilitation [68].

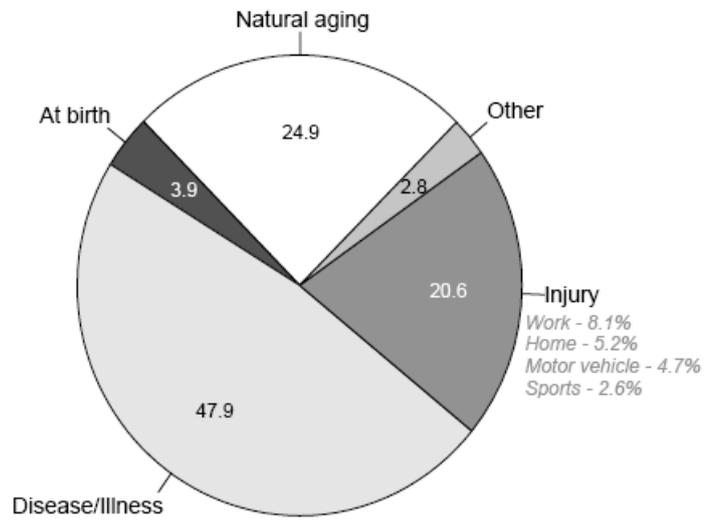


Figure 1.1: Causes of disability among mobility support devices. Data source: 2000/01 Canadian Community Health Survey. Reprinted from [61].

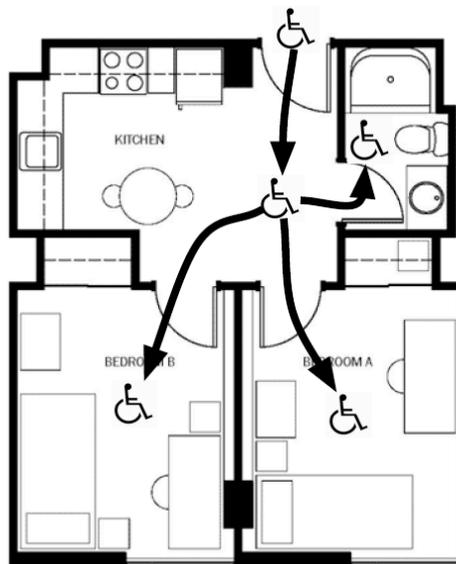


Figure 1.2: A floor plan of a student residence at York University with wheelchair accessible regions. Given an environment and a device with limited mobility (e.g. a wheelchair) is it possible to access all of the important portions of the environment?

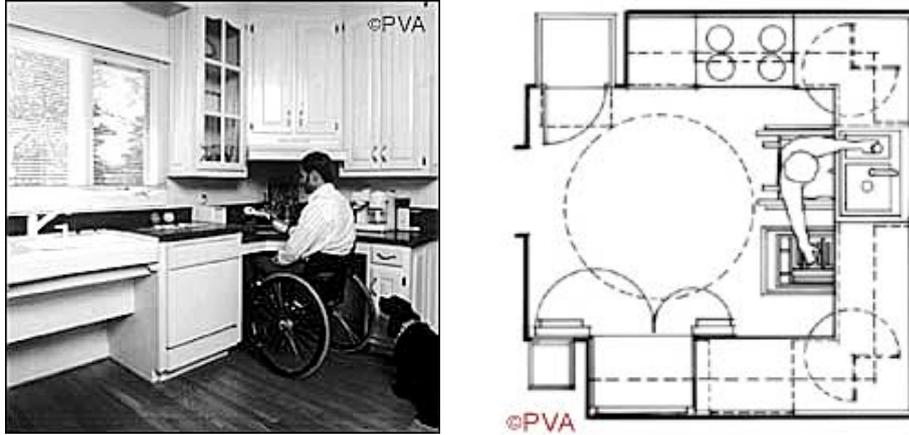


Figure 1.3: An accessible kitchen should take certain elements (e.g. the location of the water faucet) into account. Motion of the wheelchair user’s arm needs to be estimated in addition to the motion of the wheelchair itself. It is much more complicated than a 2D problem. Image reprinted from the Home Improvement Library (<http://www.BobVila.com>).

1.1 From accessibility assessment to robot workspace estimation

The problem of determining accessibility is closely related to the problem of determining possible motions of a specific kinematic structure – given an environment and a mobile device, how much of the environment is accessible? Given these similarities, by reformulating the accessibility assessment process in terms of this kinematic planning problem it is possible to leverage results from the robotic path planning literature to assist in accessibility assessment. Although a wheelchair can be very “robotic” or it can be of the more “manual” variety, the wheeled device introduces kinematic constraints on the device’s motion (e.g. it can move straight forward or straight backward, turn left or right, but it cannot move sideways) as a result of its mechanical structure. Consequently it can be much more difficult for wheelchair users to traverse an environment than it is for a person without disability. A small obstacle in a hallway might block the route of a wheelchair but not that of an able-bodied person. These constraints become even more subtle and complex when one considers the reachability of a person constrained to a wheelchair relative to the reachability of an able-bodied person. The problem goes from a 2D estimation problem on the plane (Figure 1.2) to a problem in 3D space and an even higher dimensional problem when considered in joint or actuator space (Figure 1.3).

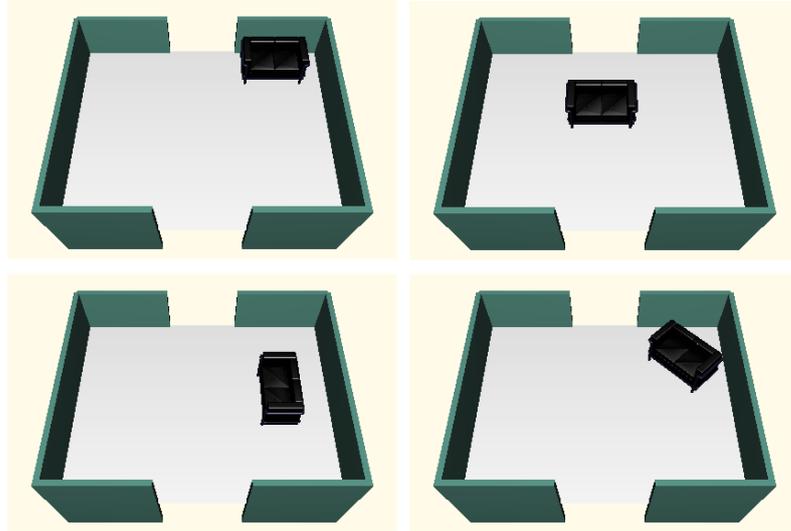


Figure 1.4: Furniture (here a sofa) may be placed in one of a number of possible positions. Which placement provides wheelchair users with the best accessibility can be difficult to determine.

The traditional accessibility standard requires an environment to provide equal access to wheelchair users as to persons who do not use such devices [18]. Although the definition of “equivalent” access is somewhat ambiguous the intent is to provide wheelchair users with as many accessible facilities as possible. In practice, providing accessibility may require modifications to the environment (see Figure 1.4). Such modifications may include structural changes (e.g. widening doorways) or adjusting the locations of objects (e.g. moving furniture).

Determining where and how to make such modifications is a difficult task as minor modifications may result in significant changes in the portion of the environment that is accessible (imagine the effect of opening a door), yet other changes may have limited or no effect on accessibility. Thus from a planning point of view, accessibility assessment can be formulated as two separate sub-problems: (i) Given a specific environment and an appropriate kinematic model of the user and wheelchair, determine those portions of the environment that are reachable. This is the path planning problem in mobile robotics; and, (ii) Given a set of possible changes to the environment, how do these changes influence the reachable portion of the environment? This is the problem of evaluating the impact of environmental changes on a robot’s reachable workspace.

1.2 Contributions

Accessibility assessment is a very complex task. Existing approaches require the detailed involvement of skilled clinicians and people with disabilities. This report develops methods that are designed to automatically evaluate the accessibility of physical environments of wheelchair users, or others with limited mobility, and suggest effective modifications for better accessibility. This report investigates algorithms that aid in the accessibility of an environment by first formally defining accessibility assessment in terms of robot workspace estimation and path planning and then adapting techniques in the mobile robot path planning literature to estimate the region of space that is physically accessible to a wheelchair or other kinematically restricted user. We use the estimated reachable workspace, points reachable by the end-effector of the user or robot, to evaluate a wheelchair user's accessibility within the environment. Based on the probabilistic roadmap (PRM) planner a hierarchical PRM method for efficient reachable workspace estimation is developed. Enhancements are developed to the basic PRM planner to enable efficient re-computation of reachability for modified environments. The resulting planner can be a useful tool for clinicians to assess accessibility by modifying certain objects. The system can also be used to find near-optimal solutions to maximize accessibility through potential environment modifications.

1.3 Structure of the report

The remainder of this report is organized as follows. Chapter 2 reviews existing methods for accessibility assessment and the supporting technology. It provides an overview of relevant robot motion planning techniques, as motion planning is a key element in this research. In Chapter 3, the problem of accessibility assessment is formalized in terms of robot workspace estimation and path planning, followed by the development of a general solution to the accessibility assessment problem using a variant of the PRM. Chapter 4 evaluates the algorithm on an implementation on the robotic wheelchair 'PlayBot' in a realistic 3D environment. Finally Chapter 5 summarizes the work and provides possible directions for future research.

Chapter 2

Related Work

This chapter reviews existing algorithms, technology and methods for accessibility assessment from two main perspectives: rehabilitation science and robotics. The first section introduces the problem of accessibility assessment and reviews traditional assessing methods and technologies that can be used to support accessibility assessment. The second section reviews robot motion planners that could be used to support accessibility assessment, including the topic of heuristic planners, the Probabilistic Roadmap Method (PRM) and its variants in both static and modifiable environments.

2.1 Accessibility assessment

Accessibility is about providing equal access to everyone. It is often used to describe buildings or facilities for people with disabilities, as in “this building is wheelchair accessible”. In this report, the term “accessibility” is used to refer to the degree to which portions of a built environment are reachable by people with limits to their mobility. Here the focus is on people who use a wheelchair as their primary means of locomotion. There are also people who use crutches, walkers, or canes (see Figure 2.1), but usually such people can travel wherever wheelchair users can [31], and in any event the technologies developed here can be easily adopted from wheelchairs to other kinematic models.

Lack of basic accessibility to public buildings is widely understood all over the world. Many laws enforce accessibility in various countries. For example, in the United States, under the *Americans with Disabilities Act (ADA)* [48], new public and private business construction must be accessible. Existing private businesses are required to increase the accessibility of their facilities when making other renovations in proportion to the cost of the other renovations. Similar legal requirements exist in Australia [46], the UK [51], and South Africa [50]. In Ontario, the *Ontarians with Disabilities Act* [49] requires that the Government of Ontario develop barrier-free design guidelines to promote accessibility for persons with disabilities to government buildings, structures and premises. The level of accessibility for persons with disabilities should be equal to or exceed the level of accessibility required by the *Building Code Act, 1992* and the regulations made under it.

In addition to these general accessibility requirements, special requirements are also made for houses where people with disabilities reside. For example, in the United States the *Fair Housing Act* [47] describes seven basic design and construction requirements that must be met for wheelchair accessible housing:

1. An accessible building entrance on an accessible route.
2. Accessible common and public use areas.



Figure 2.1: Students attempting to identify architectural barriers on campus by using mobility aid devices (wheelchair, walker, crutches). Photos courtesy of the Occupational Therapy Assistant program in Green River Community College (<http://www.instruction.greenriver.edu/OTA/photoot.htm>).

3. Usable doors (usable by a person in a wheelchair).
4. Existence of accessible route into and through the dwelling unit.
5. Light switches, electrical outlets, thermostats and other environmental controls in accessible locations.
6. Reinforced walls in bathrooms for later installation of grab bars.
7. Usable kitchens and bathrooms.

In order to fulfill these accessibility requirements an analysis of built houses and recommendation for home modifications to enhance accessibility are required. This process is known as “accessibility assessment”.

2.1.1 Traditional accessibility assessment

The normal practice for assessing accessibility is via a manual prescriptive code-based approach [18]. The evaluation of a specific environmental design follows parameters specified in the relevant official guidelines. For example, the *ADA Accessibility Guidelines* [8] contain ‘prescriptive’ specifications for clear doorway width, lavatory clearance, and the like (see Figure 2.2). Trained clinicians assess a building by checking if these specifications are met. This approach can be successful and is very straightforward but it has a number of limitations. First, it requires professional accessibility assessors to visit the buildings that need to be assessed. Providing services in rural or remote areas can require

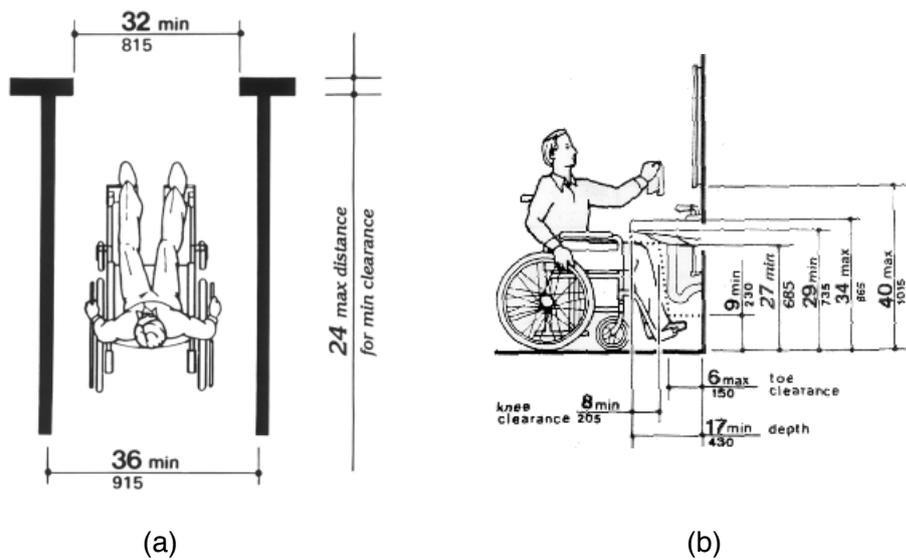


Figure 2.2: Samples of prescriptive code of the Americans with Disabilities Act Accessibility Guide [8]. (a) The minimum clear passage width for a single wheelchair shall be 36 inches (915 mm) minimum along an accessible route, but may be reduced to 32 inches (815 mm) minimum at a point for a maximum depth of 24 inches (610 mm), such as at a doorway; (b) The knee clearance is required underneath the lavatory: 27 inches (685 mm) minimum from the floor to the underside of the lavatory which extends 8 inches (205 mm) minimum measured from the front edge underneath the lavatory back towards the wall; if a minimum 9 inches (230 mm) of toe clearance is provided, a maximum of 6 inches (150 mm) of the 48 inches (1220 mm) of clear floor space required at the fixture may extend into the toe space.



Figure 2.3: Photo of President Clinton and iBOT’s developer Dean Kamen in the White House. iBOT is a wheelchair that enables the chair to balance and run on only two of its four wheels on some surfaces. From <http://www.technology.gov/Medal/2000/>.

extended travel time. This can lead to unaffordable expenses on the part of individuals who need the service. Second, the prescriptive assessment document cannot address all possible building design configurations or wheelchair use patterns. The gross structure of different wheelchairs may be similar, but the details including their motor constraints and kinematic structures can vary considerably. As technology develops new types of wheelchairs are being introduced including examples such as the iBOT shown in Figure 2.3. Providing a standard guideline for all kinds of wheelchairs and buildings is almost impossible. Third, even for a wheelchair whose structure is known, its performance against an environment is hard to predict exactly especially when we consider a person sitting in the wheelchair who can move their arms. A design configuration that is code-compliant does not necessarily imply real usability, and a design that does not satisfy the prescriptive accessibility requirement might actually be accessible by wheelchair users [18, 54].

2.1.2 Advanced tools to support accessibility assessment

Given the needs of accessibility assessment, advanced tools to support and extend the traditional accessibility assessment are being developed. A fundamental problem in accessibility assessment is the need to be able to reason or manipulate a model of the environment. We begin by considering how such models can be obtained, and then discuss the techniques that can be used to assist the reasoning process. The goal of these tools is to enable clin-



Figure 2.4: Five examples of photographs of an office taken from different positions. Reprinted from [31].

icians to provide the same or more accurate assessment of a given environment within a shorter time either on site or from a remote location.

Environmental acquisition and simulation

Many tools for accessibility assessment require the capture of digital 3D models of the environment to be assessed. It may be that such models exist already as blueprints. More generally it is necessary to capture the model for an existing environment. 3D acquisition and modeling has been intensively studied in the past few decades, and this has led to the development of a number of systems that are currently being used to assess accessibility or which could be easily modified to do so. For example, the Virtual Reality Telerehabilitation System (VRTS) [31] integrates telerehabilitation and virtual reality technologies for accessibility assessment. This pilot project in accessibility assessment uses environmental acquisition and modeling techniques to assist accessibility assessment. The system relies on the Photomodeler Pro [25] to construct 3D virtual models from a collection of 2D photographs of an interior environment (an example can be seen in Figures 2.4 and 2.5) and customized algorithms to analyze the 3D models for wheelchair accessibility.

The image-based acquisition technique used by the VRTS has the advantages of low cost and that the process of collecting the necessary photos to build the 3D model does not require advanced technical skills. The system does, however, require the user to manually establish correspondences between features in different images to estimate their 3d coordinates which can be a time consuming process. It also generates models composed of simple objects with polygonal faces and the quality of the model relies on the camera resolution.

Laser and stereo-based scanners are an alternative to the multiple monocular image-

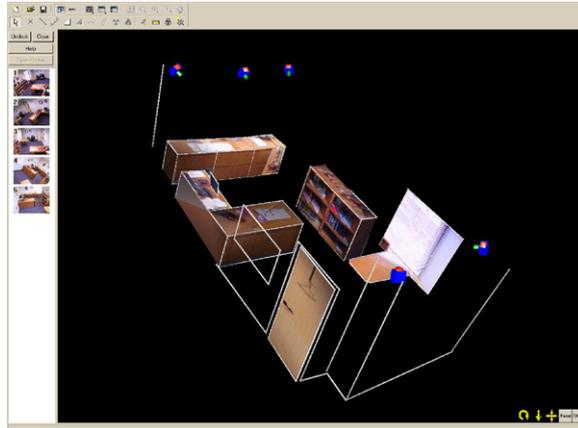


Figure 2.5: 3D model of an office produced from the photographs in Figure 2.4 by the software Photomodeler. The 3D model can be rotated which provide specialists better views of the scene. Reprinted from [31].



Figure 2.6: A sample 3D crime scene model. Data within the crime scene may be collected over a wide geographical area by many different people. The tool for data acquisition is a vision-based stereo reconstruction system using both MDA's iSM [60] and the AQUASensor [56]. Reprinted from [36].

based approach. Hand-held devices such as the instant Scene Modeler (iSM) [60] and the AQUASensor [20] automatically deal with issues related to depth acquisition, view registration and model construction. The user points the camera at a scene of interest to record images and the system creates a photo-realistic 3D calibrated model automatically. In practice, the iSM has been used to capture photo-realistic 3D models of crime scenes (see Figure 2.6) [36], and the AQUASensor has been used to generate accurate 3D models of underwater areas [20]. Many other 3D digitizing devices and systems exist, and a review of developments in this field for the past 20 years can be found in [7].

Once a 3D environmental model has been constructed it can be used to model the accessibility/usability of the environment through either simulation of the entire population or the motion of a single individual. Many computer-aided systems for environmental design and assessment have been developed based on these approaches (e.g. [72, 19, 18, 54]). By producing visible results of users' behavior the simulation can assist evaluation and the comparison of design alternatives, and this can help designers gain a better understanding of the interrelationship between the environment and its users.

Modeling groups of users One approach is to simulate human crowds and to predict group's behavior under different conditions such as pedestrian traffic simulation [19] and fire egress simulation [63]. The aim here is to evaluate the amount of space people need to conduct certain activities, such as the width of walkways, corridors, and doors. For instance, [19] describes a generalized force model of pedestrian behavior that investigates the mechanisms of panic and jamming by uncoordinated motion in crowds (Figure 2.7(a)).

Another simulation system that includes both a building model and a virtual human model is presented in [72]. The building model used here represents both geometric information and usability properties of design elements, and is generated automatically from a standard CAD model of the environment. Virtual humans are modeled as autonomous agents based on a large corpus of real-world behavior data collected through an automated video tracking system. The human model emulates the appearance, perception, social traits and physical behavior of real people. Motion is determined by scripting a series of basic motions (walking, standing, and sitting, etc. as shown in Figure 2.7(b)) that are encapsulated in advance. By inserting the virtual human models in the building model and then letting them "explore" it on their own volition, the system reveals the usability of the environment to its users. The environment can then be modified to see how different arrangements affect user behaviors.

Modeling an individual user The studies described above focus on the group behavior of humans since individual human behavior can be difficult to simulate [19]. However, when considering a kinematic structure with mobility limits (e.g. a wheelchair or a walker), such simulation may be possible. HabiTest [52] is an interactive living environment model,

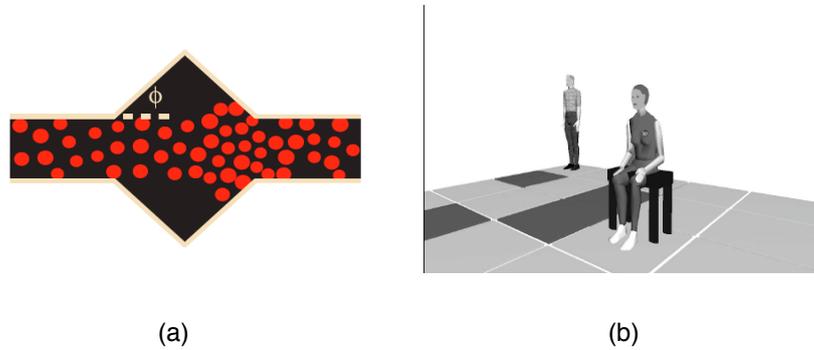


Figure 2.7: Example simulation systems for environmental analysis. (a) Snapshot of a pedestrian simulation of an escape route with a wider area. The corridor is 3 m wide and 15 m long, the length of the triangular pieces in the middle being $2 \times 3m = 6m$. Pedestrians enter the simulation area on the left-hand side and flee towards the right-hand side. Experiments show that the efficiency of escaping drops significantly if the corridor contains a widening. Reprinted from [19]. (b) Virtual humans demonstrating sitting and standing behaviors respectively. Reprinted from [72].



Figure 2.8: Screen shots of a HabiTest environment. Various views (egocentric view, bird's eye view and side view) of the scene within which the wheelchair navigates. Reprinted from [52].

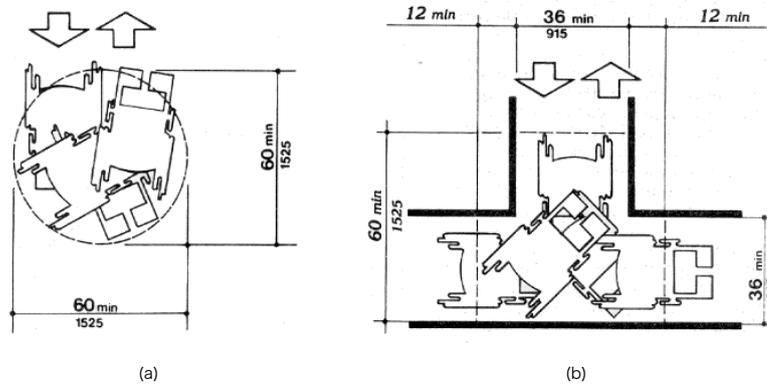


Figure 2.9: Wheelchairs require extra turning space for 180 degree turns. (a) 60-in (1525 mm) diameter space; (b) T-shaped space. Reprinted from [8].

that facilitates the planning, design and assessment of home and work settings for people with physical disabilities. HabiTest is implemented as an immersive virtual reality system and responds to user-driven manipulations such as navigation within the environment and alternation of its design (see Figure 2.8). The user drives the wheelchair in the virtual environment using a joystick. The joystick provides vibratory collision feedback to the user. The system also implements a virtual hand which can be controlled by the user. Control is achieved by putting the hand at a fixed distance in front of the user's eyes first and then using the joystick to move it to some offset from the center. HabiTest allows users to identify barriers that block their ability to navigate.

HabiTest enables finding a fit between the individual and the environmental setting through use of Virtual Reality and simulation technology. It can also be a useful tool to train people with disabilities to drive a wheelchair in the virtual environment. However, as it requires the users' step-by-step operation to navigate, its efficiency becomes a concern.

Modeling the kinematic structure of mobility assists

In order to model the accessibility of a given mobility assist, it is necessary to develop a mathematical model of the device's reachability. A wheelchair is an example of a kinematic structure, and as such can be treated mathematically using standard kinematic techniques. Following [37], let \mathcal{A} denote the mathematical model of the desired kinematic structure. A configuration of \mathcal{A} is a specification of the position of every point in \mathcal{A} relative to the global Cartesian frame \mathcal{W} . This configuration is usually described as a vector of parameters. For instance, a configuration of a wheeled vehicle moving in \mathbb{R}^2 might be described using three parameters x , y , and θ . A configuration of a kinematic chain can be described using the angles of the joints between the consecutive links j_1, j_2, \dots, j_n . The minimal number of



Figure 2.10: Examples of various wheelchair drive mechanisms. (a) A typical manual wheelchair with large rear wheels and front casters; (b) A powerchair with powered rear wheels and front casters; (c) Playbot consists of a wheelchair with electric powered front wheels and rear casters; (d) A tricycle drive scooter steered and powered by the single front wheel; (e) A tricycle drive scooter steered by the dual front wheels and powered by the rear wheels; (f) A large car-like scooter steered by front wheels and powered by rear wheels. Pictures (a-b, d-f) courtesy of ActiveLite Mobility Systems Inc. (www.activelite.com).

parameters needed to uniquely describe a robot configuration is known as the number of degrees of freedom (DOFs) of \mathcal{A} .

A typical wheelchair can move forwards, backwards, and in general follows curved trajectories in order to turn. Besides this, the possible movements with regard to a particular position depend on the current orientation of the vehicle [34]. As a result, wheelchairs require significant clearance space to navigate (see Figure 2.9). This constraint on the possible trajectories of a wheelchair is an example of a nonholonomic constraint [16], which is a common constraint for wheeled vehicles. Nonholonomic constraints are motion constraints that cannot be reduced [16].

The control and simulation of a wheelchair requires the development of a kinematic model of the device. Although a wheelchair can be propelled either manually (by pushing the wheels with the hands) or via various automated systems (see Figure 2.10), it can be modeled as a wheeled mobile robot (WMR). The kinematics are different for WMRs

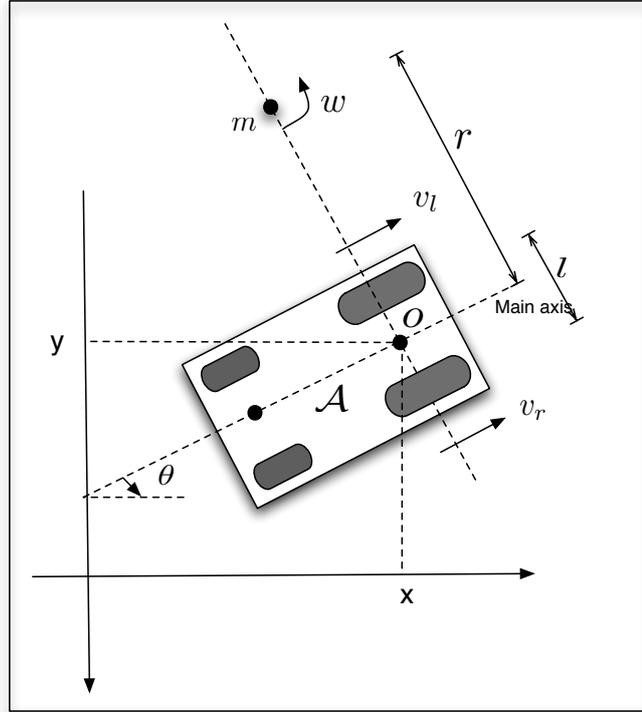


Figure 2.11: Differential drive kinematics. The vehicle is powered by two wheels centered at point o , and uses two castor wheels for stability. By providing independent velocity control to the two driven wheels, the vehicle follows different motions on the plane.

with different wheel and drive arrangements. Example WMR kinematic systems include differential, synchronous, tricycle and car drive [37, 15, 16, 66, 39].

Differential drive is perhaps the simplest drive mechanism and corresponds to the kinematic structure of many wheelchair designs (see Figure 2.11). Let $o = (x, y)$ be the position of the center of the powered wheels and θ be the angle that the vehicle's main axis makes with the positive x -axis. The powered wheels have controllable ground velocities v_l and v_r , and by varying these the vehicle moves along different trajectories. Suppose v_l and v_r are constant, then there are two basic motions: a translational motion and a rotational motion. *Straight motion.* When $v_l = v_r$ during the period $t \rightarrow t + \delta t$, the vehicle moves in a straight line path along its main axis with velocity $v = v_l = v_r$. The configuration at time $t + \delta t$ can be determined as:

$$\begin{aligned}
 x' &= x + v \cdot \delta t \cdot \cos(\theta) \\
 y' &= y + v \cdot \delta t \cdot \sin(\theta) \\
 \theta' &= \theta
 \end{aligned}
 \tag{2.1}$$

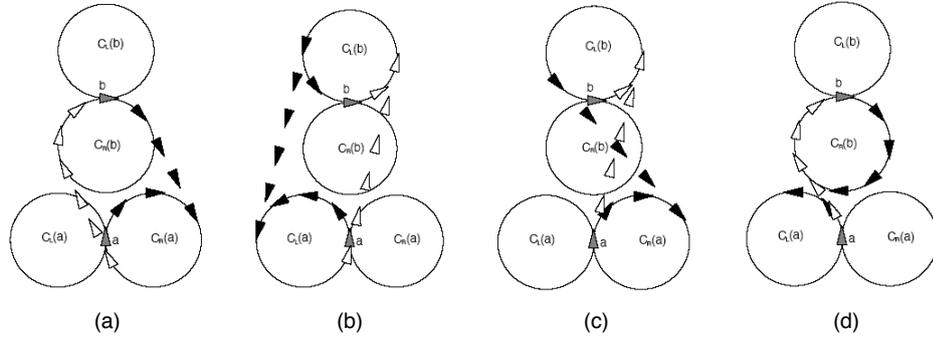


Figure 2.12: There can be eight possible Arc-Line-Arc (ALA) paths for a mobile wheeled robot from configuration a to b . For a given configuration $c = (x, y, \theta)$, the left circle $C_L(c)$ and right circle $C_R(c)$ are the two circles of minimum turning radius which pass through (x, y) with angle θ . The resulting ALA path varies depending on which circle is chosen. Picture derived from [65].

Circular motion. The vehicle moves in an arc path when $v_l \neq v_r$. The center of rotation $m = (x_m, y_m)$ lies on the common axis of the two driven wheels. The turning radius r is the distance between m and o , and w is the angular rate of the vehicle around m . These are determined by v_l and v_r :

$$\begin{aligned} r &= \frac{v_l + v_r}{2(v_r - v_l)} \\ w &= \frac{v_r - v_l}{l} \end{aligned} \quad (2.2)$$

where l is the distance between the two driven wheels. When $v_l < v_r$, the vehicle turns to its left; when $v_l > v_r$, it turns to its right. Note that when $v_l = -v_r$, the turning radius is zero and the vehicle rotates about the point o , which is called extreme rotation. Under extreme rotation the vehicle rotates in the smallest turning space.

Although other motion strategies are possible [58, 39], these two basic motions can be combined to achieve various paths from one configuration to another. For example, in [65] an Arc-Line-Arc (ALA) path from configuration a to b is defined as a concatenation of an arc path, a straight line path and another arc path. Figure 2.12 shows eight possible ALA paths.

Other than differential drive (e.g. Figure 2.10(a-c)), tricycle drive (e.g. Figure 2.10(d, e)) and car drive (e.g. Figure 2.10(f)) can be used to model in the design of wheelchair mechanisms. Wheelchairs that operate under tricycle or car drive are often designed to be used in outdoor environments. These devices do not use caster wheels as differential drive systems do. Instead they use fixed wheels for stability, which limits the vehicle from certain motions (e.g. the extreme rotation that is available under differential drive). The

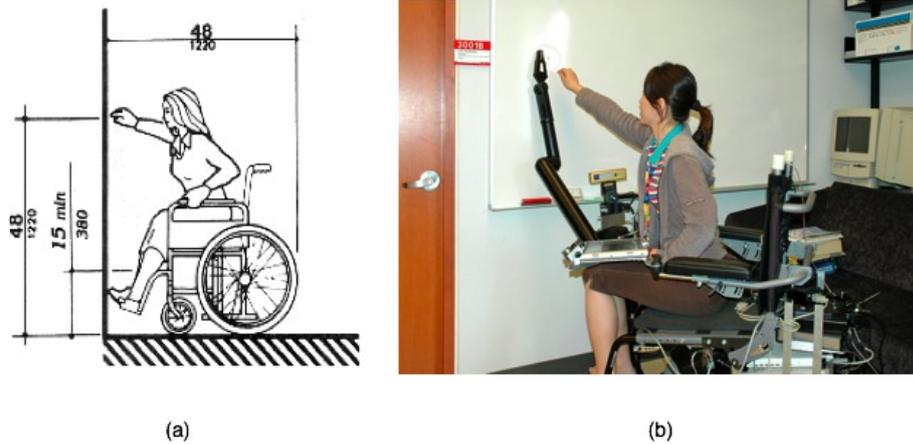


Figure 2.13: People sitting in a wheelchair have forward reach limit. (a) Prescriptive figure in ADAAG [8] specifies the forward reach range to be 48 inches maximum and 15 inches minimum. (b) The attached kinematic chain shows similar forward reach range with a person sitting in the wheelchair.

kinematics for the tricycle and car drives can be more complex than different drive (refer to [16, 65] for a complete description).

In addition to the constraints introduced by the wheeled vehicle itself, most wheelchair users are unable to stand and instead remain constrained to their wheelchairs during navigation within the environment. For example, the maximum height that the hands of a person in a wheelchair can reach is lower than that for people standing on their feet (Figure 2.13(a)). Assuming that the wheelchair user is confined to their wheelchair, then there is a kinematic link between the wheelchair base (whose kinematics is described above) and the reach of the user. A kinematic chain can be used to model the reach of the user seated in the wheelchair (Figure 2.13(b)). If we assume that the user is rigidly connected to their chair, and that the various links and joints can be modelled as a collection of rigid links connected via joints, then the position and orientation of the user's hand can be modelled using traditional techniques from the robot manipulator literature. Given the kinematic chain's joint variables values, the position of the links and hand (the end-effector) can be easily determined [30, 37] (see Figure 2.14).

The traditional approach to modelling kinematic chains is to establish a sequence of geometric frames throughout the chain, and to use simple geometric constructs to model the rigid relationship between these frames [43, 30]. Consider two consecutive link frames F_{i-1} and F_i . F_i can be determined uniquely from F_{i-1} by a rigid transformation that is a function of the adjustable joint. Many different formalisms are possible for representing this transformation. One common representation is through the use of Denavit-Hartenberg

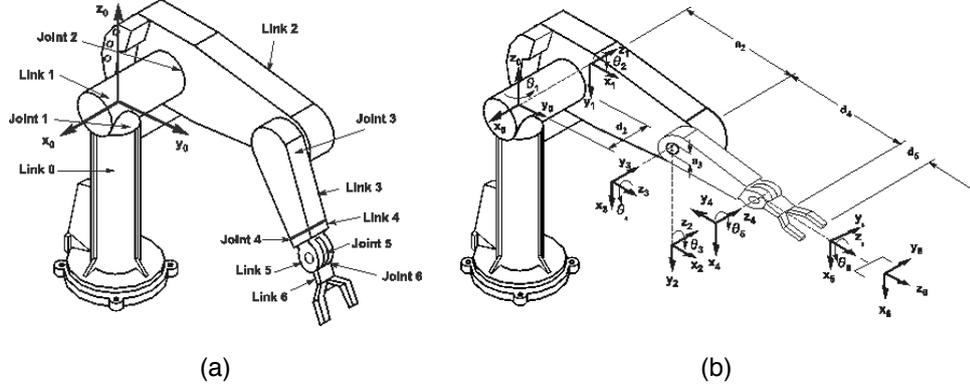


Figure 2.14: (a) PUMA industrial robot with six revolute joints. (b) Frame assignment of each link for PUMA robot. In a link frame F_i , z_i is always on a joint axis (except the end-effector frame which does not have a joint axis); x_i lies along the common perpendicular to axes z_{i-1} and z_i and is oriented from z_{i-1} to z_i . Picture derived from [43].

parameters [43, 30] (DH-Parameters are illustrated in Figure 2.15). Under the DH formalism the transformation between two links is formed from four basic transformations – a translation along z_{i-1} of distance d_i , a rotation around z_{i-1} of angle θ_i , a translation along x_i of distance a_i and a rotation around x_i of angle α_i (see Figure 2.14):

$${}^{i-1}T_i = \text{Trans}(z, d_i) \cdot \text{Rot}(z, \theta_i) \cdot \text{Trans}(x, a_i) \cdot \text{Rot}(x, \alpha_i), \quad (2.3)$$

where

$$\text{Trans}(z, d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$\text{Rot}(z, \theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$\text{Trans}(x, a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

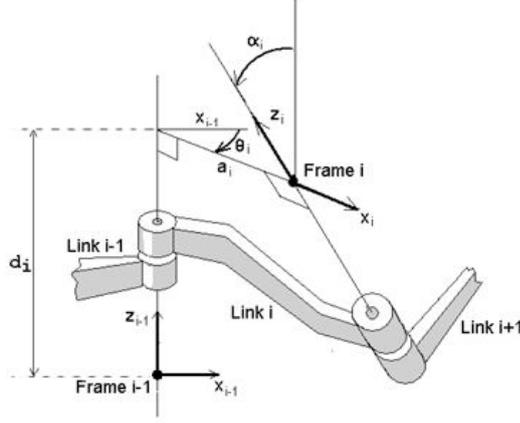


Figure 2.15: DH-Parameters. The link offset d_i is the distance along axis z_{i-1} to the point where the common perpendicular to axis z_i is located; the link length a_i is the length of the common perpendicular to z_{i-1} and z_i ; the link angle θ_i is the angle around z_{i-1} that the common perpendicular makes with vector x_{i-1} ; the link twist α_i is the angle around x_i that vector z_i makes with vector z_{i-1} . Reprinted from [43].

$$Rot(x, \alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

Given all the joint variables of the kinematic chain, the end-effector pose matrix E with respect to the manipulator base frame F_0 can be computed by successively applying the link frame transforms:

$$E = {}^0T_1 {}^1T_2 \dots {}^{n-1}T_n \quad (2.8)$$

For simple kinematic chains, computing the Forward Kinematics (evaluating E above) is straightforward. The problem of solving for the Inverse Kinematics (IK) requires finding one or more sets of joint values from a known end-effector pose, which can be obtained analytically via the forward equation (2.8). In practice, this problem can be extremely difficult and in general is unsolved [43].

In summary, given a model of the kinematics of the wheelchair itself along with a kinematic model of the user's mobility relative to the chair, it is possible to construct a mathematical model of the user's reach relative to the chair itself, as well as a model of the motion of the chair. A specification of the robot's model is known as its configuration. This mathematical model can be used to estimate wheelchair reachability.

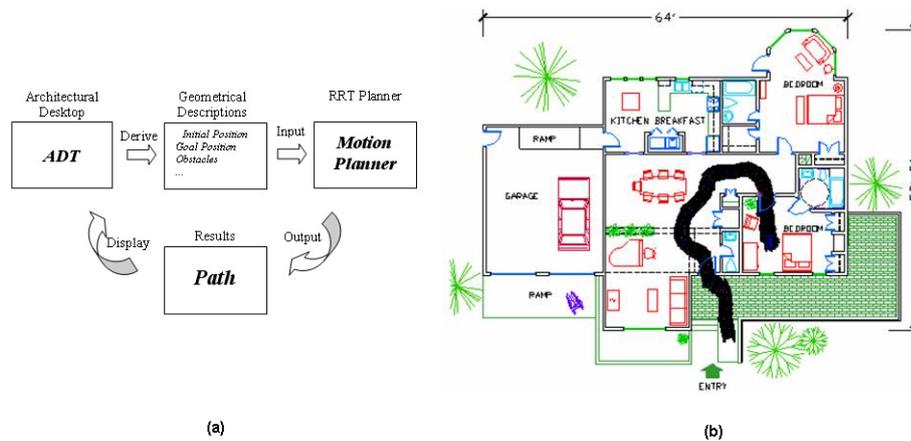


Figure 2.16: (a) Information Flow of WADA; (b) A Wheelchair Access Path Generated by WADA. Reprinted from [53].

Planning to determine accessibility

Given a model of the environment and a kinematic model of the wheelchair, accessibility assessment can be reformulated as the problem of determining if a plan for motion exists such that the hand of a user (the end-effector of a kinematic chain) constrained to a wheelchair (the mobile base) can reach the point in space for which accessibility is desired. Similarly the problem of determining accessibility becomes the problem of determining the set of reachable end-effector positions from a given starting pose.

Han et al. [18] suggested that wheelchair simulation could be used to demonstrate the performance of a wheelchair against a built environment in order to avoid the ambiguity and complexity associated with traditional prescriptive-based approaches to accessibility assessment. The idea here is to use motion planning techniques to model the possible performance of a wheelchair within a designed environment. The behavior of a moving wheelchair is simulated directly under the constraints of the wheelchair itself, the geometry of the environment, and guidelines for accessibility analysis. The set of building elements that needs to be accessible (such as doors and toilets) are represented within the simulation. The motion planning technique determines accessible route segments between adjacent elements by verifying the existence of adequate paths for a wheelchair to negotiate given the kinematic constraints imposed by the wheelchair. In order to consider and evaluate environment modifications, the designer can make “what-if” changes to the environment and invoke the motion planner to analyze the accessibility of each design option.

As a product of Han’s research, the Wheelchair-Accessible Design Assistant (WADA) [53] was developed. WADA uses a 2D simulation of the environment and the

wheelchair (Figure 2.16 (a)) to assess accessibility. The WADA derives the geometrical description of an architectural space from the Architectural Desktop (ADT) and then sends this information to a motion planner (together with a wheelchair description and a set of input vectors that represent the nonholonomic constraints of the wheelchair). The motion planner then determines whether or not the space is accessible (Figure 2.16 (b)) and sends the results back to the Architectural Desktop (ADT) [53].

Han et al.'s work demonstrated the power of motion planning in automated accessibility assessment, but it also suggests areas for improvement. For instance, in [18] and [53] the planning is done in 2D so it cannot determine if the wheelchair user can reach the light switches, electrical outlets, etc. as required in the Fair Housing Act [47]. Han initially used a potential field planner, and constrained the wheelchair motion to three options - left, right and forward. Later [54] improved the approach to include a larger set of motions by using a Rapidly-exploring Random Tree (RRT) planner, although the dimension of the environment was still restricted to 2D. As higher degree of freedom introduces higher complexity the choice of motion planner is critical. The properties of various motion planning techniques and their suitability for accessibility assessment are discussed in the next section.

2.2 Motion Planning

Motion planning emerged as a crucial and productive research area in robotics in the late 1960's [37]. There are various aspects in this field. This section provides a brief overview of the basic motion planning problem and provides a survey of current techniques with special attention paid to planners appropriate for high dimensional search spaces such as the Probabilistic Roadmap Method (PRM) and its variants.

2.2.1 Definition and application

A robot \mathcal{A} moves in a Euclidean space \mathcal{W} , called its workspace, represented over \mathbb{R}^N with N (typically in the range of 2 or 3) as the spatial dimension. B_1, B_2, \dots, B_q are fixed rigid objects (called obstacles) distributed in \mathcal{W} . The configuration space of \mathcal{A} is the space \mathcal{C} of all possible configurations of \mathcal{A} , and it is a key concept in solving motion planning problems. The obstacles B_i 's cause some regions in \mathcal{C} to be forbidden. Every B_i maps in \mathcal{C} to a set of configurations at which \mathcal{A} collides with B_i , which is called the C-obstacle of B_i and is denoted as \mathcal{CB}_i . The set $\mathcal{C}_{free} = \mathcal{C} \setminus \bigcup_{i=1}^q \mathcal{CB}_i$ is called the free space of \mathcal{A} . A free path between two free configurations q_{init} and q_{goal} of \mathcal{A} is a continuous map $\tau: [0, 1] \rightarrow \mathcal{C}_{free}$, with $\tau(0) = q_{init}$ and $\tau(1) = q_{goal}$. Given the above, the motion planning problem can be formalized as follows: Given an initial configuration q_{init} and a goal configuration q_{goal} of \mathcal{A} , generate a free path between q_{init} and q_{goal} if there exists one, and report failure otherwise [37].

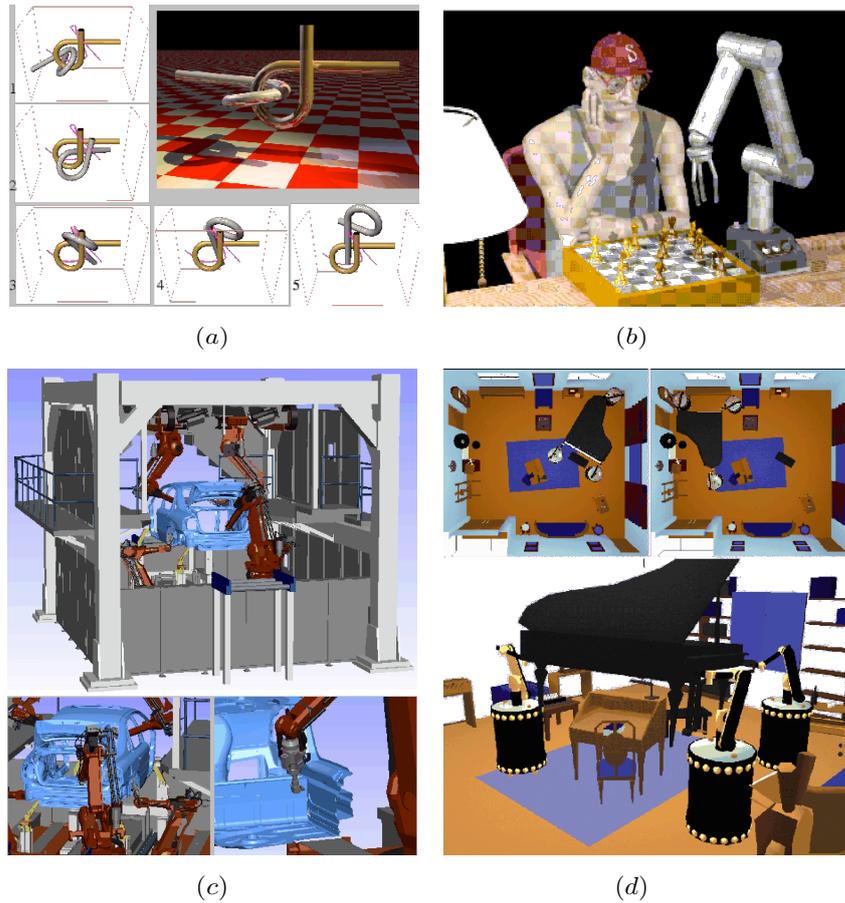


Figure 2.17: Examples of the applications of motion planning. (a) Complicated theoretical problems such as the Alpha Puzzle can be solved by motion planning techniques; (b) A digital actor plays chess with a virtual robot: the digital actor is animated while maintaining the elusive human characteristics; (c) An application of motion planning to the sealing process in automotive manufacturing: the robotic manipulators' motion can be computed automatically by specifying the high-level sealing tasks; (d) Using mobile robots to move furniture (here a grand piano): the motion planner must detect collisions between robots and with other furniture. Reprinted from [40].

Motion planning plays an important role in various fields, including robotics, manufacturing, computer games and virtual environments, drug design, and aerospace applications (see Figure 2.17). It is a fundamental computational problem in the creation of autonomous robots. It can also be used to check the geometrical feasibility of planned operations. [38] summarizes some of the important achievements in the development of motion planning techniques and discusses the problems regarding computational issues.

Traditional motion planning approaches for few-DOF robots can be very efficient, even in large and complex environments encountered in practical problems. For many-DOF problems and more complex problems, including planning with dynamic constraints and planning for reconfigurable robots, traditional planning is inappropriate and randomized or probabilistic algorithms are needed.

2.2.2 Traditional motion planning approaches

Motion planning algorithms entail considering the abilities of the robot and the structure of the environment to solve the planning problem. They usually (see [37, 11]) break the problem into two basic steps: (i) Define a graph to represent a geometric structure of the environment; (ii) Perform a graph search to find a connected path between the node corresponding to q_{start} and the node corresponding to q_{goal} . Although there exist a large number of methods for solving the motion planning problem most traditional methods are based on one of three general approaches: roadmap, cell decomposition, and potential field [37, 11, 57].

Roadmap

The roadmap approach is based on the formal concept of configuration space. The roadmap $\mathcal{R} = (N, E)$ is a graph, i.e. a network of one-dimensional curves, capturing the connectivity of \mathcal{C}_{free} , where N is a set of configurations of \mathcal{A} appropriately chosen over \mathcal{C}_{free} , E is a set of (simple) paths; an edge (a, b) corresponds to a feasible path connecting the configuration a and b . Once \mathcal{R} is constructed, each motion-planning query is processed by first connecting q_{init} and q_{goal} to two nodes in \mathcal{R} and then searching the roadmap for a path connecting these two nodes (e.g. using a graph search algorithm such as Breath-First search, Dijkstra's algorithm or A* search).

A well constructed roadmap \mathcal{R} should have these two characteristics [57]: (1) Any configuration in \mathcal{C} can be easily connected to \mathcal{R} ; (2) Each connected component in \mathcal{R} corresponds to one and only one connected component in \mathcal{C} . Classical examples of roadmaps include visibility graphs [42] (Figure 2.18(a)), Voronoi diagrams [45] (Figure 2.18(b)), and silhouettes [10].

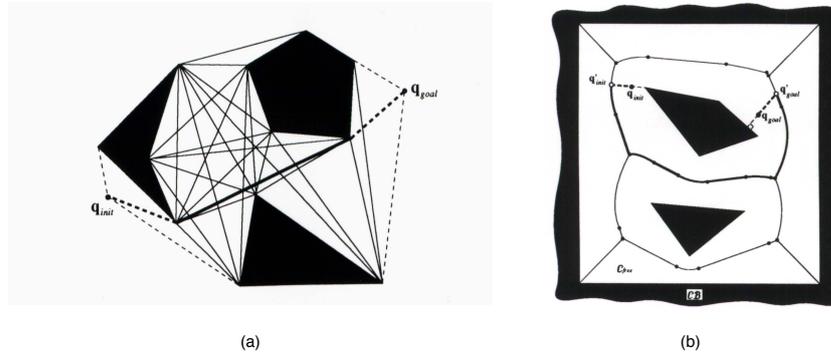


Figure 2.18: Examples of roadmap approaches applied on two-dimensional configuration spaces with polygonal C-obstacles. The shaded areas represent obstacles. The solid lines are the edges of \mathcal{R} . The dotted lines connect q_{init} and q_{goal} to \mathcal{R} . (a) In the visibility graph, the vertices of C-obstacles are connected by a link if the straight line segment joining them does not intersect the C-obstacles' interior. (b) The Voronoi diagram is a network of line segments and parabolic curves, which are the set of points equidistant from at least two C-obstacles. The path generated by this method keeps the robot as far away from the obstacles as possible, unlike the visibility graph. Reprinted from [37].

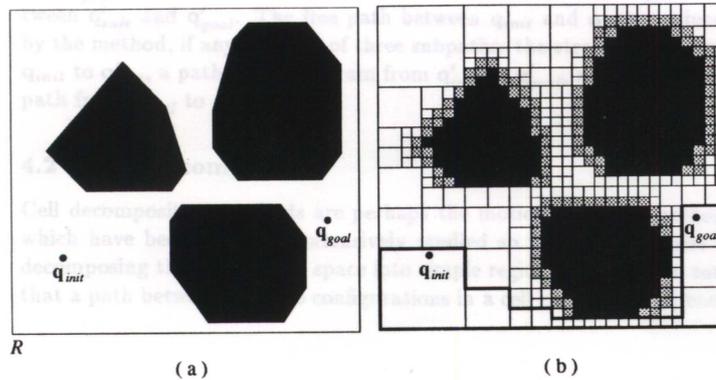


Figure 2.19: An illustration of the quadtree decomposition method in a two-dimensional configuration space. (a) The configuration space is bounded by a rectangle and contains three polygonal C-obstacles. (b) The rectangle is divided into four identical rectangles. If the interior of a rectangle lies completely in \mathcal{C}_{free} or in some \mathcal{CB}_i , then it is not decomposed further. Otherwise, it is recursively decomposed into four rectangles until some predefined resolution is attained. A channel extracted from this decomposition is shown in bold contour. Reprinted from [37].

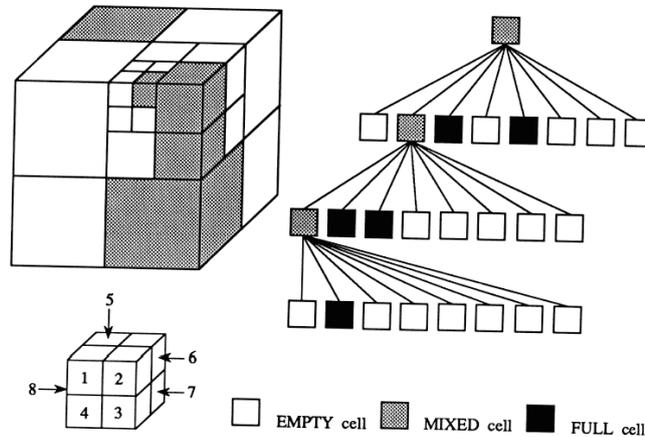


Figure 2.20: Octree decomposition is very similar to the quadtree decomposition. Instead of rectangles, each node in octree is a cube. Each cube can be recursively decomposed into eight cubes. Reprinted from [37].

Cell decomposition

The idea behind cell decomposition methods is to decompose \mathcal{C}_{free} into simple regions, called cells. A cell is marked as 1 (dark) if it is occupied by a C-obstacle, or else marked as 0 (white) if it is free. An undirected connectivity graph is constructed to represent the adjacency relation between the cells. This graph can then be searched to find a sequence of cells connecting the two cells that contain q_{init} and q_{goal} , from which the final free path can be extracted.

Two categories of cell decomposition methods exist: exact and approximate ones. Exact cell decomposition methods decompose \mathcal{C}_{free} into cells whose union is exactly \mathcal{C}_{free} , but approximate ones do not. Examples of exact methods include trapezoidal maps and triangulation maps (tetrahedralization maps in 3D) [14]. Many approximate cell decomposition methods have a recursive nature. A coarse approximation becomes finer at each level by subdividing cells that partially overlap both free and forbidden space. Examples of such recursive representations include the quadtree shown in Figure 2.19 and the octree shown in Figure 2.20).

Potential field

The potential field method was initially proposed for online collision avoidance, but can also be used to solve general planning problems [38]. Unlike the previous two approaches, the potential field method does not pre-compute a connectivity graph. Instead it uses an artificial potential function as a heuristic to guide the search for a path. The function is

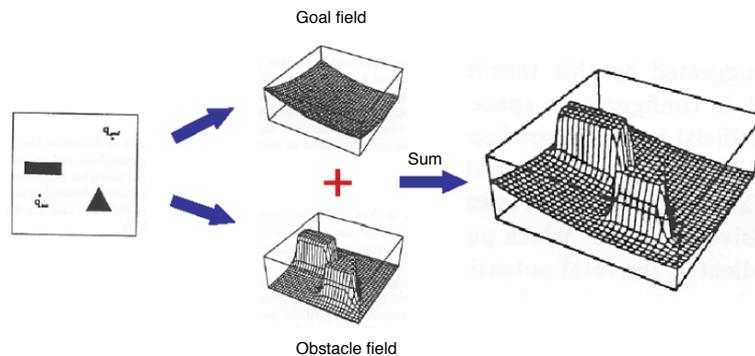


Figure 2.21: Potential field function can be sum of the goal field function and the obstacle field function. Pictures derived from [37].

produced by a goal configuration as an “attractive potential” which pulls \mathcal{A} toward the goal and the C-obstacles as a “repulsive potential” which pushes \mathcal{A} away from them (see Figure 2.21). The negated gradient at a given configuration q suggests the most promising direction of motion at q . Potential field methods can be very efficient and compute optimized solutions, but they can become trapped in a local minima of the function and it is difficult to construct local-minima-free potential functions [38, 57].

Summary

An important issue in motion planning is the computational complexity of the algorithms, which has attracted interest from theoretical computer science researchers since the 1980’s [55, 59, 10]. A complete solution to the motion planning problem is known to be exponential to the robot’s degree of freedom and a historical account of the computational analysis is given in [10]. The traditional planning algorithms described in this section are resolution-complete, if properly implemented: they always find a path if there exists one. They can solve complex path planning problems in 2D and 3D configuration spaces quickly, but none of these planners extends well to robots with more than 4 or 5 DOFs [38, 57]. Solutions to these more complex problems require probabilistic solutions, and these are discussed below.

2.2.3 Randomized planning algorithm

A number of heuristic techniques have been proposed for high-dimensional planning. Perhaps the earliest of these is the Randomized Path Planner (RPP), a variant of the potential field method which was introduced in 1991 [3, 2]. The RPP alternates “down motions” to track the negated gradient of a potential field and “random motions” to escape local minima. The approach has proven to be successful for planning for robots with 3 to 31 DOFs.

To avoid pathological cases caused by the deterministic potential field, the Probabilistic Roadmap Method (PRM) [27, 28, 29, 22] was later developed and continues to be used and developed (see [23, 5, 26, 57, 21, 13]). It is important to note that these methods are probabilistic and it is not guaranteed that these planners will find a path even though one exists, but if they do find a path it will take the device from the initial configuration to the goal.

Probabilistic Roadmap Method (PRM)

The Probabilistic Roadmap Method (PRM) [27, 28, 29, 22] is one of the most successful methods for solving complex motion planning problems. It belongs to the roadmap approach introduced in previous section. The main difference between the PRM and earlier approaches is that the PRM does not attempt to construct an exact representation of the shape of \mathcal{C}_{free} so that the roadmap \mathcal{R} can be constructed in reasonable time. The idea here is to create a very simplified \mathcal{R} that approximately “covers” \mathcal{C}_{free} . The basic PRM proceeds in two main phases: the preprocessing phase and the query phase.

In the preprocessing phase (also referred as learning phase or roadmap construction phase in the literature) \mathcal{R} is constructed by connecting randomly sampled collision-free configurations using a simple local planner. The sequence of steps in this phase is outlined below:

1. *Generation of random nodes* (Figure 2.22(b,c)). Configurations are sampled by picking a random configuration of \mathcal{A} . The basic PRM uses a uniform sampling scheme which has difficulty in finding paths through narrow passages in the scene, because it places many samples in “open” regions but not enough samples in “tight” regions. A thorough analysis on this issue is given in [24]. The sampling methodology is crucial to the algorithm’s performance, and many different sampling strategies have been proposed. The earliest methods used the form of the constructed roadmap to add more samples in the neighborhood of nodes that were connected with few neighbors in an “enhancement phase” [27, 29]. Several configuration space-based methods have been developed, including the dilated free space method [24], and MAPRM (a PRM with sampling on the medial axis of \mathcal{C}_{free}) [71]. However, these methods can be expensive to implement in high dimensional configuration spaces. Considering the low dimension of \mathcal{W} (typically only two or three) and the correspondence of narrow passages in \mathcal{C} and \mathcal{W} , methods based on workspace information to guide the sampling have proven to be successful. Such methods sample more densely near obstacle boundaries [1, 9] or far from obstacles, on the medial axis [21], or in “difficult” regions obtained directly from the workspace computation [5, 35]. A general sampling approach that recursively refines an initially sparse sampling was introduced in [13]. This hierarchical PRM (HPRM) is discussed in the next section.

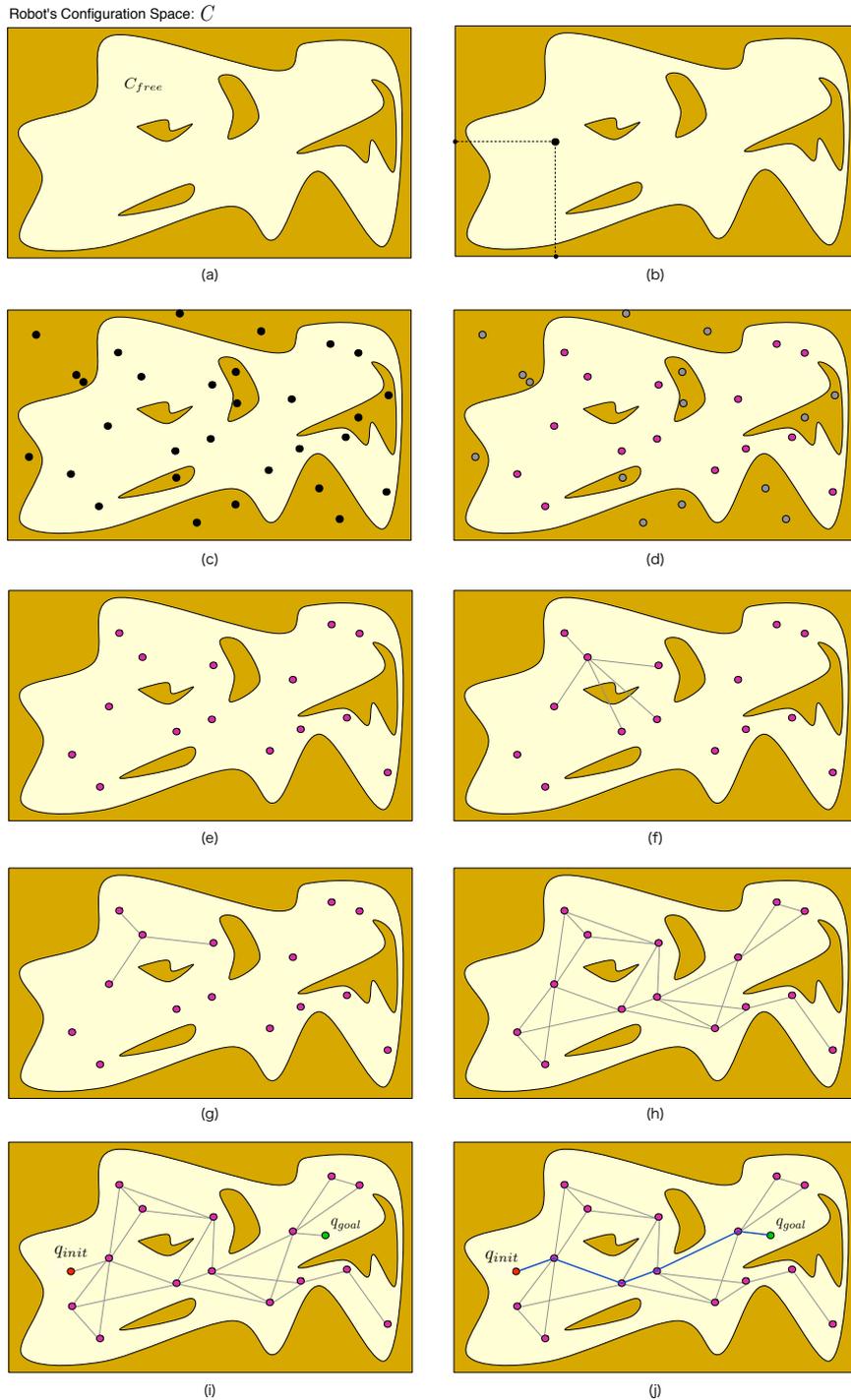


Figure 2.22: Basic PRM steps. See text for details. Figure derived from [11].

2. *Collision detection* (Figure 2.22(d,e)). Sampled configurations from the above step are tested for collision with obstacles and self-collision in workspace. Collision-free configurations (configurations in \mathcal{C}_{free}) are retained in \mathcal{R} . There exist a variety of techniques for efficient collision detection such as the grid method and the Bounding Volume Hierarchy (BVH) method [70, 33, 62]. The grid method is similar to the cell decomposition method described in the previous section. The main difference between them is that the grid method for collision decomposes the workspace rather than the configuration space. The BVH method approximates each object with a tree of bounding volumes (spheres or boxes). The bounding volume at a node encloses the bounding volumes of its children. If the leaves of two trees overlap, then there is a collision between the two objects the trees represent. Usually the first two steps are interleaved until a pre-specified number N of nodes has been computed.
3. *Interconnection of the nodes with a local planner* (Figure 2.22(f,g,h)). Given some metric defined on \mathcal{C} , for each node x , all the other nodes are ordered according to increasing distance from x and the local planner tries to connect x to each of the K (K is a predefined parameter) closest nodes. Choosing the proper distance function, local planner, and K is important to the performance of the PRM planner. There are tradeoffs in the choice of the local planner. Powerful local planners often succeed in finding a local path when one exists, but they take more time and require more space to store the local paths. Simple deterministic local planners are less successful in connecting two nodes and thus require more nodes to be generated in the roadmaps, but the local paths computed do not need to be recorded since they can easily be recovered in the query phase. Best experimental results have been obtained using simple deterministic planners [65, 29]. As an example, a typical fast local planner that tries to connect two configurations with a straight line in \mathcal{W} has shown success in solving planning problems for very high-dimensional holonomic robots [27, 29].

In the query phase, a query (q_{init}, q_{goal}) is processed by first connecting q_{init} and q_{goal} to \mathcal{R} (Figure 2.22(i)). Assume that \mathcal{R} is a single-component graph. If the attempt to connect q_{init} and q_{goal} to \mathcal{R} fails, then report failure. Otherwise, perform a graph search on \mathcal{R} for a global path that starts at q_{init} followed by a concatenation of local paths and ends at q_{goal} (Figure 2.22(j)). The local paths here are recomputed without collision checking and should be the same as the ones computed when the roadmap was constructed.

The execution of the two phases can be varied in different situations. A PRM planner may pre-compute a roadmap and use it to process many queries. It requires the constructed roadmap “cover” the entire \mathcal{C}_{free} well. In contrast, single-query planners build a new roadmap for each new query and do not have to achieve a good coverage of \mathcal{C}_{free} .

Note that the quality of the computed global path may not be satisfying as it relies on the sampling scheme, the number of nodes in \mathcal{R} , properties of the local path planner, and

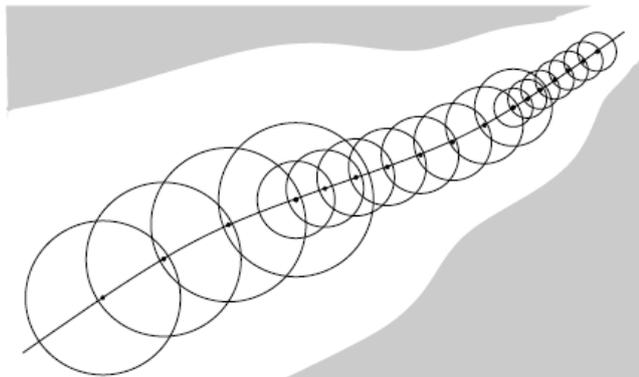


Figure 2.23: A path given by HPRM: higher clearance parts of the path are covered by larger balls. Reprinted from [13].

the global search method. In practice a good path may be required to keep some distance (clearance) from the obstacles and be smooth, and many techniques that can improve the path quality exist. For example, [32] uses an augmented version of Dijkstra’s algorithm by incorporating a higher cost for edges that have small clearance, [39] randomly selects segments of the global path and shortcuts them by the local planner, and in [4] local geometric operations (i.e. cutting off triangle corners) are performed iteratively on the global path.

Hierarchical PRM (HPRM)

As a hierarchical variant of the PRM, Hierarchical PRM (HPRM) [13] samples \mathcal{C}_{free} adaptively by recursively refining an initially sparse sampling. Sampling is biased towards difficult regions by using the information contained in the nodes to estimate the local size of \mathcal{C}_{free} . The key element in the approach is the clearance $cl(c)$ – the distance from a configuration $c \in \mathcal{C}_{free}$ to the nearest point on an obstacles. Imagine covering a path τ in \mathcal{C} with spheres whose radius is indicated by the local clearance of τ (see Figure 2.23). With a clearance oracle which determines whether $cl(c) \leq r$ for a given c and r , it suffices to sample densely only in regions of small clearance [12]. This method not only generates a smaller roadmap than that obtained by uniform sampling, but also has a higher probability of success. The main drawback is the running time of the clearance oracle. Although this can be reduced by only computing a bound on the clearance, it still requires an examination of the entire \mathcal{C} , which can be expensive.

PRM in dynamic and reconfigurable environments

While most motion planning strategies assume a static environment, environments are hardly static in reality. Considerable research has been done in motion planning in the presence of dynamic objects, and some of these approaches are based on or are related to PRM (e.g., [41, 6, 69]). Different dynamic environments can be defined. The change of dynamic obstacles might be predefined [6, 69] or unknown [41], it might be continuous [69] or occasional [6]. A straightforward approach to model a dynamic environment with a PRM would be to create the roadmap as if none of the moving obstacles is present and to then mark parts of the roadmap as being temporarily blocked as long as they are occupied by a moving obstacle. For example, [41] uses a voxel grid covering the environment, and introduces an efficient mapping from the voxels to nodes and edges in the roadmap. When some obstacle changes position, it is easy to identify the nodes and edges invalidated by the motion of the obstacle.

Given the specifics of the Accessibility Assessment problem here we are specifically interested in a static environment with prior knowledge of the potential placements of “changeable” obstacles. That is a set of related static environments where the environments are identical except for the removal (or addition) of specific known obstacles. Although related to dynamic environments, such reconfigurable environments are simpler in that the changes are discrete rather than continuous. A solution to this reconfigurable environment is given by [6] which describes the development of the robust roadmap approach. In the robust roadmap, each edge has an associated potential set describing for which obstacle placements the roadmap is valid.

2.3 Summary

This chapter reviewed several aspects of accessibility assessment, from its social importance to a discussion of the existing technologies supporting it. As 3D sensing, modeling and planning technologies have advanced, intelligent 3D systems for fast and accurate accessibility assessment of built environments for different types of mobility are being developed. Such techniques can be used build 3D geometry models that can be used for accessibility assessment. The second half of the chapter focused on related research in motion planning. Users seated in wheelchairs and capable of reaching from the chair can be modelled as high DOF kinematic structures connected to a wheeled mobile base. Accessibility assessment can be framed as the task of estimating that portion of the environment that can be reached by such a device – a task that is equivalent to motion planning. Planning in high DOF environments requires sophisticated search algorithms and PRM algorithms and its variants are an appropriate choice for solving planning problems for high DOFs within a reasonable time. They can also be adapted for efficient replanning in different but

related environments. In this report, a PRM-based planner is used to perform accessibility assessment over a collection of possible modifications to an environment. The details of the technique are presented in Chapter 3.

Chapter 3

Accessibility Assessment

The problem of accessibility assessment is to find that part of the environment that is accessible to people with mobility limits and to suggest modifications of the environment to improve the size of the region that is accessible. The efficiency of randomized and probabilistic motion planners makes them attractive as a primitive upon which accessibility assessment algorithms can be constructed. In this chapter a PRM variant is presented and this is combined with an environmental planner to solve the accessibility assessment problem.

Traditionally the problem of high-dimensional motion planning assumes that each of the joint angles are equivalent but in the accessibility assessment domain (and likely in many other domains) not all DOFs are ‘equal.’ Consider a person in a wheelchair who attempts to reach an object in the environment. It is more likely that the person will move the wheelchair to an area close to the object first and then move his arm than to first move the arm and then the wheelchair. Clearly in this task the movement of the arm is ‘secondary’ to that of the wheelchair in terms of the wheelchair user’s reachability. Motivated by this observation we explore a hierarchical structure of the DOFs of the kinematic device to improve the efficiency of the search process.

In this hierarchy we consider subversions of the kinematic structure in which subversional joints are considered over their range of motion. This effective search mechanism forms the core of the accessibility assessment. Given a effective mechanism to determine the reachable portion of space a search process is used to determine which modifiable obstacles should be modified and how they should be modified in order to maximize reachability. The algorithm is sketched in Figure 3.1.

This chapter starts by giving a formal statement of the problem of accessibility assessment which is intended to formalize the definition of the reachable workspace of a kinematic structure as well as to characterize the potential environmental modifications (Section 3.1). Then a hierarchical variant of the PRM is presented (Section 3.2), followed by an environmental planner which is used to evaluate potential environmental modifications (Section 3.3).

3.1 Formal statement of the problem

The problem of accessibility assessment involves maximizing the reachable portion of the environment, given (i) a fixed and known environment, (ii) a set of modifiable environmental features, and (iii) a set of kinematic constraints introduced by the wheelchair and the user (here modeled as a generic ‘arm’). We begin by defining each of these properties in terms of motion planning notation.

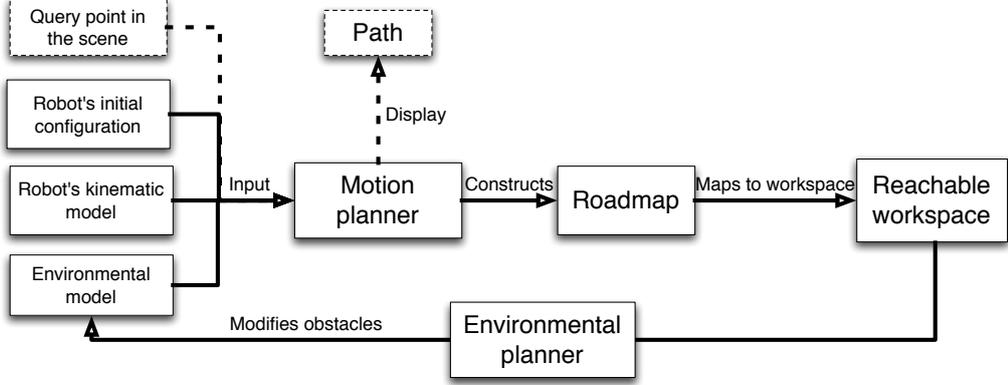


Figure 3.1: The workflow of the accessibility assessment. The solid lines are the main steps in the process. The dotted lines are the optional steps, which let a user know if a point in the scene is reachable.

3.1.1 Environment

Following [37], let \mathcal{A} be a kinematic device defined in an n -dimensional configuration space \mathcal{C} , and operating on a plane in a three-dimensional Euclidean space \mathcal{W} . \mathcal{A} consists of a mobile base \mathcal{A}_{base} and an attached kinematic chain \mathcal{A}_{arm} . \mathcal{W} consists of permanent obstacles $\mathcal{O} = \{o_1, o_2, \dots, o_l\}$ and modifiable obstacles $\mathcal{M} = \{m_1, m_2, \dots, m_h\}$. The environment is static in real time, i.e. once \mathcal{A} is “turned on” then no obstacle will change position. Each obstacle has exactly one pose in \mathcal{W} . A modifiable obstacle m_i may be in one of k_i predefined possible states. Following the concept of potential placements introduced by [6], for each m_i , a set of potential placements $\mathcal{PP}(m_i) = \{pp_i^0, pp_i^1, \dots, pp_i^{k_i}\}$ is defined, which includes one pp_i^0 representing that m_i is not present in the environment, and pp_i^1 representing that m_i is in some default (initial) location. There is a finite number k_i ($k_i \geq 2$) of discrete placements of m_i . Each $pp_i^j \in \mathcal{PP}(m_i)$ ($0 \leq j \leq k_i$) maps m_i to a placement in \mathcal{W} .

There are $\prod_{i=1}^h k_i$ possible combinations of placements of all modifiable obstacles. Let $\mathcal{P}^1, \mathcal{P}^2, \dots$ denote members of the placements superset. These represent all possible “configurations” of the environment. To be more precise, each \mathcal{P}^x can be imagined as a vector of length h $(pp_1^{x_1}, pp_2^{x_2}, \dots, pp_h^{x_h})$, where the i -th entry $pp_i^{x_i}$ denotes the placement of m_i . Let $\mathcal{P}^1 = (pp_1^1, pp_2^1, \dots, pp_h^1)$ denote the initial environment and $\mathcal{P}^0 = (pp_1^0, pp_2^0, \dots, pp_h^0)$ denote the minimalist environment in which all modifiable obstacles have been removed.

3.1.2 Reachable workspace

Given an initial configuration $c_{init} \in \mathcal{C}_{free}$ and \mathcal{P}^i , the reachable workspace $\mathcal{W}_{reach}(\mathcal{P}^i)$ of \mathcal{A} is the set of points in \mathcal{W} reachable by the end-effector of \mathcal{A} ’s manipulator from c_{init} . Here, for $w \in \mathcal{W}_{reach}(\mathcal{P}^i)$, w is “reachable” iff it satisfies the following two criteria:

1. There exists at least one configuration $c \in \mathcal{C}_{free}$ such that the end-effector is positioned at w ;
2. There exists at least one path from c_{init} to c that \mathcal{A} can execute (subject to both kinematic and obstacle constraints).

The size of \mathcal{W}_{reach} is denoted as $|\mathcal{W}_{reach}|$ and is defined as the volume of the space \mathcal{W}_{reach} in \mathcal{W} . Here we define that $\mathcal{W}_{reach}(\mathcal{P}^x)$ is larger than $\mathcal{W}_{reach}(\mathcal{P}^y)$ iff $|\mathcal{W}_{reach}(\mathcal{P}^x)| > |\mathcal{W}_{reach}(\mathcal{P}^y)|$. However, we do note that other definitions can be imagined. For example, a reachable workspace with more feasible paths might be considered larger than the other. It is, however, sufficient to use this simple definition for now.

3.1.3 Accessibility assessment

Given the formalism, accessibility assessment can be decomposed into the task of finding the non-trivial placement \mathcal{P}^{opt} that has the maximum reachability $|\mathcal{W}_{reach}(\mathcal{P}^{opt})|$ (clearly \mathcal{P}^{opt} is no better than $(pp_1^0, pp_2^0, \dots, pp_h^0)$). The problem is that the set of all possible environmental configurations is very large and it is impractical to search through them in an exhaustive manner. Here we take the simple greedy approach to search from \mathcal{P}^1 to the ‘best’ environment within a bounded modification cost as defined below.

Define $\zeta(pp_i^{x_i})$ to be the cost function of modifying obstacle m_i from its initial configuration pp_i^1 to $pp_i^{x_i}$. Assume that the cost of changing objects is independent, then $\zeta(\mathcal{P}^x)$, the overall cost of the modification of each modifiable obstacle from \mathcal{P}^1 to \mathcal{P}^x , is given by $\sum_{i=1}^n \zeta(pp_i^{x_i})$. The cost function can be implemented in many ways. In general, the cost of keeping the obstacle in its initial configuration should be zero and the cost of removing it from the scene entirely should always be more expensive than any other modifications. Formally, the cost function should satisfy the following criteria:

1. $\forall i \in [1..h], \zeta(pp_i^1) = 0$;
2. $\forall i \in [1..h], \forall j \in [1..k_i], \zeta(pp_i^0) > \zeta(pp_i^j)$.

Accessibility assessment makes use of two sub-algorithms: a motion planner and an environmental planner (see Figure 3.1). The motion planner is the core. It takes the environmental model, the user’s kinematic model and initial configuration as inputs, and constructs a roadmap in \mathcal{C}_{free} , which can be mapped to the workspace to estimate \mathcal{W}_{reach} . Then the environmental planner is executed iteratively to modify obstacles until an optimized environment configuration \mathcal{P}_{opt} is found. These two planners are discussed in the following sections.

3.2 Motion planner: hierarchical PRM

In many ways traditional PRMs treat the entire world as being equally difficult and all of the joint angles as being equally effective in terms of solving the problem. In the accessibility assessment domain these assumptions are quite conservative. This section first describes a few observations of this followed by a formal definition of the hierarchical roadmap. Finally a new PRM variant based on this hierarchical roadmap is described.

3.2.1 Observations

Effects of \mathcal{A}_{base} and \mathcal{A}_{arm}

For accessibility assessment, the problem domain is likely to be ‘easy’ in many regions and only ‘difficult’ in certain areas. For example, a normal indoor environment is likely to have large regions of open space within which the computational costs associated with PRM are excessive and where simpler approaches would be suitable. At the same time, it is likely that certain degrees of freedom of the kinematic structure should be evaluated before others. Imagine a person trying to get an object from far away, they probably first move to the area near the object and then reach for it. For the accessibility assessment problem, we observe that movements of \mathcal{A}_{base} often have a more significant effect on \mathcal{A} than movements of \mathcal{A}_{arm} . Observe that there are two computations that are critical in accessibility assessment: Occupancy Analysis (can the kinematic structure exist at a particular configuration without striking obstacles?), and Reachability Analysis (starting from some initial configuration, where can the kinematic structure reach?). If we know a priori that certain dimensions of the problem are more likely to be critical to the solution than others, we can exploit this in terms of both computations. We begin by considering this informally and then provide a more formal definition.

Occupancy analysis

In PRM, configuration space obstacles are not represented explicitly, instead, a collision checker determines whether a configuration is free or not. Collision detection is performed to verify the random selected nodes as well as the local paths connecting them. This is one reason for the high computational cost of the preprocessing phase of the PRM. Exploiting the hierarchical nature of the problem in accessibility assessment and the large areas of open space ‘combined’ joints can be used to detect collisions more effectively.

The occupancy of the robot in \mathcal{W} is used for collision detection. When a configuration is checked for collision, the position and orientation of each part of the robot in \mathcal{W} are computed and tested against all obstacles. It is possible to perform these tests for collisions using conservative approximations of the wheelchair and user. For example in Figure 3.2

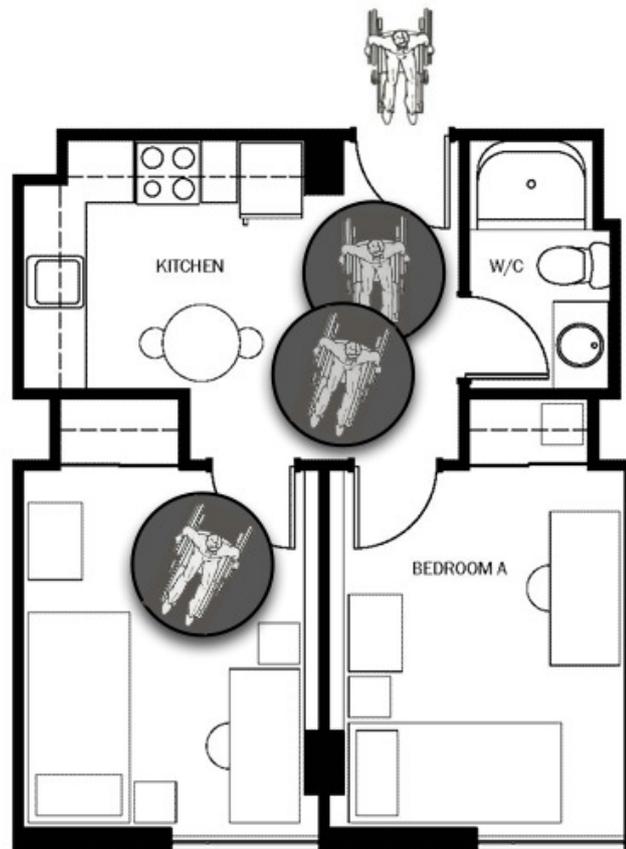


Figure 3.2: There exist open regions in a typical residence environment, the wheelchair can be estimated as a cylinder for efficient collision detection.

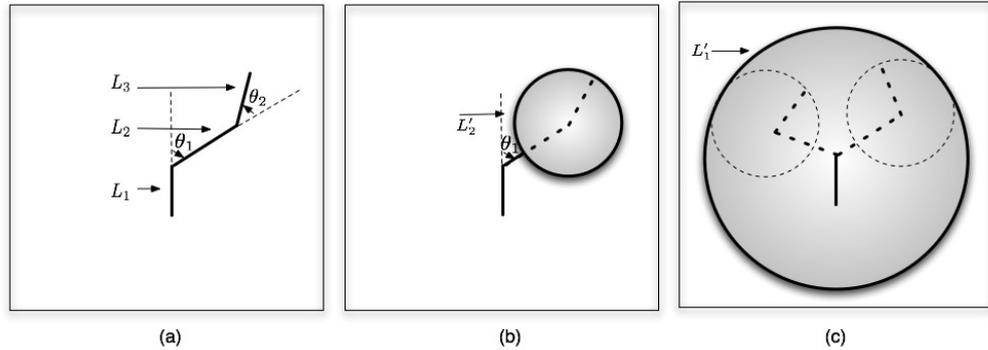


Figure 3.3: An illustration of the combination of a planer manipulator that consists of two revolute joints. Testing for intersections of the object in (a) involves a 2D search process. (b) shows a conservative approximation to (a) in which every possible joint angle θ_2 is considered at each instance by replacing L_3 and θ_2 by a single circular object. Any configuration that is not in \mathcal{C}_{free} for (a) will also not be in \mathcal{C}_{free} for (b). (c) shows an even more conservative version of (a) in which both θ_1 and θ_2 have been replaced with a conservative physical structure.

collisions between a person in a wheelchair can be estimated as a large cylinder in open regions of space, while more accurate (and hence more expensive) representations may be required near obstacles.

We can exploit this property in a hierarchical manner where more DOFs are available. Consider the planar 3-link manipulator example shown in Figure 3.3. The structure is indexed by θ_1 and θ_2 as shown in Figure 3.3(a) but it can be represented by the larger physical structure shown in Figures 3.3(b) and (c). Determining if a given configuration (θ_1, θ_2) is in \mathcal{C}_{free} can be tested first by checking to see if the structure in Figure 3.3(c) is in \mathcal{C}_{free} . If it is, then we are done. If it is not, then the structure in Figure 3.3(b) with the specific value of θ_1 should be tested. Again, if this test shows that the structure is in \mathcal{C}_{free} then we are done. If neither of these tests succeed then the original object must be tested.

It is important to note that these conservative approximations not only test specific configurations of the robot for collision but if they succeed they also test a wide range of configurations for reachability. As the goal here is to determine all reachable configurations, this more conservative approximation to the kinematic structure is particularly powerful.

Reachability analysis

In the traditional PRM algorithm, nodes are sampled by randomly choosing a value at each of \mathcal{A} 's DOFs. Each node corresponds to an exact pose of \mathcal{A} in \mathcal{W} , at which \mathcal{A} 's end-effector reaches one point in \mathcal{W} (Figure 3.4(a)). As a result, the reachable area mapped

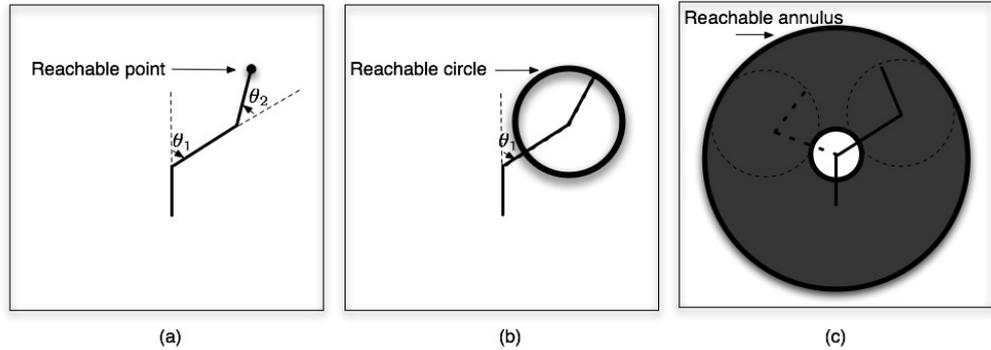


Figure 3.4: (a) Given a valid configuration, only a reachable point can be determined by computing the position of the end-effector. (b) Rotating the L_3 by any angle turns a reachable point into a reachable circle. (c) Shows an even more conservative version of (a) in which both θ_1 and θ_2 can take any value.

from a collection of random nodes is a set of points distributed over \mathcal{W} . The PRM uses a network of random nodes to capture the connectivity of the \mathcal{C}_{free} such that an explicit representation of the entire \mathcal{C}_{free} can be avoided. In accessibility assessment not only is the connectivity of the workspace a concern but the size of the reachable portion is also of interest.

The reachability problem can be seen as a sub-problem of the occupancy problem, since the robot's reachable area is a subset of its occupied area. Consider the example shown in Figure 3.4, in open regions the reachable point becomes a reachable circle or annulus. For a general kinematic structure, by ordering its DOFs and applying this strategy hierarchically we may be able to explore the structure's reachability more efficiently.

3.2.2 Hierarchical probabilistic roadmaps

Given the above observations in considering the problem of accessibility assessment, we can exploit specific properties of the domain. Specifically we can order the DOF of the kinematic structure and apply a hierarchical approach to the planning task. We begin by extending the definition of the traditional roadmap given an 'ordered importance' of the configuration space.

Normally a configuration c of \mathcal{A} is written as a vector of length n , say $c = (j_1 \dots j_n)$. Instead we seek a representation within which certain joint angles are 'free' and can assume arbitrary values within some previously defined domain. Order the joints such that more important joints have a lower index. Let the domain of j_i be \mathcal{D}_i , $c^r = (j_1, j_2, \dots, j_r)$ is a subset of $\mathcal{D}_1 \times \mathcal{D}_2 \times \dots \times \mathcal{D}_n$, given by $\{\forall x_{r+1} \in \mathcal{D}_{r+1}, x_{r+2} \in \mathcal{D}_{r+2}, \dots, x_n \in \mathcal{D}_n \mid (j_1, j_2, \dots, j_r, x_{r+1}, x_{r+2}, \dots, x_n)\}$. That is c^r is the set of possible configurations with joints

1... r having specific values but joints $r + 1...n$ being free. This hierarchical concept applies to general kinematic structures in the domain of motion planning. To establish the general representation of c^r , we first order the n DOFs, then fix the first r DOFs and take all possible values of the remaining DOFs to construct the hierarchical body. This definition implies the hierarchy: joint i is “more important” than joint $i + 1$.

A configuration c is said to be valid if the robot in configuration c is in the free space of \mathcal{W} . Similarly c^r is said to be valid if every element of c^r is in the free space of \mathcal{W} . Let $V(c^r)$ denote the function that returns true if c^r is valid. The reachable area of a configuration c is the points in \mathcal{W} where the end-effector can reach. The reachable area $RA_c(c^r)$ of c^r is therefore the union of the reachable points of every element in c^r . In addition, the region of the world that the robot can occupy is also of interest and let $OA_c(c^r)$ ¹ be the union of \mathcal{A} 's occupied area of every element in c^r . Under the hierarchy assumption nodes with lower r occupy and reach larger workspaces than those with higher r . To be precise, we have these three lemmas:

Lemma 3.2.1. $\forall i, j \in [0, n], i < j \wedge V(c^i) \implies V(c^j)$. For some configuration of \mathcal{A} , that its lower hierarchical representation is free implies the higher hierarchical representation is free, too.

Lemma 3.2.2. $\forall i, j \in [0, n], i < j \implies RA_c(c^i) \supseteq RA_c(c^j)$. For some configuration of \mathcal{A} , the reachable workspace of the lower hierarchy is the superset of that of the higher hierarchy.

Lemma 3.2.3. $\forall i, j \in [0, n], i < j \implies OA_c(c^i) \supseteq OA_c(c^j)$. For some configuration of \mathcal{A} , the occupied workspace of the lower hierarchy is the superset of that of the higher hierarchy.

The basics of the notation is illustrated in Figure 3.5 which shows a mobile manipulator \mathcal{A} that consists of a mobile base \mathcal{A}_{base} and a two link manipulator \mathcal{A}_{arm} . (This model is a simplified version of the kinematic structure in the problem of accessibility assessment of a user on a wheelchair with a single arm.) Based on the observation of the different effects of \mathcal{A}_{base} and \mathcal{A}_{arm} , we order the DOFs of \mathcal{A} from its base to its end-effector such that its configuration is written as an ordered array $c = (x, y, \theta, \phi_1, \phi_2)$. Suppose $\mathcal{D}_1 = [x_{min}, x_{max}]$, $\mathcal{D}_2 = [y_{min}, y_{max}]$, $\mathcal{D}_3 = [-\pi, \pi]$, $\mathcal{D}_4 = [-\pi, \pi]$, and $\mathcal{D}_5 = [-\pi, \pi]$. Figure 3.6 shows the hierarchy of occupancy, and Figure 3.7 shows the hierarchy of reachability. c^0 is not illustrated but can be easily imagined, it is the entire workspace of \mathcal{A} . Note the relationship between Figure 3.6 and Figure 3.7. If the test for occupancy for Figure 3.6(a-d) passes then the corresponding Figure 3.7(a-d) is reachable.

Hierarchical representations can be very complex shapes and they can also have holes. In practice the computation of the exact hierarchical representations is time consuming and unnecessary. For occupancy estimation conservative representations of these complex

¹The subscript c of the notations RA_c and OA_c indicates that the mappings are for nodes, while e indicates mappings for edges.

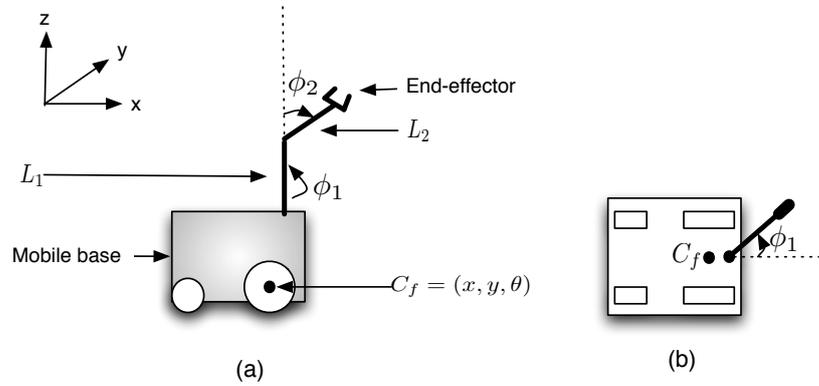


Figure 3.5: \mathcal{A} consists of \mathcal{A}_{base} and \mathcal{A}_{arm} that has two links L_1 and L_2 connected by revolute joints. The configuration of \mathcal{A} is written as $(x, y, \theta, \phi_1, \phi_2)$.

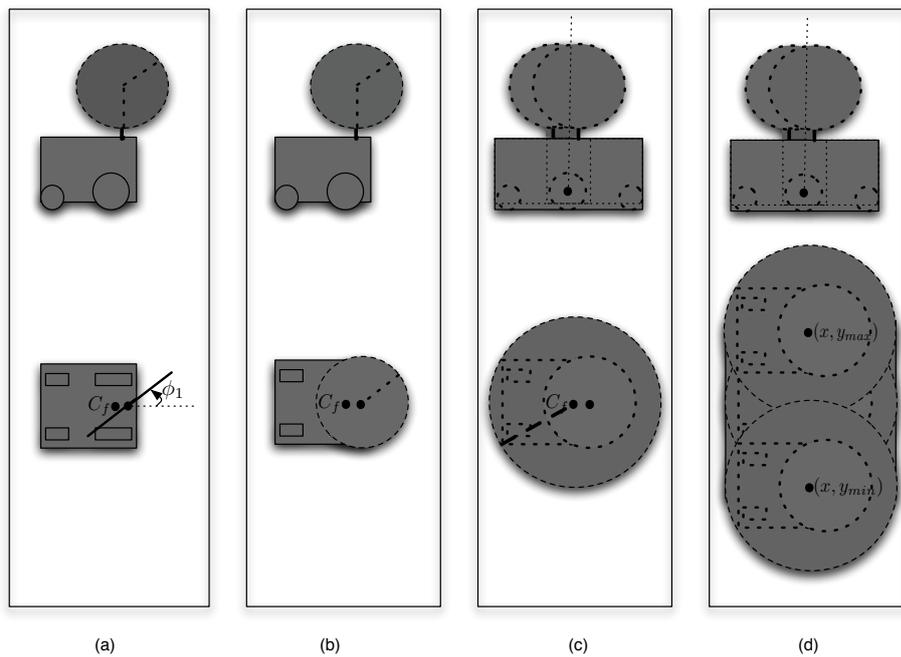


Figure 3.6: Representations of hierarchical occupancy. (a) $OA_c(c^4)$; (b) $OA_c(c^3)$; (c) $OA_c(c^2)$; (d) $OA_c(c^1)$.

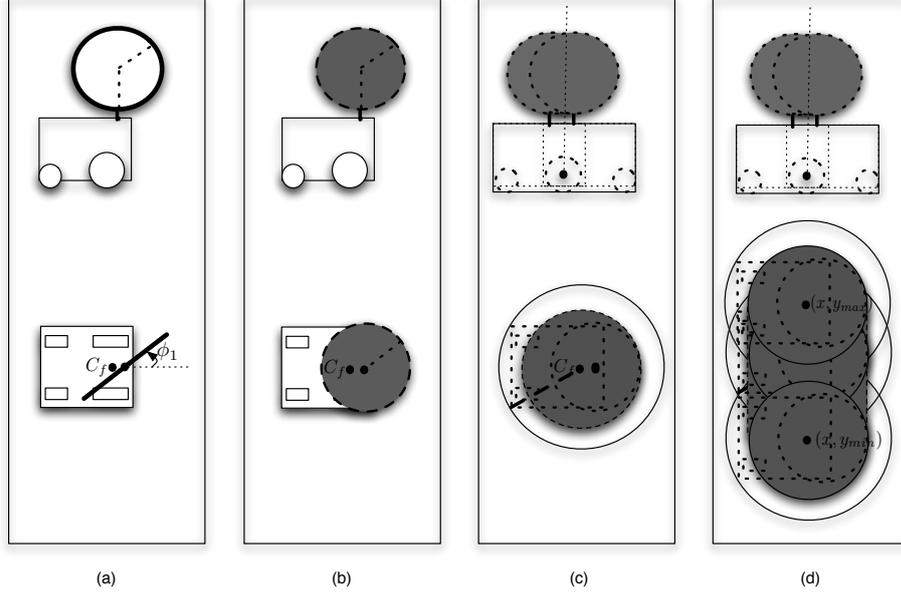


Figure 3.7: Representations of hierarchical reachability. (a) $RA_c(c^4)$; (b) $RA_c(c^3)$; (c) $RA_c(c^2)$; (d) $RA_c(c^1)$. The reachable region changes from a circle, to a shell, and finally to complex volumes.

shapes can provide significant computational savings. Note that the hierarchical representations of the combined robot bodies can be computed prior to the execution of the motion planner. This needs to be done only once for each DOF of the robot, independent of the robot's configuration. It is not repeated for each new planning problem. Moreover, in the domain of accessibility assessment, models of the kinematic structures are often available long before they are used for motion planning leaving plenty of opportunity for pre-computation.

The hierarchy is applicable not only to the kinematic configurations (nodes of the roadmap), but also can be applied to the paths connecting configurations. The motivation here is the observation of the different effects of \mathcal{A}_{base} and \mathcal{A}_{arm} discussed in the previous section. To generalize the local path to incorporate the hierarchy of states, we define a label e^r , meaning that each configuration c along edge e is associated with the same or smaller r (see Figure 3.8). The function $V(e^r)$, $RA_e(e^r)$ and $OA_e(e^r)$ can be defined in terms of the hierarchical nodes on e^r :

$$V(e^r) = true \iff \forall c^r \in e^r, V(c^r) = true \quad (3.1)$$

$$RA_e(e^r) = \bigcup_{c^r \in e^r} RA_c(c^r) \quad (3.2)$$

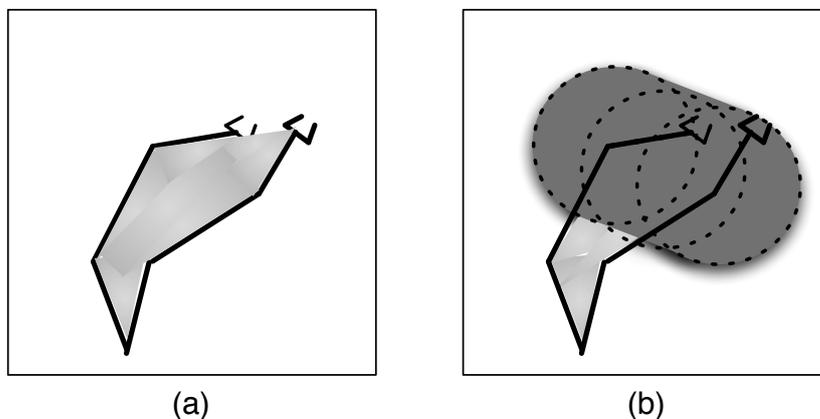


Figure 3.8: A three-link planer manipulator moves from one configuration to another one following a local path e . (a) The shaded area is the workspace that the manipulator occupies while moving along e ; (b) The shaded area is the workspace that the manipulator occupies while moving along e^2 (dark shaded area is generated by the hierarchical body).

$$OA_e(e^r) = \bigcup_{c^r \in e^r} OA_c(c^r) \quad (3.3)$$

3.2.3 Construction of the hierarchical roadmap \mathcal{R}

We now describe the main steps of the construction of the hierarchical roadmap introduced in the previous section. Nodes with large reachable areas are preferred in the domain of accessibility assessment (they establish more of the environment as being reachable for each calculation). So for each configuration c , we look for minimum values r_{min} such that $V(c^{r_{min}})$ is true, and we call r_{min} the rank of c . The procedure described in the pseudocode below tries to find a new random configuration and establish its most general representation in the hierarchy.

Algorithm 1 Node selection

```
1:  $nodeFound \leftarrow false$ 
2: while  $\neg nodeFound$  do
3:    $c \leftarrow$  a randomly chosen configuration in  $\mathcal{C}$ 
4:   for  $k \leftarrow 1$  to  $n$  do
5:     if  $V(c^k)$  then
6:        $nodeFound \leftarrow true$ 
7:       break
8:     end if
9:   end for
10: end while
11:  $N \leftarrow N \cup \{c^k\}$ 
```

In the for loop from Line 4 to Line 9, the algorithm attempts to compute the rank of the node by checking collision of the hierarchical representations. Once the minimal valid hierarchical representation is established, the configuration together with the computed rank will be added to the set of nodes N (Line 11).

Whenever a new hierarchical node is found, we select a number of candidate nodes from the current set N and try to connect the new node to each of them. In addition to the connection computation performed by the traditional local planner, we need to establish the rank (i.e. the minimal hierarchy) of the edge. For an edge e we look for the minimum dimension r_{min} such that $V(e^{r_{min}})$ is true.

The hierarchical nodes interconnection is built upon an existing local path locator and a hierarchy establisher. The local path locator returns an edge candidate, i.e. a local path that \mathcal{A} can follow from one configuration to another. Then the hierarchy establisher tries to check if the edge candidate is collision free and meanwhile establishes the edge's most general representation in the hierarchy. The process of establishing the hierarchical node interconnection is outlined below in Algorithm 2.

Algorithm 2 Connect(a^{r_1}, b^{r_2})

```
1:  $\tau \leftarrow$  the edge candidate returned by the local path locator
2: Discretize  $\tau$  into a list of configurations  $\tau' = (c_1, c_2, \dots, c_m)$ 
3:  $r_{current} \leftarrow \text{MAX}(r_1, r_2)$ 
4: for all  $c_i \in \tau'$  do
5:   for  $k \leftarrow r_{current}$  to  $n$  do
6:     if  $V(c_i^k)$  then
7:        $r_{current} \leftarrow k$ 
8:       break
9:     else
10:      exit and report failure
11:    end if
12:  end for
13: end for
14:  $E \leftarrow E \cup \{(a, b)^{r_{current}}\}$ 
```

In line 3, the hierarchy r is initialized to be the maximal value of the two ends. There is an obvious lemma according to the definition of the hierarchical edge connecting two nodes a^{r_1} and b^{r_2} :

Lemma 3.2.4. $V(e^r) = \text{true} \implies r \geq r_1 \wedge r \geq r_2$, i.e. the rank of an edge is not less than the rank of either end node of the edge.

Algorithm 2 searches over the sequence of configurations on the edge for verification and hierarchy establishment. This general approach is straightforward and easy to implement. A number of components of the algorithm are still unspecified. In particular, we need to define the mapping RA and OA between \mathcal{W} and \mathcal{R} for the computation of reachable workspace as well as for collision checking. Also the distance function between two hierarchical nodes needs to be defined.

Mapping between \mathcal{W} and \mathcal{R}

A key element of the motion planning approach is the mapping between the workspace \mathcal{W} and the roadmap \mathcal{R} . On one hand, the reachable workspace depends on the constructed roadmap. On the other hand validation of the roadmap depends on the distributions of obstacles in the workspace. Hence \mathcal{W} and \mathcal{R} are closely related. Let the workspace \mathcal{W} be represented by a uniform cell decomposition denoted by D . In terms of this representation the functions RA_c and RA_e can be defined:

Definition 3.2.1. $RA_c(c^r) = \{d \in D \mid d \text{ is reached by the end-effector of } \mathcal{A} \text{ from some configuration of } c^r\}$.

Definition 3.2.2. $RA_e(e^r) = \{d \in D \mid d \text{ is reached by the end-effector of } \mathcal{A} \text{ from some configuration of } c_i^r \text{ for } c_i^r \text{ on } e^r\}$.

Given these two functions the reachable workspace can be computed by the nodes and edges of the connected component \mathcal{R}' ($\mathcal{R}' \subseteq \mathcal{R}$) that contains c_{init} :

$$\mathcal{W}_{reach} = \left(\bigcup_{c^r \in \mathcal{R}'} RA_c(c^r) \right) \cup \left(\bigcup_{e^r \in \mathcal{R}'} RA_e(e^r) \right) \quad (3.4)$$

RA_c and RA_e are many-to-one mappings. Define the complete inverse maps RA_c^{-1} and RA_e^{-1} as follows:

Definition 3.2.3. $RA_c^{-1}(d) = \{c \in N \mid RA(c) \cap d \neq \emptyset\}$.

Definition 3.2.4. $RA_e^{-1}(d) = \{e^r \in E \mid RA(e^r) \cap d \neq \emptyset\}$.

In practice, $RA_c^{-1}(d)$ can store just one node that maps to d and $RA_e^{-1}(d)$ can store just one edge that maps to d . Given an arbitrary cell d in D , RA_c^{-1} can identify one node in \mathcal{R} for which the robot's end-effector reaches d . We are not only interested in the state of the end-effector in D , but also that of the entire body of the robot, from which collisions can be detected. Define two additional mapping OA for this purpose:

Definition 3.2.5. $OA_c(c^r) = \{d \in D \mid d \text{ is occupied by } \mathcal{A} \text{ for some configuration of } c^r\}$.

Definition 3.2.6. $OA_e(e^r) = \{d \in D \mid d \text{ is occupied by } \mathcal{A} \text{ for some configuration of } c_i^r \text{ for } c_i^r \text{ on } e^r\}$.

Clearly $V(c^r) \iff OA_c(c^r) \cap \mathcal{W}_{free} = \emptyset$, and $V(e^r) \iff OA_e(e^r) \cap \mathcal{W}_{free} = \emptyset$.

We would like to know the impact of environment modifications on \mathcal{R} . Assume \mathcal{R} is first computed in a minimalist environment, i.e. an environment within which all modifiable obstacles are missing. If obstacles are added to \mathcal{W} then some nodes and edges in \mathcal{R} may become invalid. The inverse mappings OA_c^{-1} and OA_e^{-1} are needed for this purpose, which are defined as follows:

Definition 3.2.7. $OA_c^{-1}(d) = \{c^r \in \mathcal{R} \mid \xi(c^r) \cap d \neq \emptyset\}$.

Definition 3.2.8. $OA_e^{-1}(d) = \{e^r \in \mathcal{R} \mid \xi(c_i^r) \cap d \neq \emptyset \text{ for some } c_i^r \text{ on } e^r\}$.

Distance function

The distance function is used to select and sort candidate neighbors of each node. For a pair $(c_1^{r_1}, c_2^{r_2})$ the distance function should reflect the likelihood that the local planner will compute a feasible path between these two configurations. There can be many ways to define this function. Here, by taking the hierarchies of nodes into consideration, the function is defined by using the reachability function RA_c of the two nodes $c_1^{r_1}$ and $c_2^{r_2}$ as follows:

$$d(c_1^{r_1}, c_2^{r_2}) = \min \|x_1 - x_2\| \text{ for some } x_1 \in RA_c(c_1^{r_1}), x_2 \in RA_c(c_2^{r_2}), \quad (3.5)$$

where $\|x_1 - x_2\|$ is the Euclidean distance between points x_1 and x_2 . In this definition $d(c_1^{r_1}, c_2^{r_2})$ is the minimum distance between the end-effector at two generalized configurations $c_1^{r_1}$ and $c_2^{r_2}$.

3.2.4 Roadmap Enhancement

One of the difficulties with the traditional PRM algorithm is placing samples in difficult regions in \mathcal{C}_{free} . This leads to an additional enhancement phase in the algorithm whose goal is to increase the connectivity of the roadmap. This can be achieved by placing more samples around the existing nodes that are likely to lie in such narrow passages in the \mathcal{C}_{free} . Since our roadmap is constructed for workspace estimation, connecting the entire \mathcal{C}_{free} is not the only goal for our enhancement phase. Furthermore, we are interested in regions in $\mathcal{W}_{free} \setminus RA(\mathcal{R})$, which have not been established yet. The purpose of the enhancement here is to add more nodes in a way that will increase reachable workspace coverage.

The heuristic proposed here identifies areas for enhancement via reachable workspace estimation. First identify undiscovered regions $\mathcal{W}_{free} \setminus RA(\mathcal{R})$ that are of interest. This can be done either automatically or by users. This is not a complex problem since the workspace \mathcal{W} is known and is usually 2D or 3D. For each undiscovered region \mathcal{W}_x we identify nodes in \mathcal{R} that are close to \mathcal{W}_x . A distance function is needed for this purpose. Different from the distance function (3.5) which takes two nodes in \mathcal{C}_{free} as inputs, this distance function deals with $W_x \subseteq \mathcal{W}_{free}$ and $c^r \in \mathcal{C}_{free}$ and is defined as follows:

$$\hat{d}(W_x, c^r) = \min \|x_1 - x_2\| \text{ for some } x_1 \in W_x, x_2 \in RA_c(c^r). \quad (3.6)$$

Using this function, $N(W_x)$, the set of K -nearest nodes of W_x in \mathcal{R} can be found. K is a user-defined constant. For each node c^r in $N(W_x)$ additional samples are created around c^r . Define the neighborhood $B(c^r)$ from which new samples are chosen as follows:

$$B(c^r) = [j_1 - w_x, j_1 + w_x] \times [j_2 - w_x, j_2 + w_x] \dots \times [j_r - w_x, j_r + w_x] \times \mathcal{D}_{r+1} \dots \mathcal{D}_n, \quad (3.7)$$

where w_x is a predefined parameter associated with the x -th joint. Recall the definition of c^r , which is the set of configurations possible with joints $1 \dots r$ having specific values but joints

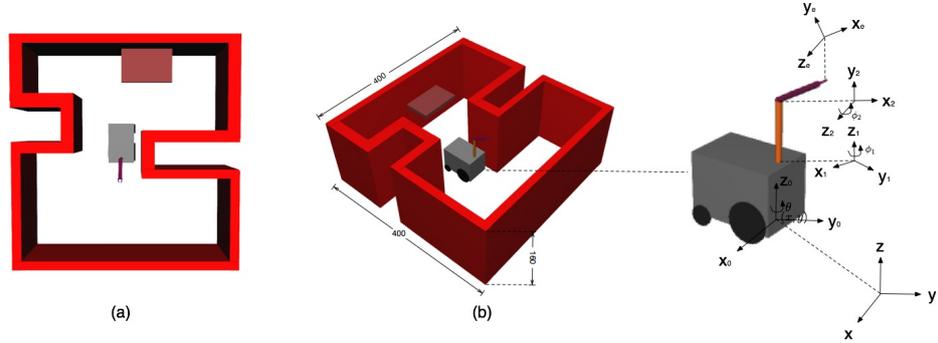


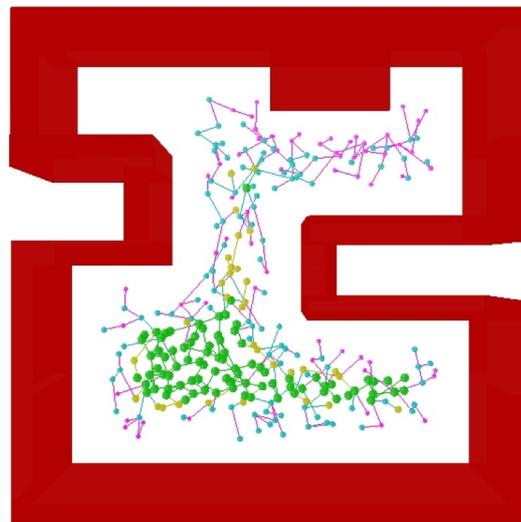
Figure 3.9: A mobile manipulator in a 3D toy environment. The toy environment contains open regions and a narrow passage. There is a shelf on the wall whose reachability is of interest. (a) Top view; (b) Bird view. The frame assignment of \mathcal{A}_{base} and each link of \mathcal{A}_{arm} is illustrated.

$r + 1 \dots n$ being free. Since only the first r joints of c^r have specific values its neighborhood can be decided by considering these joints. In order to add new samples, the first r values can be chosen by a random walk within certain distance from the corresponding values of c^r and the rest are chosen randomly from within their domains. The choice of the distance parameter w_x depends on the corresponding joint j_x .

3.2.5 Example

This section provides an example that illustrates the hierarchical strategy described in the previous section. The environmental setup is shown in Figure 3.9. The environment is in 3D (discretized into $64 \times 64 \times 25$ cells) and the kinematic structure is a mobile manipulator \mathcal{A} (\mathcal{A} consists of \mathcal{A}_{base} and a 2-link \mathcal{A}_{arm}). First assume everything in the environment is static. Figure 3.10 shows the constructed hierarchical roadmap of \mathcal{A} , where the hierarchies of the corresponding nodes and edges are indicated by different colors. As can be seen in this example, the hierarchies of the nodes tend to be lower in open regions and higher in narrow regions.

Figure 3.11 provides details of the execution of the hierarchical PRM on this example. Initially the roadmap contains only one node that represents the initial configuration of \mathcal{A} . The rank of each randomly generated node is determined by looking for the most general occupancy representation of \mathcal{A} that does not collide with any obstacles. Similarly the rank of each edge is determined by looking for the most general occupancy representation of \mathcal{A} along the edge that does not collide with any obstacles. The top row shows tests for a randomly generated node. c^2 and c^3 generate collisions while c^4 did not, so this specific node is classified in c^4 . The construction of the roadmap can be incremental. The lower



(a) Environment

5	4	3	2

(b) Occupancy colors

Figure 3.10: A sample representation of the HPRM description of the environment. Nodes and edges are colored based on their rank. The table shows hierarchical occupancy representations of \mathcal{A} . Occupancy includes the base of the device.

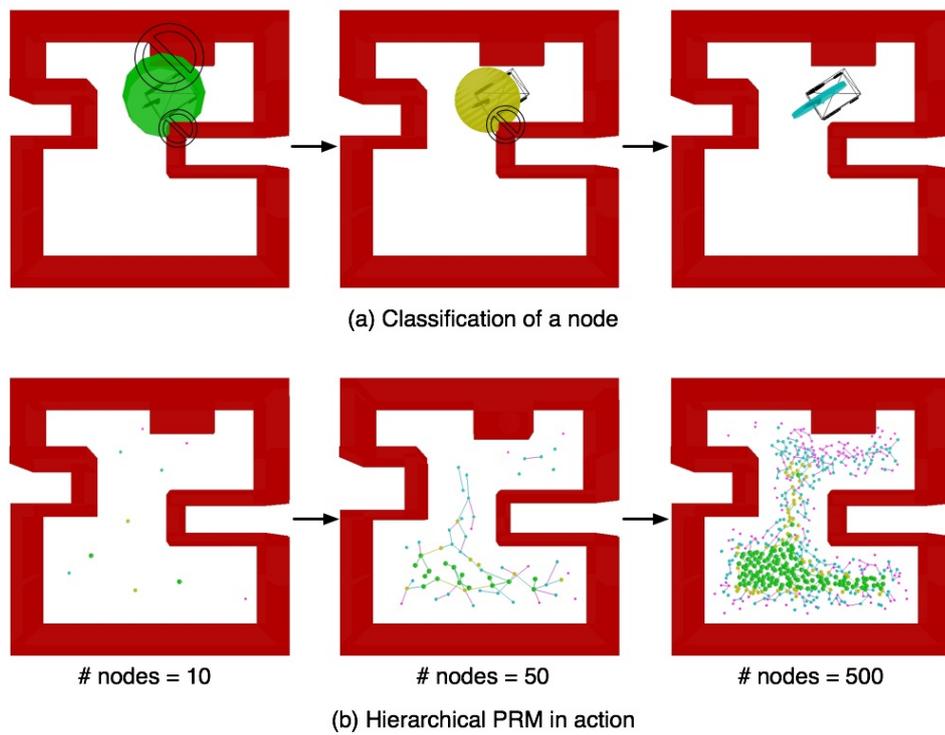


Figure 3.11: Construction of the hierarchical roadmap \mathcal{R} . Upper row shows the generation and classification of one node. The rank of a node c^r is calculated by checking the hierarchical occupancy representation of \mathcal{A} . Since c^2 and c^3 both introduced collision but c^4 does not, the rank of the node is set to be 4. Lower row shows the hierarchical PRM in operation. The coverage and connectivity of \mathcal{R} increases as more nodes are added.

row shows incremental changes in the roadmap. As more nodes are added both coverage and connectivity of the roadmap increases.

Figure 3.12 shows the result reachable workspace \mathcal{W}_{reach} that is mapped from the constructed hierarchical roadmap. \mathcal{W}_{reach} can be viewed in different layers in 3D.

There are empty regions in the environment that are not included in \mathcal{W}_{reach} which may be reachable and the roadmap can be enhanced to discover them. Figure 3.13 shows an example of such enhancement. The user selects a potentially unreachable region and the HPRM attempts to link these seeds to the connected graph. It is possible that the regions that are actually reachable by \mathcal{A} may not be included in \mathcal{W}_{reach} even after the enhancement. But such an enhancement still provides insight into the reachability analysis. In the domain of accessibility assessment, such regions can be considered as “difficult” or “unreachable” since they are difficult or impossible to reach.

3.3 Environmental planner

The final task in accessibility assessment is to explore the impact of changes of the set of obstacles on the roadmap and therefore the size of reachable workspace. We do not want to re-compute the roadmap each time we change the position of an obstacle. We need to compare the sizes of reachable workspaces for different obstacle placements, and find the placement \mathcal{P}^{opt} that provides the largest reachable workspace. One ‘optimal’ solution is to remove every object from the environment but in practice this trivial solution should be avoided. (It is not desirable to find optimally accessible environments that contain no useful furniture or objects.) Theoretically the optimal non-trivial solution could be obtained via a brute-force search by computing the workspaces for all possible placements of obstacles. However, such an approach is very expensive computationally and impractical for large numbers of obstacles and possible placements. Here the solution is approximated via a greedy algorithm.

3.3.1 A greedy algorithm

Under an assumption that the potential placements of modifiable obstacles are independent of each other, the heuristic algorithm looks for local optimal placement for each modifiable obstacle separately. The idea is to modify one obstacle at a time so as to obtain a larger workspace. There often exists some critical obstacle in the environment that limits \mathcal{A} ’s reachability. Motivated by this fact we identify the most critical obstacle first and then find its best placement. This process is then repeated for the remaining obstacles. Although the reachable workspace of \mathcal{A} always becomes larger (or at least stay the same) after removing an obstacle, the trivial solution of removing all obstacles is avoided by providing a modification cost for removal that is much larger than the cost of moving objects to “real”

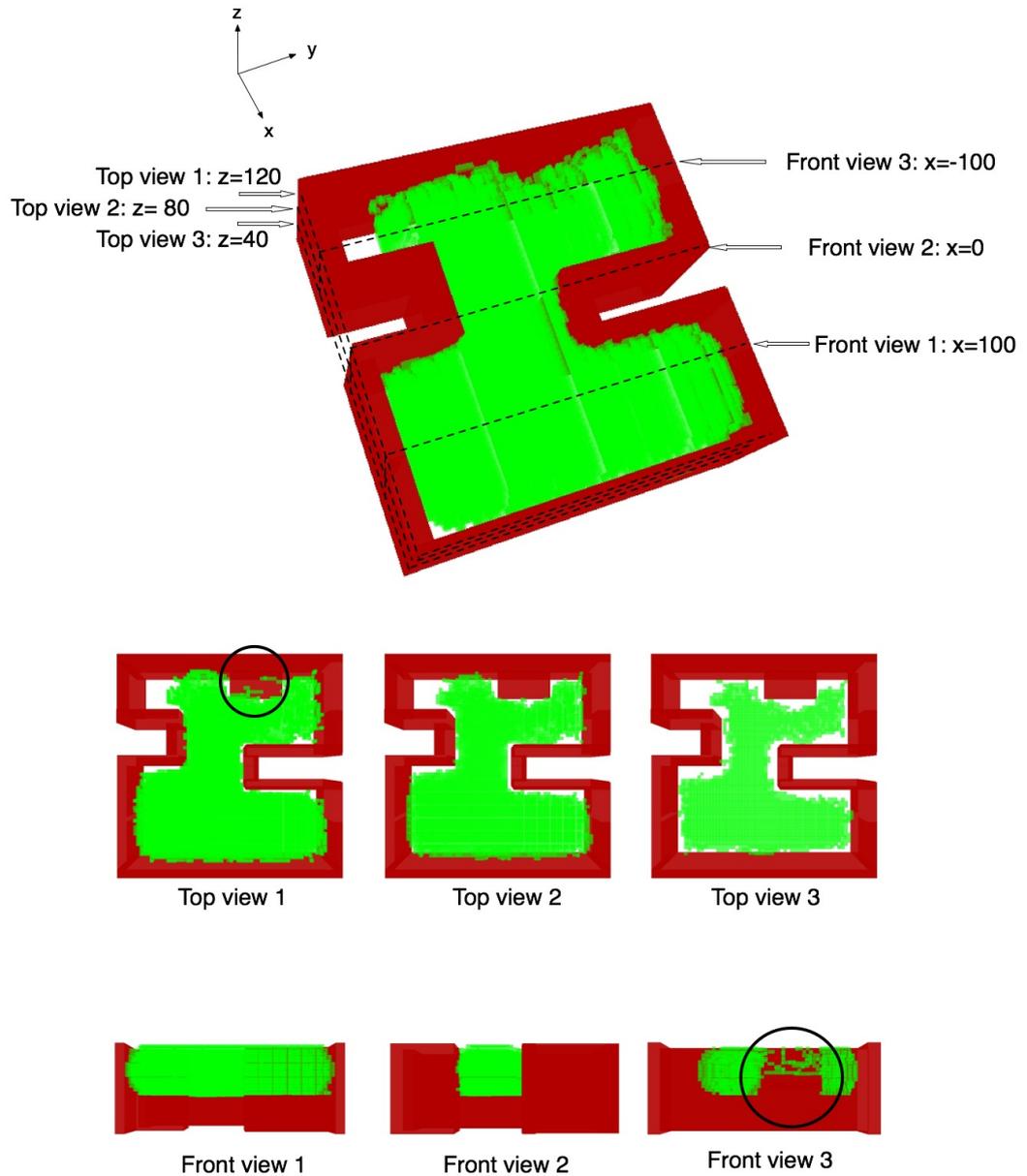


Figure 3.12: The estimation of the 3D reachable workspace \mathcal{W}_{reach} (green area) showed in different layers of top views and front views. The area above the shelf is reachable but the area under it is not.

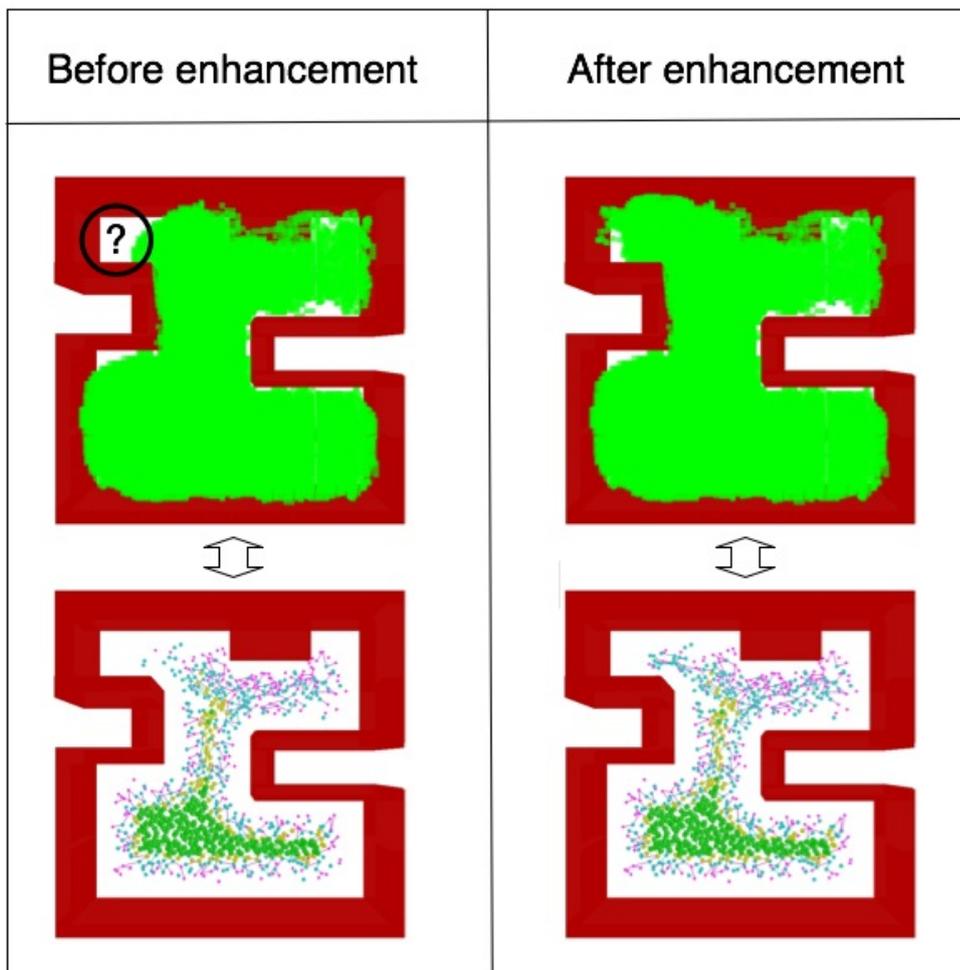


Figure 3.13: The enhancement phase. Before enhancement the left upper corner is not included in \mathcal{W}_{reach} , so more samples are added to the roadmap by the user in the suspected area. After enhancement \mathcal{W}_{reach} grows to include this volume.

positions in the environment. Hence solutions with minimal obstacles will be avoided as the cost of such solutions will be out of bounds (according to the definition of the cost function). Let $P_{rep}(\mathcal{P}, i, j)$ be a function that returns a placement \mathcal{P}' that is obtained from \mathcal{P} by moving obstacle m_i to position pp_i^j . The algorithm tries to move/remove modifiable obstacles from the initial placement \mathcal{P}^1 within the given cost bound towards the largest reachable workspace as follows:

Algorithm 3 Greedy environmental planning

```

1:  $\sigma \leftarrow$  user cost bound
2:  $\mathcal{P}^{opt} \leftarrow \mathcal{P}^1$ 
3:  $M \leftarrow \{1, 2, \dots, h\}$ 
4: while  $\sigma > 0 \wedge M \neq \{\}$  do
5:   (1) Choose  $best \in M$  s.t.
6:      $|\mathcal{W}_{reach}(P_{rep}(\mathcal{P}^{opt}, best, 0))|$  is maximum of  $\forall i \in M, |\mathcal{W}_{reach}(P_{rep}(\mathcal{P}^{opt}, i, 0))|$ 
7:      $M \leftarrow M - \{best\}$ 
8:   (2)  $N \leftarrow \{0 \dots k_{best}\}$ 
9:     Choose  $q \in N$  s.t.
10:    (a)  $|\mathcal{W}_{reach}(P_{rep}(\mathcal{P}^{opt}, best, q))|$  is maximum of
11:       $\forall i \in N, |\mathcal{W}_{reach}(P_{rep}(\mathcal{P}^{opt}, best, i))|$ 
12:    (b)  $\tau(pp_{best}^q) \leq \sigma$ 
13:     $\mathcal{P}^{opt} \leftarrow P_{rep}(\mathcal{P}^{opt}, best, q)$ 
14:     $\sigma \leftarrow \sigma - \tau(pp_{best}^q)$ 
15: end while
16: return  $\mathcal{P}^{opt}$ 

```

In the course of the planning process, the modifiable obstacles are divided into two sets: I. the obstacles whose best placement is found (their best placements are stored in \mathcal{P}^{opt}) and II. the indices of the remaining obstacles (stored in M). Initially M is the set of indices of all the modifiable obstacles \mathcal{M} , and \mathcal{P}^{opt} is the initial placement \mathcal{P}^1 . The modification cost bound σ is initialized as a user selected value. From then onwards the algorithm performs two steps repeatedly.

1. Find the obstacle $best$ in the remaining set M whose removal has the most impact on \mathcal{A} 's reachable workspace and remove it from M
2. Find the best placement for obstacle $best$ among all its potential placements that does not exceed the current cost bound and provides the largest reachable workspace. Note

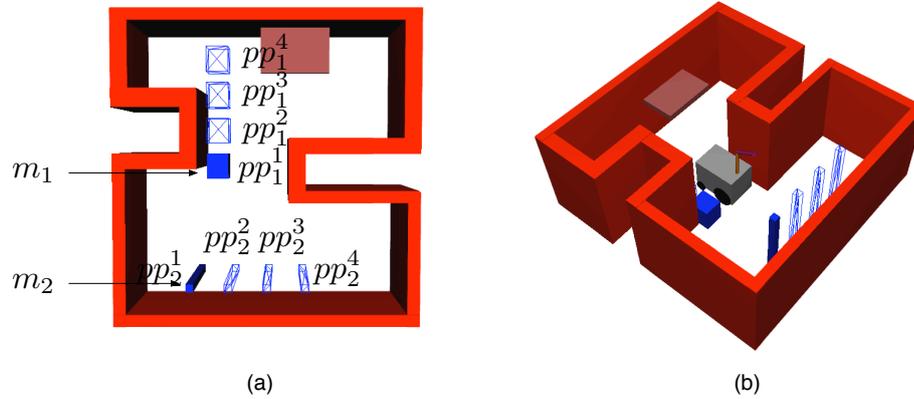


Figure 3.14: An example environment has two modifiable obstacles. Each obstacle can be moved to three other positions or removed from the scene.

that provided it is cost valid then this includes removing the modifiable obstacle from the workspace.

The process is repeated until set M is empty (all the modifiable obstacles have been examined) or the cost bound σ is zero (no more modification is allowed).

In the practice of accessibility assessment there may be other termination conditions in addition to the cost bound. For example, sometimes it is not necessary to know the entire reachable workspace but rather only the reachability of specific important elements in the environment such as light switches and the water faucet. In this case the algorithm can be terminated when all the important elements are included in \mathcal{W}_{reach} .

If the motion planner was invoked every time, this computation would be prohibitively expensive. However, when only small changes are made to the environment it is often the case that most portions of the roadmap in configuration space remain unchanged. The roadmap computed given an environment where all the modifiable obstacles are placed in all of their possible configurations is always valid. Therefore we maintain this pessimist roadmap and use the additional portion computed in the previous iteration as a starting point when constructing the new roadmap.

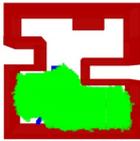
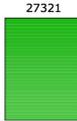
3.3.2 Example

Figure 3.14 shows an example of a toy environment containing two modifiable obstacles. Each modifiable obstacle has five potential placements including one in which the obstacle is missing. A modification cost function τ is needed to perform the operations which should meet the following criteria:

State	Initial state	Iteration 1					
		Step 1: Identifying best obstacle to modify			Step 2: Identifying best placement of obstacle 1		
\mathcal{W}_{reach}	 (pp_1^1, pp_2^1)	 (pp_1^0, pp_2^1)	 (pp_1^1, pp_2^0)	 (pp_1^2, pp_2^1) $\tau(pp_1^2) < \sigma$	 (pp_1^3, pp_2^1) $\tau(pp_1^3) < \sigma$	 (pp_1^4, pp_2^1) $\tau(pp_1^4) < \sigma$	
$ \mathcal{W}_{reach} $	 16062 1,1	 27321 0,1	 17239 1,0	 17694 2,1	 24153 3,1	 25613 4,1	
Outcome		Obstacle 1 identified as best obstacle to move			Placement 4 identified as best placement for obstacle 1		

State	Iteration 2		
	Identifying best placement of obstacle 2		
\mathcal{W}_{reach}	 (pp_1^4, pp_2^2) $\tau(pp_2^2) < \sigma$	 (pp_1^4, pp_2^3) $\tau(pp_2^3) < \sigma$	 (pp_1^4, pp_2^4) $\tau(pp_2^4) < \sigma$
$ \mathcal{W}_{reach} $	 25843 4,2	 25852 4,3	 22047 4,4
Outcome	Placement 3 identified as best placement for obstacle 2		

Figure 3.15: Illustration of the execution of the environmental planner in case 1 when no obstacle can be removed. Iteration 1: in Step 1 m_1 is selected to be the *best*; in Step 2 it is moved to position 2, 3, 4 and the size of \mathcal{W}_{reach} is maximized in the case of position 4. So fix obstacle 1 to pp_1^4 and continue to the next iteration. Iteration 2: m_2 is moved to position 2, 3, 4 and $|\mathcal{W}_{reach}|$ is maximized in the case of position 3 (yellow column). The “best” environmental configuration $\mathcal{P}^{opt} = (pp_1^4, pp_2^3)$ is found.

State	Initial state	Iteration 1	
		Identifying best obstacle to modify	
\mathcal{W}_{reach}	 (pp_1^1, pp_2^1)	 (pp_1^0, pp_2^1)	 (pp_1^1, pp_2^0)
$ \mathcal{W}_{reach} $	 16062 1,1	 27321 0,1	 17239 1,0
Outcome		Obstacle 1 identified as best obstacle to modify, and placement 0 is its best placement	

State	Iteration 2		
	Identifying best placement of obstacle 2		
\mathcal{W}_{reach}	 (pp_1^0, pp_2^2) $\tau(pp_2^2) < \sigma$	 (pp_1^0, pp_2^3) $\tau(pp_2^3) < \sigma$	 (pp_1^0, pp_2^4) $\tau(pp_2^4) < \sigma$
$ \mathcal{W}_{reach} $	 27551 0,2	 27560 0,3	 23755 0,4
Outcome	Placement 3 identified as best placement for obstacle 2		

Figure 3.16: Illustration of the steps of the greedy environmental planner in case 2 when at most one obstacle can be removed. Iteration 1: in Step 1 obstacle 1 is selected to be the *best*; in Step 2 it is removed (i.e. pp_1^0) since it is within the cost bound and definitely provides no less reachable workspace than other modifications. So continue to the next iteration. Iteration 2: obstacle 2 is moved to position 2, 3, 4 and $|\mathcal{W}_{reach}|$ is maximized in the case of position 3 (blue column). The “best” environmental configuration $\mathcal{P}^{opt} = (pp_1^0, pp_2^3)$ is found.

1. $\forall i \in [1..h], \zeta(pp_i^1) = 0$;
2. $\forall i \in [1..h], \forall j \in [1..k_i], \zeta(pp_i^0) > \zeta(pp_i^j)$.

There can be many ways to define a τ that meets the above requirements. The idea here is to distinguish the removal of the obstacle with other possible modifications. Assume that the cost of removing one obstacle is more expensive than moving all the obstacles in \mathcal{M} . If there are h modifiable obstacles in total, then define the modification cost function as follows:

$$\begin{aligned}
\tau(pp_i^1) &= 0 \\
\tau(pp_i^j) &= 1 \quad (j = 2, 3, \dots) \\
\tau(pp_i^0) &= h + 1
\end{aligned} \tag{3.8}$$

All the modifications except removal have the same cost (1), and the removal of each obstacle has the same cost (3 in this specific example). It is obvious that the reachable workspace of \mathcal{A} always becomes larger (or at least stay the same) after removing an obstacle. However, we cannot remove an arbitrary number of obstacles because otherwise the cost would be out of bound. The cost bound σ is used to indicate how many obstacles are allowed to be removed. In this particular example, we consider the following two special cases:

1. $\sigma = 2$: Both obstacles are allowed to be moved around but no obstacle is allowed to be removed from the scene.
2. $\sigma = 4$: Both obstacles are allowed to be moved around but at most one obstacle is allowed to be removed from the scene.

The planning processes corresponding to the two cases are shown in Figure 3.15 and Figure 3.16. The two main steps of each iteration in the searching process are outlined. The \mathcal{W}_{reach} computed in the intermediate steps are also illustrated along with the corresponding sizes measured in terms of number of cells in \mathcal{W} . In this particular example although there are 5×5 possible environmental configurations, this greedy approach manages to solve the problem within 9 computations of the reachable workspace. Figure 3.17 shows the size of reachable workspaces for all the 25 possible environmental configurations. The modifications suggested by the environmental planner for case 1 and 2 have shown considerable improvement over the initial obstacle placement. As the cost bound becomes less restricted the planner leads the result toward the optimal solution.

3.4 Summary

This chapter presented a hierarchical PRM for accessibility assessment. Unlike traditional PRMs that treat each DOF equally, we order the DOF's of the kinematic structure and consider a hierarchical approach to the planning task. Considering the characteristic of the

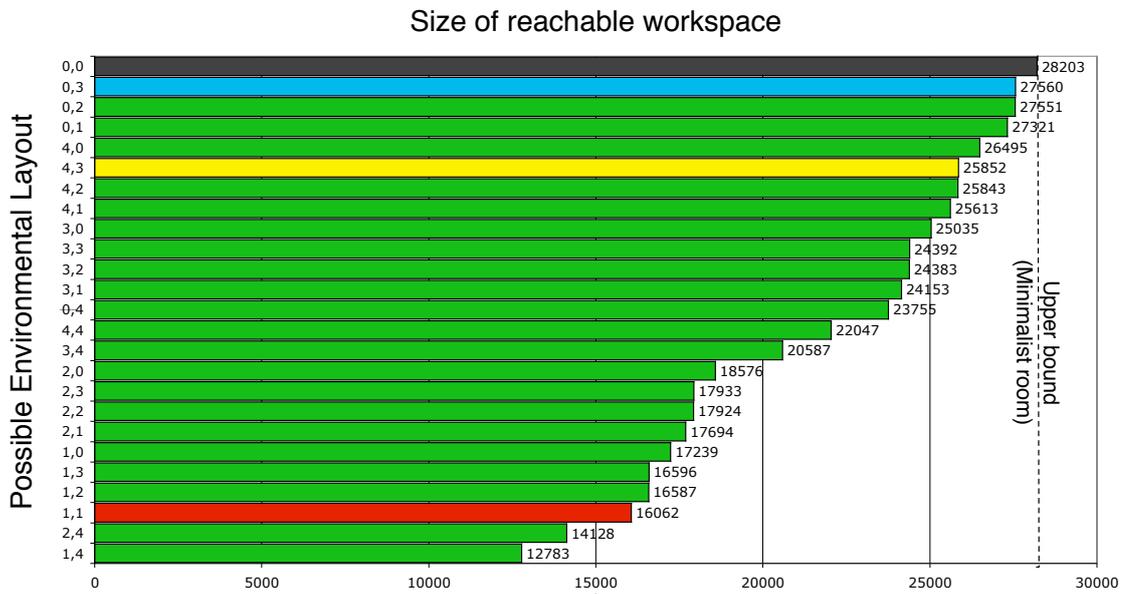


Figure 3.17: The size of reachable workspace (x-axis) computed for the 25 possible layouts of the environment. The vertical axis denotes the layout of the environment that specifies the placement for each modifiable obstacle (x, y stands for (pp_1^x, pp_2^y)). The horizontal axis denotes the size of the reachable workspace measured in terms of numbers of cells in \mathcal{W} . The upper bound is the minimalist room (the grey bar). The red bar corresponds to the initial placement of modifiable obstacles. The yellow bar corresponds to the result from case 1. The blue bar corresponds to the result from case 2.

accessibility assessment problem, this hierarchy exploration improves the planning process through two critical computations. First, it accelerates collision detection in open regions by approximating the robot using a conservative occupancy analysis. Validation of the configuration begins by doing fast tests on simple representations and only progresses to more accurate (and more expensive) evaluations as necessary. As for reachability analysis, because randomness is involved it is hardly possible to estimate the size of the entire reachable workspace by mapping from the PRMs within reasonable time. However, by iteratively computing the maximal reachable workspaces from each node and edge our hierarchical PRM can be more effective in the computation process than traditional PRMs.

This hierarchical PRM allows fast re-planning on related environments, but the direct invocation of the motion planner does not solve the problem of accessibility assessment completely. Optimization of the environmental layout is a complex problem considering the superset of all the possible obstacle placements. Our environmental planner searches the superset of possible placements in a “greedy” way. The modification cost bound is introduced to avoid the trivial solution where all obstacles are removed. The greedy algorithm iteratively manipulates the placement of each modifiable obstacle to obtain a locally optimal placement. At each iteration, it selects the next obstacle to be modified with the criteria that its removal can have the most significant impact on the reachable workspace and then searches for its best placement within a given cost bound.

Chapter 4

Assessing accessibility

This chapter presents an experimental application of the methodology for accessibility assessment presented in the previous chapter. We evaluate the methodology using a wheelchair robot model and a typical living environment containing static and modifiable objects (Figure 4.1). As the algorithm contains a motion planner and an environmental planner, tests and evaluations for these two planners were also performed separately. The goal of this testing is two-fold (i) to demonstrate the motion planner’s ability to estimate the reachable workspace, and (ii) to demonstrate the environmental planner’s ability to suggest proper environmental modifications.

4.1 Experimental setup

To perform the experiments of our algorithm we require a kinematic model of the wheelchair and user and an environmental model. Chapter 2 provided descriptions of existing methods to acquire and to build these models. The models used in the experiments are described here.

4.1.1 Kinematic model

In the experiments, the kinematic model \mathcal{A} is developed based on the Playbot Robotic Wheelchair[67] shown in Figure 4.2. The wheelchair base \mathcal{A}_{base} is powered by the front wheels using a differential drive mechanism and drive its powered wheels both forward and backward. The attached kinematic chain \mathcal{A}_{arm} has four revolute joints with no limit on the angles that these joints can make. This is a 7 DOF vehicle. The configuration of \mathcal{A} is written as an ordered vector $c = (x, y, \theta, \phi_1, \phi_2, \phi_3, \phi_4)$, where (x, y) is the position of the center of the front wheels, θ is the angle that the vehicle’s main axis makes with the positive x-axis, and $\phi_1 \dots \phi_4$ are the corresponding angles of each link of \mathcal{A}_{arm} .

4.1.2 Environment

The environment assessed for accessibility is a two-bedroom suite in the Pond Road residence at York University (see Figure 4.1). The Pond Road residence was opened in September 2004 and is home to approximately 430 undergraduate students. Among the 14 student residences at Keele campus the Pond Road residence is the newest one.

Figure 4.3 shows the 3D model of this environment and the furniture is represented by simple polygon structure. Assume the doors are absent and there are four modifiable obstacles in the scene each of which has more than one possible placement. Following the



(a)



(b)



(c)



(d)



(e)



(f)

Figure 4.1: (a) The building of the Pond Road Residence at York University. (b) The architectural floor plan of a standard two bedroom suite in the Pond Road Residence. (c-f) Photographs of the suite taken from different positions.

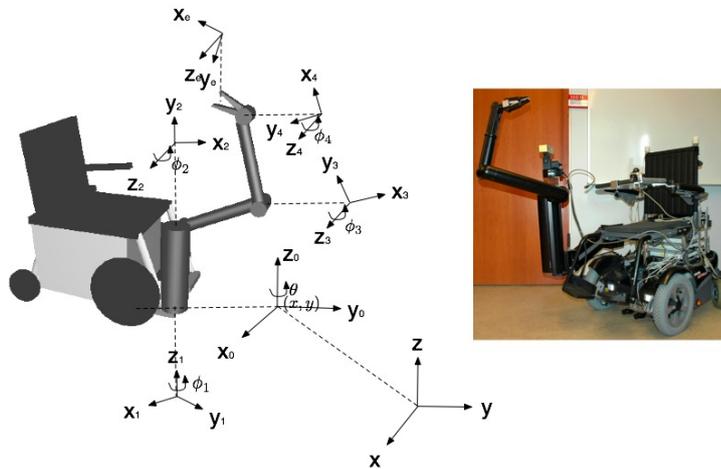


Figure 4.2: The Playbot Robotic Wheelchair consists of a wheelchair-like base under differential drive and an attached manipulator.

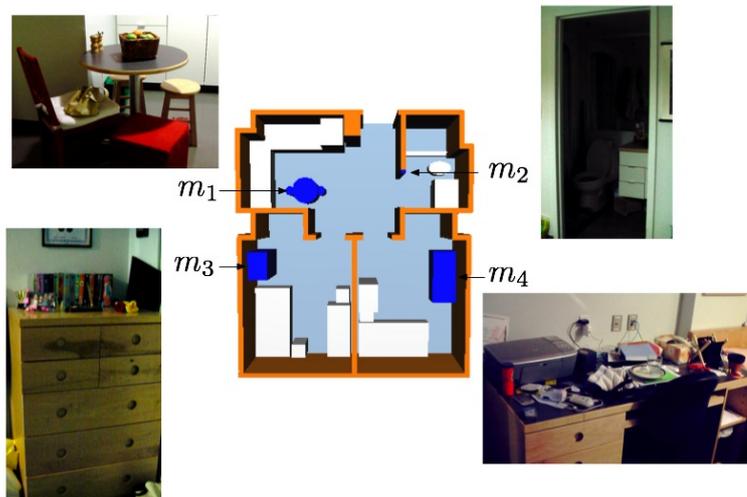


Figure 4.3: A 3D model created based on the standard suite in the Pond Road Residence. There are four modifiable obstacles in the scene: m_1 (dining table), m_2 (washroom door width), m_3 (drawers) and m_4 (desk). The rest of the scene is static.

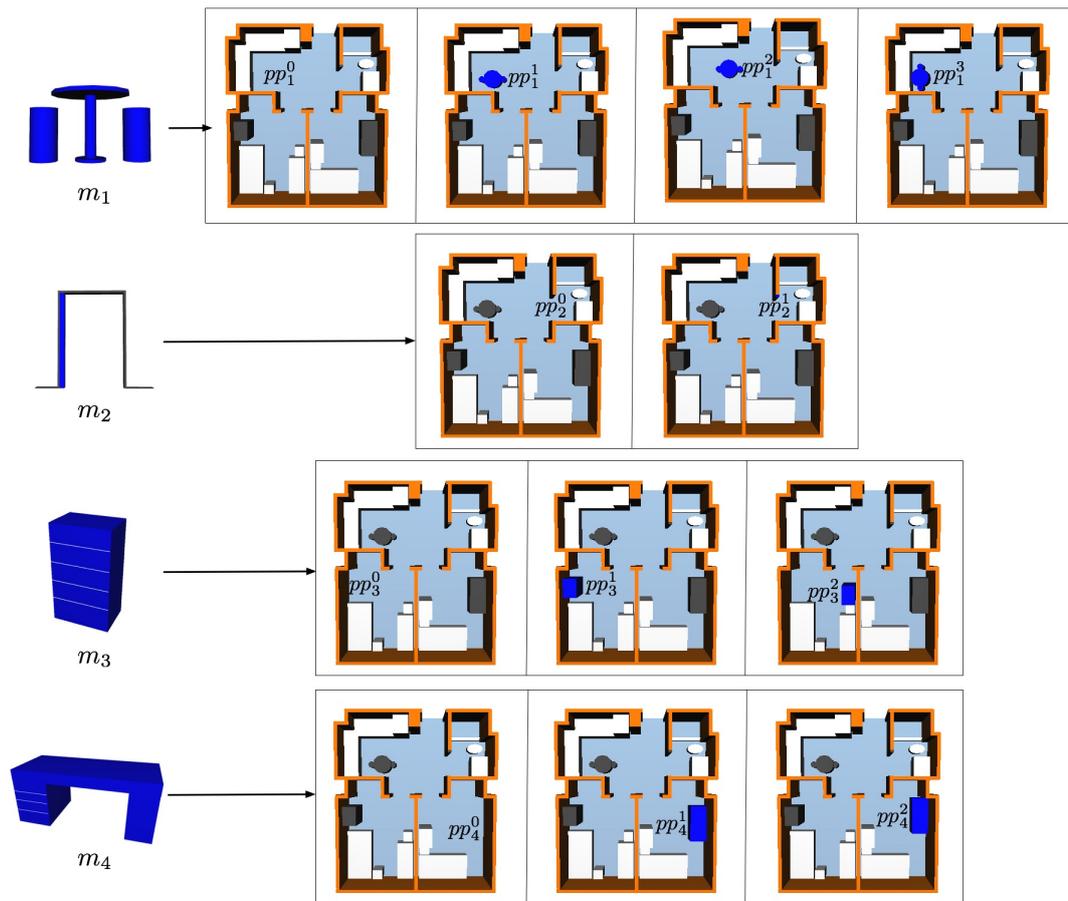


Figure 4.4: The potential placements of each modifiable obstacle m_i . Row 1: there are four possible placements of the dining table m_1 . Row 2: the door width of the washroom can be widened by removing m_2 . Row 3: the closet m_3 can be missing or placed in either side of the bedroom. Row 4: the desk m_4 can be missing or moved slightly aside.

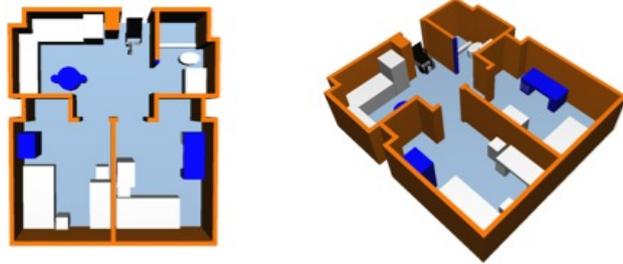


Figure 4.5: \mathcal{A} is placed initially at the entrance of the residence .

notation introduced in Chapter 3, m_i denotes a modifiable obstacle. In this specific example m_1 represents the dining table, m_2 represents part of the wall of the washroom, by removing which the door width of the washroom can be increased, m_3 represents a closet in one of the bedrooms and m_4 represents a desk in the other bedroom.

Each modifiable obstacle m_i has several potential placements $\{pp_i^0, pp_i^1, \dots\}$ among which pp_i^0 denotes the case that m_i is missing, pp_i^1 is m_i 's initial placement, and the rest are other predefined placements. Figure 4.4 shows the potential placements of each m_i in the environment. There are 72 combinations of possible modifiable obstacle placements ($4 \times 2 \times 3 \times 3$).

4.1.3 Implementation details

Choosing c_{init}

According to its definition, the size of the computed \mathcal{W}_{reach} depends the part of the roadmap that is connected to the initial configuration c_{init} of \mathcal{A} . Here, let the wheelchair robot be placed initially at the main entrance of the residence as shown in Figure 4.5. This is a reasonable assumption – the front door of the unit must be accessible.

Local planner

There is tradeoff between completeness and efficiency in the choice of local planner and a fast and deterministic local planner is commonly preferable. Concerning the kinematic model of the wheelchair and its user, we choose a local planner that can be applied generally to a mobile vehicle with a manipulator. The local planner is divided into two parts, planning for the base and planning for the manipulator. A simple straight line local planner for the manipulator which has been widely used in the PRMs for holonomic robots [29] was chosen. The method connects two given configurations by a straight line segment in configuration space and checks this line segment for collision in the workspace.

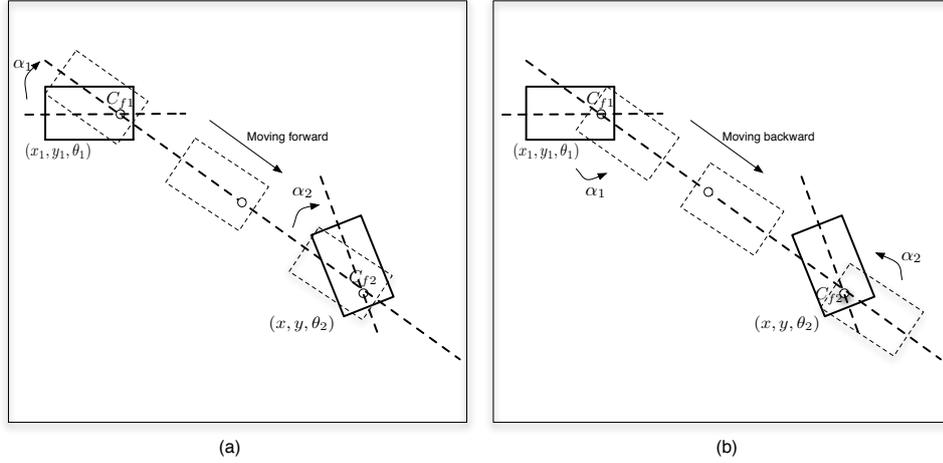


Figure 4.6: Two possible paths of the mobile base moving from (x_1, y_1, θ_1) to (x_2, y_2, θ_2) under differential drive. The moving distance along the straight line segment is the same for the two paths. However, (a) has a smaller turning angle $|\alpha_1| + |\alpha_2|$ than (b) so (a) is preferable.

Planning for the mobile base is more complex due to the existence of non-holonomic constraints. Under differential drive the mobile base can move in circles and straight lines. A simple and deterministic planner was implemented as a concatenation of an extreme rotation, a straight line and another extreme rotation (Figure 4.6). Let C_f denote the center of the front wheels of the vehicle. For the vehicle to move from (x_1, y_1, θ_1) to (x_2, y_2, θ_2) , it first rotates at C_{f1} till it is parallel to line $C_{f1}C_{f2}$, then moves along $C_{f1}C_{f2}$ to C_{f2} , and at last rotates to the desired orientation θ_2 . There are eight possible such paths connecting the two configurations (see the ALA strategy described in Chapter 2). The local planner should return the same path from (x_1, y_1, θ_1) to (x_2, y_2, θ_2) and from (x_2, y_2, θ_2) to (x_1, y_1, θ_1) . We choose the one that has minimum turning angle $|\alpha_1| + |\alpha_2|$ because of the fact that smaller turning angle usually increases the chances of a collision-free path. As a result, among the two possible paths shown in Figure 4.6, we would choose (a) rather than (b).

The edges that are computed by the local planner during the construction step do not need to be stored since they can be quickly recomputed. However, their rank is saved for later use. For example, while searching for a path in the roadmap the edges with low rank are preferable.

4.2 Reachable workspace estimation

The motion planner builds a roadmap, which is later used to compute the part of the workspace where the robot's end-effector can reach. In these experiments, we assume

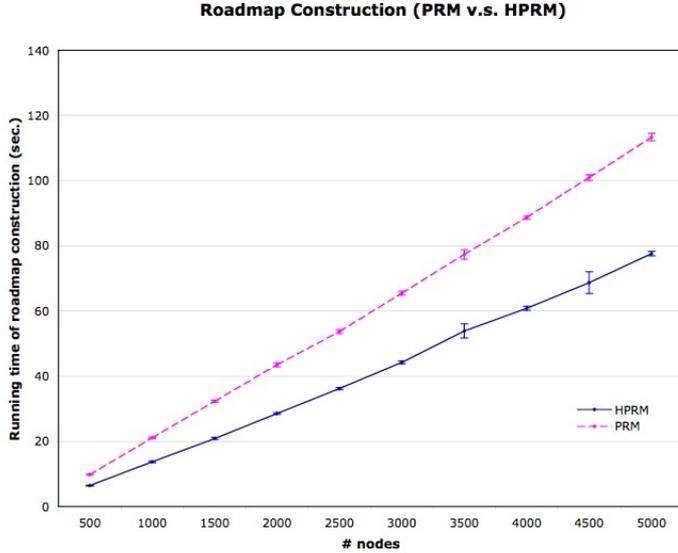


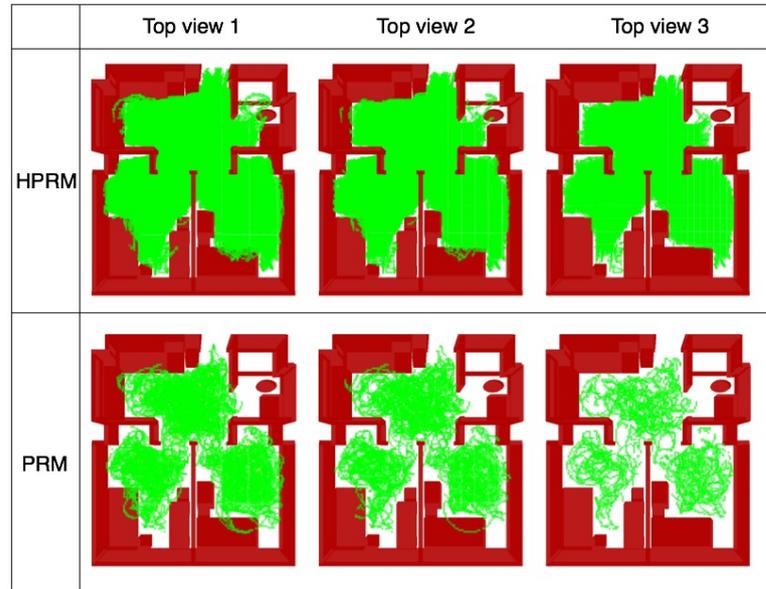
Figure 4.7: Comparison of the average running time of the construction of traditional roadmaps and the hierarchical ones. Averages are for 20 trials. Standard deviations are shown.

that obstacles remain static. We compare the basic PRM and the hierarchical PRM with regard to the workspace estimation as well as the required computation time.

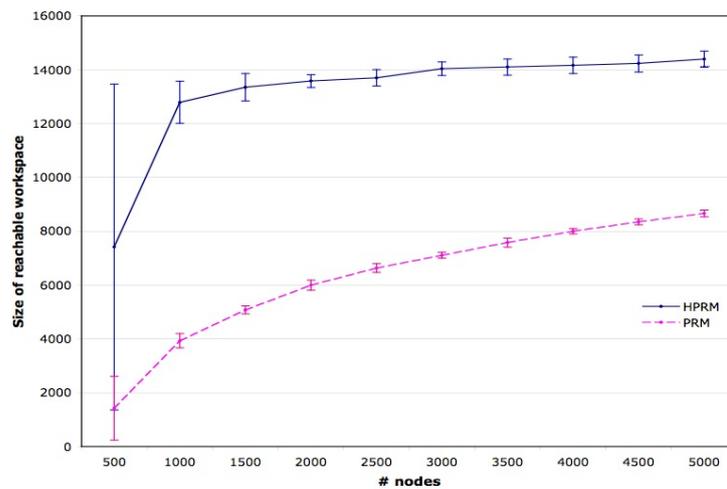
Figure 4.7 shows the comparison of the running time of the roadmap construction in the basic PRM and the hierarchical approach for the model given in Section 4.1. Because randomness is involved, running times for 20 independent runs for each case were averaged. The hierarchical PRM performed reasonably well in these experiments. As can be seen from the results, creating a hierarchical roadmap takes less time than creating a traditional roadmap. This is because the hierarchical PRM saves time in collision checking in easy regions.

In terms of the resulting reachability estimation, the hierarchical approach shows significant improvement over the traditional PRM. Figure 4.8 compares the reachable workspaces in a minimalist room mapped from the hierarchical roadmap and from the traditional roadmap, denoted by \mathcal{W}_{reach} and $\hat{\mathcal{W}}_{reach}$ respectively. \mathcal{W}_{reach} in (a) is larger than $\hat{\mathcal{W}}_{reach}$ in (b). The effect of the hierarchical approach is that more general regions in the workspace are learned.

To compare the two planners more precisely, we constructed roadmaps with identical nodes and compared the results of the average of 10 independent executions. The workspace \mathcal{W} ($420 \times 420 \times 200cm^3$) is represented using uniform cell decomposition (unit cell size is $10 \times 10 \times 10cm^3$). In Figure 4.8(b) the x-axis represents the number of nodes



(a)



(b)

Figure 4.8: Comparison of reachable workspaces computed using a hierarchical roadmap and a traditional one. The picture illustrates top views (different height) of \mathcal{W}_{reach} computed from hierarchical PRM and PRM. The reachable workspace of \mathcal{A} is represented in uniform cell decomposition. The graph shows a comparison of average size of reachable workspaces computed from hierarchical roadmap and traditional one. Standard deviations are plotted.

contained in the roadmap and the y-axis represents the size of the estimated reachable workspace measured in terms of the number of cells in \mathcal{W} . The figure shows that the size of the reachable workspace computed by the hierarchical PRM is much bigger than that computed by the traditional PRM. When the number of nodes is small, the result cannot be very reliable (the deviation value is large compared with the mean value). As the number of nodes increases, the result becomes more stable.

4.3 Environmental modifications

In this experiment we use the same modification cost function that is used in Chapter 3 (Equation 3.8). All the possible modifications except removal have the same cost 1 and the removal of each obstacle has the same cost 5 ($4 + 1$) since there are 4 items in \mathcal{M} . While setting the cost bound σ consider the following three special cases:

1. $\sigma = 2$: At most 2 obstacles are allowed to be moved and no removal is allowed.
2. $\sigma = 4$: All the 4 obstacles are allowed to be moved and no removal is allowed.
3. $\sigma = 8$: All the 4 obstacles are allowed to be moved and at most one can be removed.

The results that are computed by the environmental planner for the three cases are illustrated in Figure 4.9. In Case 1 when at most two obstacles can be moved the dining table and the desk are selected since they have greater impact on the reachability analysis. The dining table is initially placed near the corner of the kitchen. This causes an unreachable space (A). The planner suggests moving the dining table to (A) to enlarge the reachable workspace to include (F). A similar strategy applies to the desk as well, which initially had unreachable space on its two sides (B and C). It is moved towards the corner to minimize the wasted space and enlarge the reachable workspace (G). In Case 2, more obstacles are allowed to be moved than Case 1. The changes made to the dining table and the desk are the same. In addition, the drawer is moved to eliminate the wasted spaces (D) and (E). In Case 3, the desk was removed because its removal frees the largest space (I) among all the obstacles.

Note that the planner tends to remove the largest obstacle since its removal frees the largest workspace (e.g. in Case 3 the desk is removed). However, this usually happens when there is no obstacle causing dis-connectivity of the roadmap. For example, if there is a small obstacle placed at the entrance of a bedroom that blocks access then the planner will remove this obstacle because its removal can connect the kitchen and the bedroom which increases the reachable workspace the most.

Figure 4.10 shows a brute-force result of the computation of the sizes of reachable workspaces for all possible environmental layouts. The result that is computed by the

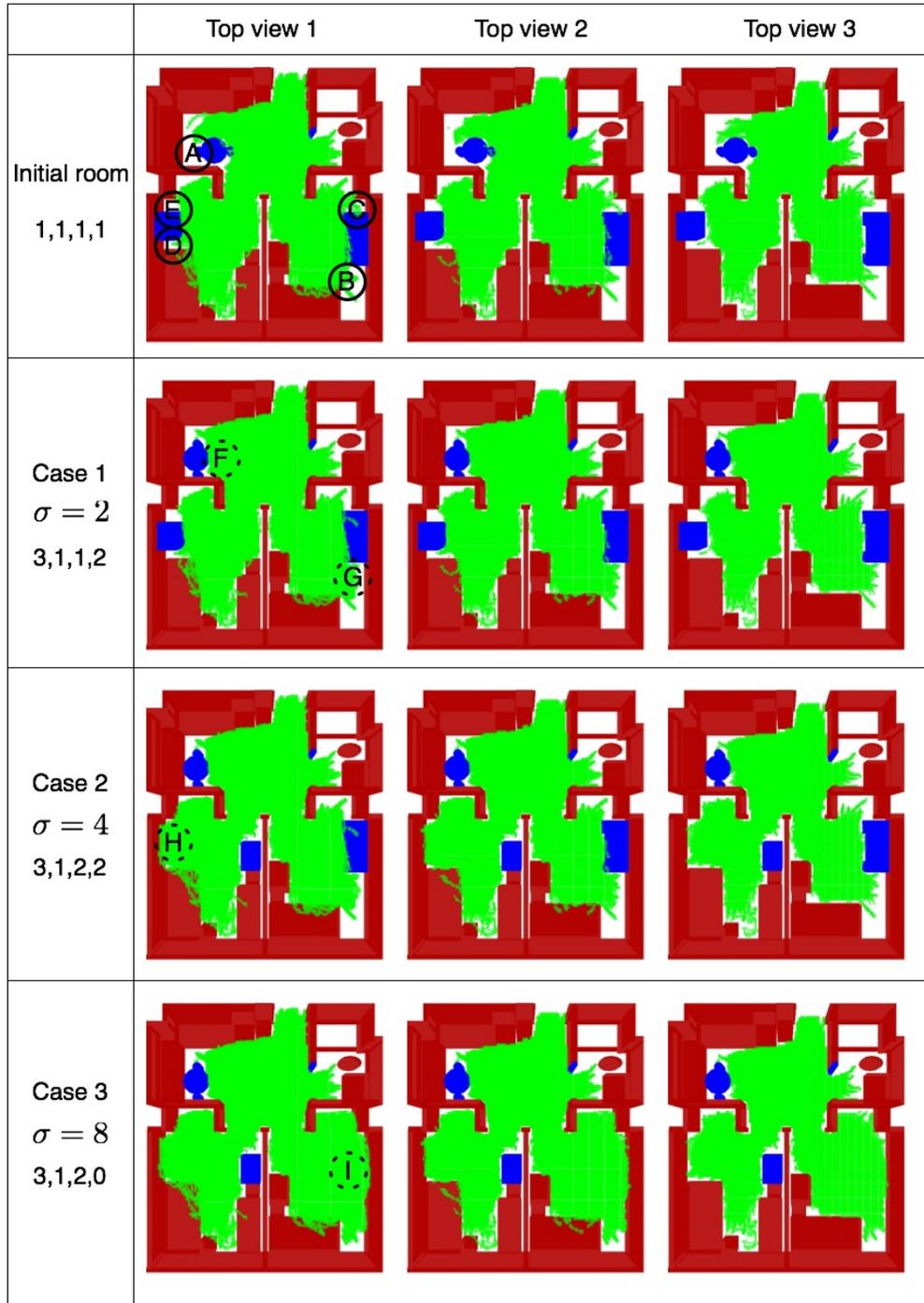


Figure 4.9: The result computed by the environmental planner in the three cases shown in top views (three layers). Certain spaces are labeled for reference. See text for details.

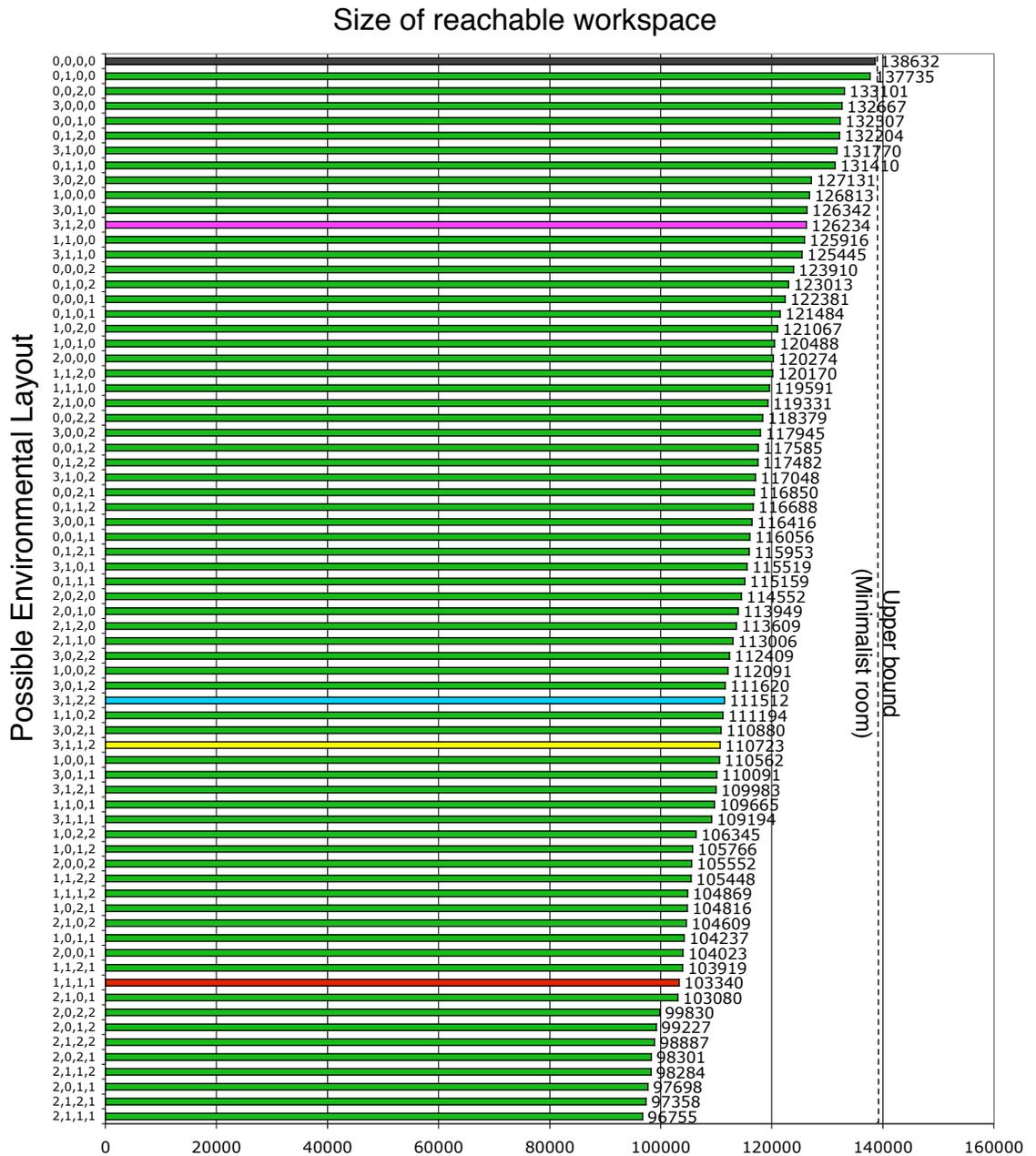


Figure 4.10: The size of reachable workspace (x-axis) computed for the 72 possible environmental layouts (y-axis). The upper bound is the minimalist room (the grey bar). The red bar corresponds to the initial \mathcal{W}_{reach} . Results of Case 1, 2 and 3 are colored in yellow, blue and pink respectively.

environmental planner depends on the specified cost bound. When the cost bound is zero there cannot be any modifications made to the environment. As the cost bound becomes less restricted (i.e. more obstacles can be moved or more obstacles can be removed) $|\mathcal{W}_{reach}|$ increases towards the optimal solution (minimalist room). The solution obtained with $\sigma = 2, 4,$ and 8 are colored in yellow, blue and pink respectively.

4.4 Summary

This chapter presented an accessibility assessment in a student residence at York University using the methodology described in Chapter 3. The Playbot Robotic Wheelchair is used as the kinematic model to demonstrate wheelchair users' reachability. We have demonstrated the effectiveness of the reachable workspace estimation of the motion planner and performed analysis and comparison of the environmental modifications suggested by the environmental planner.

The hierarchical PRM was compared to the traditional PRMs. We studied how the hierarchical approach affects the motion planner's performance, as measured in terms of running time and size of reachable workspace. Our experiments show that the hierarchical PRM spends less time than the traditional PRM in creating the roadmap. Moreover, it shows effectiveness of reachable workspace estimation. The reachable workspace mapped from hierarchical roadmap is a few times larger than that mapped from the traditional roadmap.

Finally, experimental results of the automatic environmental planner are presented. We have shown and analyzed the solutions provided by the environmental planner under different cost bounds. Although possible modifications are limited to four obstacles, the analysis revealed influences of different modifications on the accessibility in the environment. The experiment has shown that the integration of the motion planner and the environmental planner developed in this report can be a viable tool to assess the accessibility of real environments.

Chapter 5

Discussion and future work

5.1 Discussion

This report investigated accessibility assessment of an environment using advanced planning methodologies. In the practice of clinicians and architects, accessibility assessment is a knowledge-intensive activity. Building on a motion and environmental planner, a tool has been developed to assess accessibility and suggest environmental modifications. The algorithm combines randomized motion planning with a greedy optimization strategy to compute simultaneously a configuration of the environment and a corresponding reachable workspace.

The methodology depends on an efficient motion planner which can be generally applied to any kinematic structure such as wheelchair users and other users requiring mobility assists such as walkers. The motion planner is based on a PRM, which uses a hierarchical strategy to maximize the reachability of each configuration. Unlike traditional PRMs and most of its variants, which treat the DOF of the kinematic structure equally, the planner developed in this report applies a hierarchical strategy in the construction of the probabilistic roadmap in \mathcal{C}_{free} . This approach makes the PRM particularly useful for accessibility assessment. The collision detection in open regions is accelerated by using conservative occupancy analysis. The main advantage of this hierarchical PRM over the traditional PRM is to maximize the kinematic structure's reachability of every node and edge in the probabilistic roadmap. It has also shown tremendous effectiveness in establishing the reachable workspace.

Another important idea of our work is the combination of two algorithms. The hierarchical PRM works well with the automated environmental planner since it allows fast re-planning over related environments. Our environmental planner searches the superset of possible placements in a "greedy" way. A modification cost bound is introduced to avoid the trivial solution where all obstacles are missing. The greedy algorithm iteratively manipulates the placement of each modifiable obstacle to obtain a locally optimal placement. At each iteration it selects the modifiable obstacle whose removal can have the most significant impact on the reachable workspace and then searches for its best placement within cost bound. Arbitrary removal of obstacles is avoided by setting the cost bound, however, if removal is within the cost bound then removal is not distinguished from other modifications when choosing an optimal placement for an obstacle.

In the case study of the application of the accessibility assessment methodology, a kinematic model was developed of a wheelchair robot to estimate the reachability of wheelchair users. We conducted experiments on a 3D environment that represents a world environment. The hierarchical PRM was more efficient and effective in estimating the reachable

workspace when compared with the traditional PRM. The suggested environmental modifications provided by the planner resulted in a larger reachable workspace than the original one. The analysis revealed influences of different modifications on the accessibility in the environment. Although the experiment made a few assumptions on the kinematic model and the environmental model it has shown that the integration of the motion planner and the environmental planner developed in this report can be a viable tool to assess the accessibility of real environments.

5.2 Future work

The developed system can be a useful tool to aid people with disabilities, clinicians, architects, and engineers to assess accessibility of a built environment. This work has demonstrated the power of computer technologies in assisting knowledge-intensive practical problems. It also provides directions for future work.

5.2.1 Extending the motion planner

Our current hierarchical PRM examines hierarchies in its two main computations: occupancy analysis and reachability analysis. In some way the established hierarchies of nodes and edges indicate the difficulty of the workspace as well as the configuration space. The hierarchical characteristic may also be employed in other aspects of motion planners. One heuristic is to let the established hierarchy lead the sampling toward the boundaries of obstacles, i.e. to sample more densely near nodes with higher hierarchy labels than those with lower hierarchy labels. Another approach could be to use this information to prune the constructed roadmap. Perhaps only a representative node with low hierarchy in a neighborhood might be "useful" enough. These strategies would generate smaller roadmaps which assure low query time and low memory consumption.

The definition of the hierarchical roadmap could be more sophisticated. For example, our current definition defines the hierarchy of an edge to be the maximum value of hierarchies of all the configurations along it. Instead of forcing the edge hierarchy to be constant, we can imagine using a hierarchy variable to indicate the different reachability and occupancy on the edge, i.e. any $c_i \in e$ has its own r_i . In the examples shown in this report, we assumed the DOFs of the robot are pre-ordered. However, for a complex robotic platform the ordering may not be obvious. Imagine a robotic wheelchair that has two manipulators attached. A more adaptive definition of the ordering scheme is required.

A more sophisticated definition of reachable workspace might involve establishing the number of configurations from which the kinematic structure can reach a given location. This can provide insight into different levels of accessibility. A space for where there exists

many reachable configurations should probably be considered more accessible than one with just a few.

5.2.2 Extending the environmental planner

The environmental planner assumes that the modifiable obstacles and their potential placements are specified in advance. This requires input from the user. A logical extension to this assumption is the automatic generation of such potential modifications. It may also require approaches to deal with continuous potential placements instead of discrete ones.

An important part of the method is the modification cost function which is used to lead the greedy planner towards the optimal solutions with relatively small cost. The examples presented in this report only deal with the difference of removal and other possible modifications. In practice there may be more complicated scenarios. For example, modifying a large object may be more expensive than modifying a small one. Future work might develop a robust cost function to evaluate potential modifications among different obstacles.

5.2.3 Extending the accessibility assessment research

This research has attempted to develop algorithms and systems to tackle the problem of accessibility assessment from the perspective of Computer Science. The author has given demonstrations of the tool to trained assessors and wheelchair users and received positive feedback. Although this report presented an application of the methodology on a realistic environmental example, it is necessary to verify the solution in real environments. Such verifications can be done in two ways: (i) people with mobility assists could visit the environment and give feedback of the modified environments; (ii) trained assessors could evaluate the suggested modifications using existing building code and their experience. A formal evaluation system that can be used by the users to test the usability of the tool should also be developed.

Our developed system for accessibility assessment can be utilized in both homes and public spaces. This thesis presented experiments of the tool on a student residence suite. Future experiments on larger environments such as hospitals and rehabilitation centres where accessibility is essential would be helpful to further evaluate the tool and explore the accessibility of these spaces.

Bibliography

- [1] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo. OBPRM: an obstacle-based prm for 3d workspaces. In *Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics : the algorithmic perspective (WAFR '98)*, pages 155–168, Natick, MA, USA, 1998. A. K. Peters, Ltd.
- [2] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man and Cybernetics*, 22(2):224–241, 1992.
- [3] J. Barraquand and J.-C. Latombe. Robot motion planning: a distributed representation approach. *The International Journal of Robotics Research*, 10(6):628–649, 1991.
- [4] S. Berchtold and B. Glavina. A scalable optimizer for automatically generated manipulator motions. In *Proceedings of IEEE/RSJ/GI International Workshop of Intelligent Robots and Systems (IROS '94)*, pages 1796–1802, 1994.
- [5] J. P. Berg and M. H. Overmars. Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '04)*, 2004.
- [6] J. P. N. Berg, L. D. Jaillet, and M. H. Overmars. Creating robust roadmaps for motion planning in changing environments. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*, pages 1053–1059, August 2005.
- [7] F. Blais. Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1):231–240, January 2004.
- [8] Access Board. Americans with disabilities act accessibility guide. *U.S. Architectural and Transportation Barriers Compliance Board*, 1997.
- [9] V. Boor, M. H. Overmars, and A. F. van der Stappen. Gaussian sampling for probabilistic roadmap planners. In *Proceedings IEEE International Conference on Robotics and Automation (ICRA '99)*, volume 1, pages 1018–1023, May 1999.
- [10] J. F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA, 1988.
- [11] H. Choset, K. M. Lynch, S. Hutchinson, W. Burgard G. Kantor, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.
- [12] A. D. Collins. *Configuration Spaces in Robotic Manipulation and Motion Planning*. PhD thesis, Duke University, 2002.
- [13] A. D. Collins, P. K. Agarwal, and J. L. Harer. HPRM: a hierarchical PRM. In *Proceedings of The 2003 IEEE International Conference on Robotics and Automation (ICRA '03)*, volume 3, pages 4433–4438, 2003.

- [14] M. de Berg, M. van Krefeld, M. Overmars, and O. Schwarzkoph. *Computational Geometry, Algorithms and Applications*. Springer Verlag, 2000.
- [15] W. E. Dixon, D. M. Dawson, E. Zergeroglu, and A. Behal. *Nonlinear Control of Wheeled Mobile Robots*. Springer-Verlag London Ltd, 2000. Vol. 262 Lecture Notes in Control and Information Sciences.
- [16] G. Dudek and M. Jenkin. *Computational Principles of Mobile Robotics*. Cambridge University Press, 2000.
- [17] Wheelchair Foundation. The overwhelming need for wheelchairs, 2006. http://wheelchairfoundation.org/about_us/the_need.php.
- [18] C. S. Han, K. H. Law, J.-C. Latombe, and J. C. Kunz. A performance-based approach to wheelchair accessible route analysis. *Advanced Engineering Informatics*, 12:53–71, 2002.
- [19] D. Helbing, I. Farkas, and T. Vicsek. Simulating dynamical features of escape panic. *Nature*, 407:487–491, 2000.
- [20] A. Hogue and M. Jenkin. Development of an underwater vision sensor for 3d reef mapping. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, 2007.
- [21] C. Holleman and L. E. Kavraki. A framework for using the workspace medial axis in prm planners. In *Proceedings IEEE International Conference on Robotics and Automation (ICRA '00)*, volume 2, pages 1408–1413, April 2000.
- [22] T. Horsch, F. Schwarz, and H. Tolle. Motion planning for many degrees of freedom – random reflections at c-space obstacles. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '94)*, pages 3318–3323, 1994.
- [23] D. Hsu. *Randomized Single-query Motion Planning in Expansive Spaces*. PhD thesis, Stanford University, May 2000.
- [24] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In *Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics : the algorithmic perspective (WAFR '98)*, pages 141–153, Natick, MA, USA, 1998. A. K. Peters, Ltd.
- [25] Eos System Inc. Photomodeler pro, 2004. <http://photomodeler.com/pmpro01.html>.
- [26] M. Kallmann and M. Mataric. Motion planning using dynamic roadmaps. In *Proceedings IEEE International Conference on Robotics and Automation (ICRA '04)*, volume 5, pages 4399–4404, April-May 2004.
- [27] L. Kavraki and J.-C. Latombe. Randomized preprocessing of configuration space for fast path planning. In *Proceedings IEEE International Conference on Robotics and Automation (ICRA '94)*, pages 2138–2145, 1994.

- [28] L. E. Kavraki, J.-C. Latombe, R. Motwani, and P. Raghavan. Randomized query processing in robot path planning. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing (STOC '95)*, pages 353–362, New York, NY, USA, 1995. ACM Press.
- [29] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, August 1996.
- [30] W. Khalil and E. Dombre. *Modeling, Identification and Control of Robots*. Hermes Penton Ltd., 2002.
- [31] J. Kim. *Development and Effectiveness of a Virtualized Reality Telerehabilitation System for Accessibility Analysis of Built Environment*. PhD thesis, University of Pittsburgh, 2005.
- [32] J. Kim, R. Pearce, and N. M. Amato. Extracting optimal paths from roadmaps for motion planning. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '03)*, pages 2424–2429, 2003.
- [33] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k -DOPs. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):21–36, March 1998.
- [34] J. Kollmann, A. Lankenau, A. Buhlmeier, B. Krieg-Buhlmeier, and T. Rofer. Navigation of a kinematically restricted wheelchair by the parti-game algorithm. In *Proceedings of AISB workshop on Spatial Reasoning in Mobile Robots and Animals*, pages 1361–6161, 1997.
- [35] H. Kurniawati and D. Hsu. Workspace importance sampling for probabilistic roadmap planning. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, volume 2, pages 1618–1623, 2004.
- [36] M. Kwietnewski, S. Wilson, A. Topol, S. Gill, J. Gryz, M. Jenkin, P. Jasiobedzki, and H.-K. Ng. A multimedia database system for 3d crime scene representation and analysis. In *Proceedings of 3rd International Workshop on Computer Vision meets Databases (CVDB)*, 2007.
- [37] J.-C. Latombe. *Robot Motion Planning*. Cluwer, 1991.
- [38] J.-C. Latombe. Motion planning: a journey of robots, molecules, digital actors, and other artifacts. *The International Journal of Robotics Research*, 18(11):1119–1128, November 1999.
- [39] J.-P. Laumond, M. Taix, and P. Jacobs. A motion planner for car-like robots based on a global/local approach. In *Proceedings of IEEE International Workshop of Intelligent Robots and Systems (IROS '90)*, pages 765–773, 1990.
- [40] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at <http://planning.cs.uiuc.edu>.

- [41] P. Leven and S. Hutchinson. A framework for real-time path planning in changing environments. *The International Journal of Robotics Research*, 21(12):999–1030, 2002.
- [42] T. Lozano-Pérez and M.A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM (CACM)*, 22(10):560–570, 1979.
- [43] R Manseur. *Robot Modeling and Kinematics*. Da Vinci Engineering Press, 2006.
- [44] J. McNeil. Americans with disabilities: 1997 household economic studies. *U.S. Census Bureau*, pages 70–73, February 2001.
- [45] C. O’Dunlaing and C.K. Yap. A retracting method for planning the motion of a disc. *Journal of Algorithms*, 6:104–111, 1982.
- [46] The Parliament of Australia. *Disability Discrimination Act*. 1992. Available at <http://www.austlii.edu.au>.
- [47] The U.S. Department of Housing and Urban Development. <http://www.fairhousingfirst.org>.
- [48] The U.S. Department of Justice. *Americans with Disabilities Act*. 1990. Available at <http://www.usdoj.gov/crt/ada/publicat.htm>.
- [49] The Government of Ontario. *Ontarians with Disabilities Act*. 2001. Available at http://www.odacommittee.net/major_docs.html.
- [50] The Parliament of the Republic of South Africa. *The Promotion of Equality and the Prevention of Unfair Discrimination Act*. 2000. Available at www.iwraw-ap.org/resources/pdf/South20Africa_GE1.pdf.
- [51] The Parliament of U.K. *Disability Discrimination Act*. 1995.
- [52] O. Palmon, R. Oxman, M. Shahar, and PH. Weiss. Virtual environments as an aid to the design and evaluation of home and work settings for people with physical disabilities. In *Proceedings of 5th International Conference on Disability, Virtual Reality and Associated Technologies*, pages 119–124, 2004.
- [53] X. Pan. Wheelchair-accessible design assistant, 2006. http://eil.stanford.edu/xpan/motion_planning/index.html.
- [54] X. Pan, C.S. Han, and K. H. Law. Using motion-planning to determine the existence of an accessible route in a CAD environment. Technical Report ., Department of Civil and Environmental Engineering, 2005.
- [55] J. H. Reif. Complexity of the mover’s problem and generalizations. In *Proceedings of IEEE Symposium on Foundations of Computer Science (FOCS ’79)*, pages 421–427, 1979.

- [56] J. M. Saez, A. Hogue, F. Escolano, and M. Jenkin. Underwater 3d slam with entropy minimization. In *Proceedings of IEEE International Conference on Robotics and Automation, (ICRA '06)*, pages 15–19, May 2006.
- [57] M. Saha. *Motion Planning with Probabilistic Roadmaps*. PhD thesis, Stanford University, 2006.
- [58] A. Scheuer and Th. Fraichard. Continuous-curvature path planning for car-like vehicles. In *Proceedings of IEEE/RSJ International Workshop of Intelligent Robots and Systems (IROS '97)*, volume 2, pages 997–1003, 1997.
- [59] J. T. Schwartz and M. Sharir. On the ‘piano movers’ problem: II. general techniques for computing topological properties of real algebraic manifolds. *Advances in Applied Mathematics*, 4:298–351, 1983.
- [60] S. Se and P. Jasiobedzki. Photorealistic 3d model reconstruction. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '06)*, pages 3076–3082, 2006.
- [61] M. Shields. Use of wheelchairs and other mobility support devices. *Health Reports*, 15(3):37–41, May 2004.
- [62] G. Song and N.M. Thomas, S. and Amato. A general framework for prm motion planning. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '03)*, volume 3, pages 4445–4450, September 2003.
- [63] F. Stahl. Computer simulation modeling for informed design decision making. In *Proceedings of 13th Annual Conference of the Environmental Design Research Association*, pages 105–111, 1982.
- [64] E. Steinmetz. Americans with disabilities: 2002 household economic studies. *U.S. Census Bureau*, pages 70–107, May 2006.
- [65] P. Svestka. A probabilistic approach to motion planning for car-like robots. Technical Report RUU-CS-93-18, Department of Computer Science, Utrecht University, Utrecht, The Netherlands, April 1993.
- [66] P. Svestka and M.H. Overmars. Motion planning for carlike robots using a probabilistic learning approach. *The International Journal of Robotics Research*, 16(2):119–143, 1997.
- [67] J.K. Tsotsos, G. Verghese, S. Dickinson, M. Jenkin, A. Jepson, E. Milios, F. Nuflo, S. Stevenson, M., M. Black, D. Metaxas, S. Culhane, Y. Ye, and R. Mann. PLAYBOT: A visually-guided robot to assist physically disabled children in play. *Image and Vision Computing, Special Issue on Vision for the Disabled*, 16:275–292, 1998.
- [68] Author Unknown. Accessibility. *The Polio Chronicle*, February 1934. www.disabilitymuseum.org.

- [69] J. P. van den Berg and M. H. Overmars. Roadmap-based motion planning in dynamic environments. *IEEE Transactions on Robotics*, 21:885–897, October 2005.
- [70] G. van den Bergen. Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools: JGT*, 2(4):1–14, 1997.
- [71] S. Wilmarth, N. Amato, and P. Stiller. Maprm: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '99)*, pages 1024–1031, 1999.
- [72] W. Yan and Y. Kalay. Simulating human behaviour in built environments. In *Proceedings of the 11th International Computer Aided Architectural Design Futures Conference (CAAD '05)*, pages 301–310, June 2005.