NAVIGATION OF A MOBILE ROBOT IN AN UNKNOWN ENVIROMENT

Lee Gim Hee

Department of Mechanical Engineering
National University of Singapore

Session 2004/2005

NAVIGATION OF A MOBILE ROBOT
IN AN UNKNOWN ENVIROMENT

Submitted By
Lee Gim Hee

Department of Mechanical Engineering

In Partial Fulfilment of the Requirements for the
Degree of Bachelor of Engineering
National University of Singapore

Session 2004/2005

Contents

Acknowledgment v

Abstract vi

List of Figure vii

List of Symbols ix

1. Introduction

1.1 Background 1

1.2 Objectives 3

1.3 Scope of Project 3

1.4 Assumptions 4

1.5 Outline of Thesis 4

2. Literature Review

2.1 Basic Concepts 6

2.1.1 Definition of Mobile Robot 6

2.1.2 Holonomic versus Non-Holonomic 6

2.2 Fundamental Methods 7

2.2.1 Roadmap 7

2.2.2 Cell Decomposition 8

2.2.3 Navigation Function 9

2.2.4 Vector Field Histogram 10

2.2.5 Potential Field 11

2.3 Limitations of the Current Methods 12

3. Map Building

3.1 Introduction 15

3.2 Description of Sensors 16

3.3 Grid-Based Representation 17

3.4 Sensor Interpretation 19

3.5 Updating of Map 20

3.6 Implementation and Results 22

4. Path Planning and Execution

4.1 Introduction 24

4.2 Improved Navigation Function 25

4.3 Potential Field Method 29

4.3.1 Potential Functions 29

4.3.2 Force Functions 30

4.4 Frontier-Based Exploration 32

4.5 Integrated Method 34

4.5.1 Goal Reachability 36

4.5.2 Computation of Sub-Goal 36

4.5.3 Reaching for the Sub-Goal 38

4.5.4 Reaching for the Goal 40

4.6 Implementation and Results 41

5. Conclusion

5.1 Epilogue 42

5.2 Further Work 43

References 46

Appendices

1. Definition of 1-Neighbour 48

2. Description of the *min* function 48

Acknowledgements

The author wishes to express his heart felt gratitude to his supervisor, Associate Professor Marcelo H. Ang Jr and his co-supervisor, Mr Lim Chee Wang from the Singapore Institute of Manufacturing Technology. The willingness to share their knowledge and vision has provided guidance for the author to complete this thesis.

He would also like to show his appreciation to those dedicated individuals from the Singapore Institute of Manufacturing Technology and the Control and Mechatronic Laboratory of the National University of Singapore who has assisted him in the course of his project.

To his parents the author is grateful for their unconditional support and his brother who has given him lots of motivation.

Finally, the author would like to thank Grace Tang who is always by his side. She has always been a source of inspiration, strength, courage and happiness to the author.

Abstract

The autonomy of mobile robots is continuously gaining importance particularly in the military, manufacturing and even entertainment industries.

Three competencies are identified for a mobile robot to be autonomous. First, the mobile robot must be able to stay away from hazards such as obstacles or operating conditions dangerous to itself and it must not pose any risk to humans in its vicinity. Second, the robot must possess the capability of planning its path so that it will always be able to travel from a starting point to a given goal. Third, the first two competencies should be accomplished with minimal human intervention even when the mobile robot is operating in an unknown environment.

Many existing algorithms such as the potential field, vector field histogram, roadmap, cell decomposition and navigation function are not capable of giving the mobile robot all the competencies to achieve autonomy. The greatest challenge is thus to build an algorithm that gives the mobile robot all the three competencies in one single framework.

In this thesis, the *integrated algorithm*, which is capable of providing the three competencies in one single framework, is proposed and implemented on the Nomad XR4000 mobile robot.

List of Figures

Chapter 1

1.1 Normad XR4000 mobile robot

Chapter 2

2.1 Visibility graph with polygonal obstacles

2.2 Path planning using Navigation Function

2.3 Robot's motion influenced by Potential Field

2.4 Example of local minima from potential field method

Chapter 3

3.1 Schematic diagram for frame definition

3.2 Position of obstacle with respect to F_w

3.3 Increment and decrement of certainty values for corresponding obstacles

3.4 Map generated by the grid-based representation

Chapter 4

4.1 Path generated by improved navigation function, $\alpha = 1.0$

4.2 Path generated by improved navigation function, $\alpha = 2.0$

4.3 Orientation of ultrasonic sensor *i* is given by the sum of α_i and β

4.4 A frontier region made up of a group of adjacent frontier cells

4.5 The integrated algorithm

4.6 (a) The goal is reachable because it is in the gC_{free} region

(b) The goal is unreachable because it is in the $gC_{unknown}$ region

4.7 Sub-goal is the centroid of the frontier that intersects the NF path

4.8 Illustration of the hybrid method

4.9 Actual route taken by the robot during run-time

4.10 The robot moving towards the goal during run-time

Chapter 5

5.1 Map generated without localization

Appendices

A1.1 The shaded cells are the 1-Neighbour of the cell (x, y)

A2.1 Priority of neighbouring cells of (x,y)

List of Symbols

$\mathbf{F_R}$	Resultant force generated by the potential field method
$\mathbf{F_o}$	Repulsive force generated by obstacles in the potential field method
$\mathbf{F_g}$	Attractive force generated by the goal in the potential field method
F_A	Moving Frame attached to robot
F_w	Fixed Frame attached to workspace W
gC	Discretized retangloid grid cells of the workspace W
gC_{free}	Subset of gC in the free space
$gC_{unknown}$	Subset of gC in the unknown region
$gC_{occupied}$	Subset of gC in the region occupied by obstacles
gC_{border}	Subset of gC_{free} in the border of obstacles
$L_{\mathbf{B}}$	The link list to store all the coordinates of the gC_{border} grid cells
$L_{\mathbf{0}}$	The link list to store the coordinates of the grid cells waiting to be processed in the computation of the improved navigation function
$L_{\mathbf{p}}$	The link list to store the coordinates of the grid cells that constitutes the path generated from the improved navigation function
N	Navigation Function
${}^{\mathbf{W}}\mathbf{P_A}$	The position of F_A w.r.t F_w
${}^{\mathbf{A}}_{\mathbf{i}}\mathbf{P_o}$	The position of an obstacle detected by the i^{th} sensor reading w.r.t F_A

${}^{w}_{i}\mathbf{P}_{o}$	The position of an obstacle detected by the i^{th} sensor reading w.r.t F_w
${}^{W}\mathbf{R}_{A}$	The orientation of F_A w.r.t F_w
$\mathbf{U}_{g}$	The attractive potential generated by the goal in the potential field method
${}^{i}\mathbf{U}_{o}$	The repulsive potential generated by the obstacle in the potential field method
$\mathbf{U}$	The total potential generated by the potential field method
W	Workspace of Mobile Robot
$\boldsymbol{\alpha}$	The safe distance from the obstacles in the improved navigation function

Chapter 1

Introduction

1.1 Background

The autonomy of mobile robots is continuously gaining importance particularly in the military for surveillance as well as in the industry for inspection and material handling tasks. A further emerging market with enormous potentials is the mobile entertainment robots.

Three competencies are identified for a mobile robot to be autonomous. First, the mobile robot must be able to stay away from hazards such as obstacles or operating conditions dangerous to itself and it must not pose any risk to humans in its vicinity. Second, the robot must possess the capability of planning its path so that it will always be able to travel from a starting point to a given goal. Third, the first two competencies should be accomplished with minimal human intervention even when the mobile robot is operating in an unknown environment.

Many algorithms such as the *potential field, vector field histogram, roadmap, cell decomposition* and *navigation function*, which will be discussed in greater details in section 2.2, were developed over the years for the autonomy of mobile robots. These algorithms are however not capable of giving the mobile robot all the three competencies in one single framework. For example the *potential field* and the *vector field histogram* algorithms, which are collectively known as the *local methods*, give the mobile robots the capability of online collision avoidance but without the capability of planning its own path. The other algorithms including the *roadmap*, *cell decomposition* and *navigation function*, which are collectively known as the *global methods*, give the mobile robots path planning capability only if the information of the environment is surveyed and provided by humans. The *global methods* are also not capable of doing online collision avoidance. See section 2.3 for greater details on the limitations of the existing algorithms.

The greatest challenge is thus to build an algorithm that gives the mobile robot all the three competencies in one single framework. In this thesis, the *integrated algorithm*, which is capable of providing the three competencies in one single framework, is proposed and implement on the Nomad XR4000 mobile robot. See section 4.5 for greater details on the *integrated algorithm*.

1.2 Objectives

Figure 1.1: Normad XR4000 mobile robot

(taken from http://www.robots.com)

The objective of this Bachelor of Engineering Dissertation is to propose and implement a motion planning algorithm to the Nomad XR4000 mobile robot (shown in Figure 1.1). The proposed algorithm should give the mobile robot the capability of navigating from any starting point to a given goal in an unknown environment. The algorithm should also give the mobile robot the capability to avoid any possible static or dynamic obstacles that are in its path.

1.3 Scope of Project

To achieve the objectives, the algorithm to be implemented on the mobile robot should be able to:

i. Build a grid based map of its surrounding using sensory data from the ultrasonic and laser sensors.

ii. Generate a path connecting the start and the goal points using the grid based map.

iii. Navigate according to the planned path avoiding all static or dynamic obstacles that are in its path.

1.4 Assumptions

In this thesis, several assumptions are made:

i. The Normad XR4000 is able to report its relative and global position at any instance of time by counting the revolutions of the wheel. This method is commonly known as *odometry* method. In this dissertation, it is assumed that the odometry reading is accurate and hence no localization techniques are required.

ii. It is also assumed that the starting point and the goal belong to free space and a path connecting both positions always exists.

1.5 Outline of Thesis

This thesis is organized as follows:

Chapter 2 This chapter gives some definitions of the terms commonly used in mobile robotics. A review of some existing mobile robot navigation algorithms will also be given.

The algorithms that will be reviewed include roadmap, cell decomposition, navigation function, vector field histogram and potential field methods. The limitations to these algorithms will also be discussed.

Chapter 3 This chapter discusses the map building process. Descriptions of the ultrasonic sensors and the laser range finder will be given. The grid-based representation and the updating method for the map will be discussed.

Chapter 4 This chapter describes the improved navigation function and the potential field method in greater detail. The frontier based exploration will be introduced in this chapter. The integrated algorithm which is proposed by the author will be discussed. This algorithm modifies the frontier based exploration method, which was originally a map building technique, into a path planning algorithm. In the integrated algorithm, the improved navigation function and the potential field method are fused together with the modified frontier based exploration method into one single framework.

Chapter 5 This chapter gives the conclusion of the dissertation. Some further developments for this dissertation are suggested in this chapter.

Chapter 2

Literature Review

2.1 Basic Concepts

This section introduces some of the basic concepts common to mobile robot literature. These definitions are adapted from the standard robot motion planning text by Jean-Claude Latombe [1].

2.1.1 Definition of Mobile Robot

A mobile robot can be represented as a rigid body moving in an Enclidean workspace, $W \in \Re^N$ where N equals to 2.

2.1.2 Holonomic versus Non-Holonomic

A rigid body constrained to a plane has up to three degrees of freedom. In Cartesian space, these are often thought as X, Y and Rotation. The same rule applies to a mobile robot base moving on the floor. A non-holonomic robot is one that uses synchro-drive base, i.e. there are two axes of motion: steering and translation. When the robot wants to accelerate in a given direction, the wheels must first be oriented along that direction using the steering axis. A holonomic robot, on the other hand, can accelerate in any direction at any time. Normad XR4000 is a holonomic robot hence this thesis is restricted to holonomic mobile robots.

2.2 Fundamental Methods

A variety of methods have been used in the study of the navigation problem of mobile robots. Some methods such as *roadmap, cell decomposition, navigation function, vector field histogram* and *potential field* have been extensively used. This section gives an overview of the different methods available.

2.2.1 Roadmap

The *roadmap* approach to path planning consists of capturing the connectivity of the robot's free space in a network of one-dimensional curves, called the roadmap, lying in the free space or its closure. Once a roadmap has been constructed, it is used as a set of standardized paths. Path planning is thus reduced to connecting the initial and goal positions to points in the roadmap. Various methods based on this general idea have been proposed. They include the *visibility graph* [2], *Voronoi diagram* [3], *freeway net* [1] and *silhouette* [1].

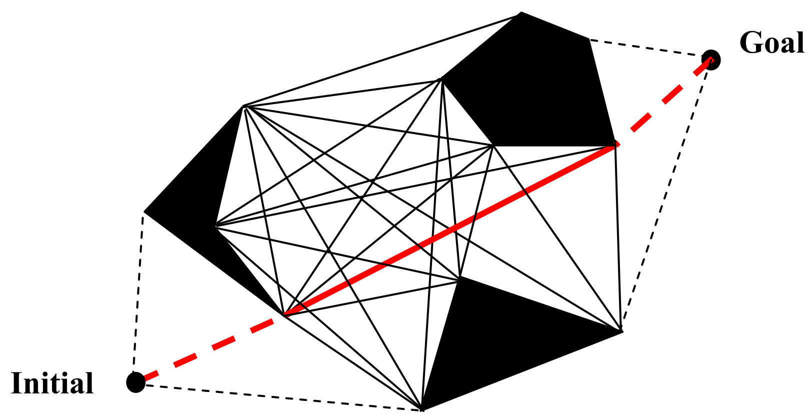


Figure 2.1: Visibility graph with polygonal obstacles

Figure 2.1 shows an example of a *visibility graph*. The plain links connects the vertices of the polygonal obstacle which forms the roadmap. The dashed link connects initial and goal to the roadmap. The bold line shows the shortest path between the initial and goal positions.

2.2.2 Cell Decomposition

Cell decomposition [4] method consists of decomposing the robot's free space into simple regions, called cells, such that a path between any two positions in a cell can be easily generated. A non-directed graph representing the adjacency relation between the cells is then constructed and searched. This graph is called the *connectivity graph*. Its nodes are the cells extracted from the free space and two nodes are connected by a link if and only the two corresponding cells are adjacent. The outcome of the search is a sequence of cells called a *channel*. A continuous path can be computed from this sequence.

2.2.3 Navigation Function

The *navigation function* method discretizes the workspace W of the robot into retangloid grid cells gC. In addition a fixed frame F_W is embedded in the workspace W of the robot. The position of individual gC can thus be specified as x and y coordinates with respect to F_W. Each gC is either free or occupied space. The subset of gC in free space is denoted by gC_{free}.

The *navigation function* N is computed using the "*wavefront expansion*" algorithm [1]. First, the value of N is set to 0 at goal. Next, it is set to 1 at every 1-neighbour[1] of goal; to 2 at every 1-neighbour of these new gC (if it has not been computed yet). The algorithm terminates when the entire subset of gC_{free} accessible from goal has been fully explored. A minimum length path can thus be generated following the steepest descent of N.

Figure 2.2 shows a simulated navigation function. The arrow tracks the steepest descent joining the starting to the goal positions. There is only one global minimum at the

[1] See Appendix 1 for definition of 1-neighbour.

goal. Hence, a path joining initial to goal positions always exist. Notice that the path however, tends to get too close to the obstacle.

The improved navigation function that prevents possible grazing of obstacle will be discussed in detail in section 4.2.

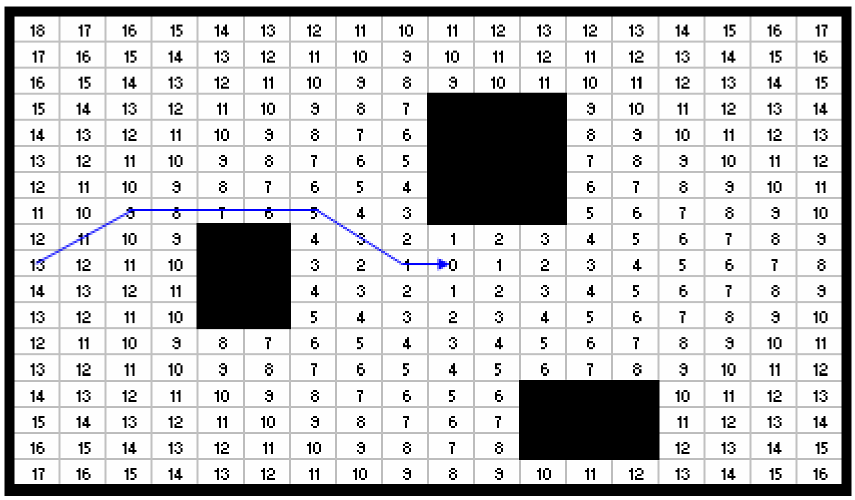


Figure 2.2: Path planning using Navigation Function

2.2.4 Vector Field Histogram

Similar to section 2.2.3, the vector field histogram method [5] discretizes the workspace W of the robot into retangloid grid cells gC. Each gC is labeled “0” if it lies in free space and “1” to “15” depending on the certainty of occupancy. The gC values are updated continuously with range data sampled by on-board sensors. A two stage data reduction process is carried out to compute the desired control command for the vehicle.

In the first stage, the gC values are reduced to a one-dimensional *polar histogram* that is constructed around the robot's momentary location. Each section in the polar histogram contains a value representing the total sum of the gC values, otherwise known as the *polar obstacle density*, in that direction. In the second stage, the algorithm selects the most suitable sector from all polar histogram sectors with a low polar obstacle density, and the next direction of movement of the robot is aligned with that direction.

2.2.5 Potential Field

The potential field method [6] is perhaps the most widely used algorithm for navigation of mobile robots. The robot is represented as a particle in the workspace moving under the influence of an artificial potential produced by the goal configuration and the obstacles. Typically the goal generates an "attractive potential" which pulls the robot towards the goal, and the obstacles produce "repulsive potential" which pushes the robot away from them. The negated gradient of the total potential is treated as an artificial force applied to the robot. At every position, the direction of this force is considered as the most promising direction of motion.

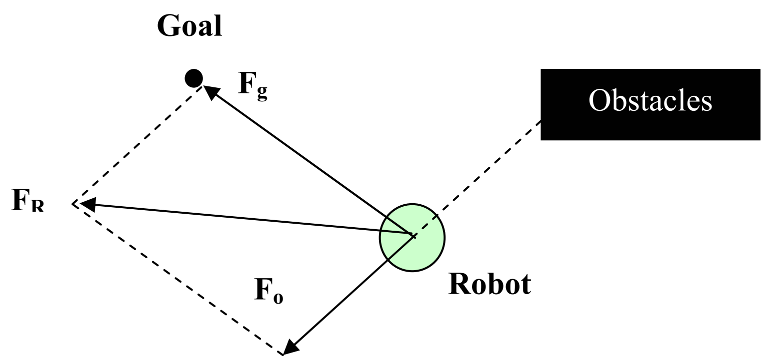


Figure 2.3: Robot's motion influenced by Potential Field

Figure 2.3 shows the repulsive force F_o that is generated from obstacles and the attractive force F_g that is generated from the goal. F_R is the resultant of both the repulsive and attractive force. It is the most promising direction whereby the robot would avoid the obstacle. A detailed discussion on Potential Field shall be presented in section 4.3.

2.3 Limitations of the Current Methods

Generally, all mobile robot navigation algorithms can be classified into two classes – *Global* and *Local Methods*.

Roadmap, cell decomposition and *navigation functions* are commonly known as *global methods*. These methods require the surrounding of the robot to be known and static. A continuous free path can thus be found in advance by analyzing the connectivity of the free space. A continuous free path always exist with the *Global method*, however any changes in the environment could invalidate this. Hence, *Global method* is usually not suitable for navigation in an initially unknown environment with dynamic obstacles.

Vector field histogram and potential field belong to another class of navigation methods commonly known as *local methods*. These methods do not include an initial processing step aimed at capturing the connectivity of the free space in a concise representation. Instead, it integrates online sensory data into motion generating process, which is otherwise known as reactive control method. Hence a prior knowledge of the environment is not needed. At any instant in time, the path is determined based on the contents of the immediate surrounding of the robot. This allows the robot to be able to avoid any dynamic obstacles in the robot's vicinity.

Local method is basically steepest descent optimization methods. This renders it to be susceptible to *local minima* [7]. Figure 2.4 shows an example of local minima in potential field method. It occurs when the attractive and the repulsive force cancels out each other. The robot will get be immobilized when it falls into a *local minima* and hence losing the capability of traveling to the goal.

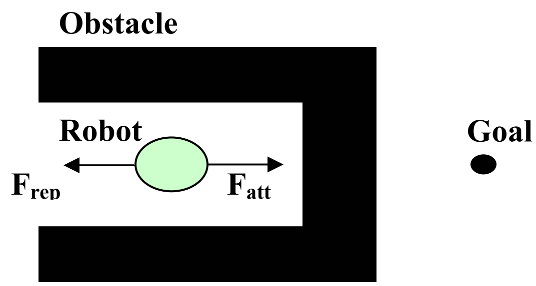


Figure 2.4: Example of local minima from potential field method

Chapter 3

Map Building

3.1 Introduction

It is impossible for a mobile robot to plan a path that allows it to navigate safely from a given starting position to a goal in an unknown environment. It is therefore imperative for the robot to have autonomy in acquiring the knowledge of its surrounding so that it would be possible for it to do path planning.

The mobile robot can gain knowledge of the world via map building. Map building is a process where the sensory information of the surrounding is made comprehensive to the robot.

This chapter starts with section 3.2 which will describe the Polaroid 6500 ultrasonic sensors and the Sick LMS 200 laser rangefinder used in the map building process. Section 3.3 will introduce the grid-based representation of the map. Section 3.4

and 3.5 will discuss about the interpretation of the sensory information and how the map is updated. The chapter ends with section 3.6 which will show the implementation results of the map building.

3.2 Description of Sensors

Two types of sensors built in the Nomad XR4000 mobile robot are used in the map building process. They are the Polaroid 6500 ultrasonic sensors and the Sick LMS 200 laser rangefinder.

Polaroid 6500 ultrasonic sensors

The Nomad XR4000 mobile robot has a total of 48 ultrasonic sensors that are divided into two sets of 24 arranged on the top and bottom perimeters of the robot respectively. The ultrasonic sensors provide range information to objects that are between 15cm to 700cm with an accuracy of one percent of the entire range. Distance information is obtained by multiplying the speed of sound by the “time of flight” of short ultrasonic pulse traveling to and from a nearby object.

Sick LMS 200 laser rangefinder

The Nomad XR4000 mobile robot has one Sick LMS 200 laser rangefinder. The laser rangefinder uses the “time of flight” range finding system based on the Sick Electro-optic LMS sensor. It provides a total of 360 readings in a planar scan of 180 degrees.

3.3 Grid-Based Representation

The sensory information of the world has to be represented in a format that is comprehensive to the robot. One possible way to do this is to use the *grid-based representation* [8].

The *grid based representation* is a tessellation of the workspace W of the robot into retangloid grid cells, gC. A fixed frame, F_w is attached to the workspace of the robot and a moving frame, F_A is attached to the robot. The position and orientation of F_w is chosen as the initial configuration of the robot. This means that the initial position and orientation of F_A is coincident with F_w.

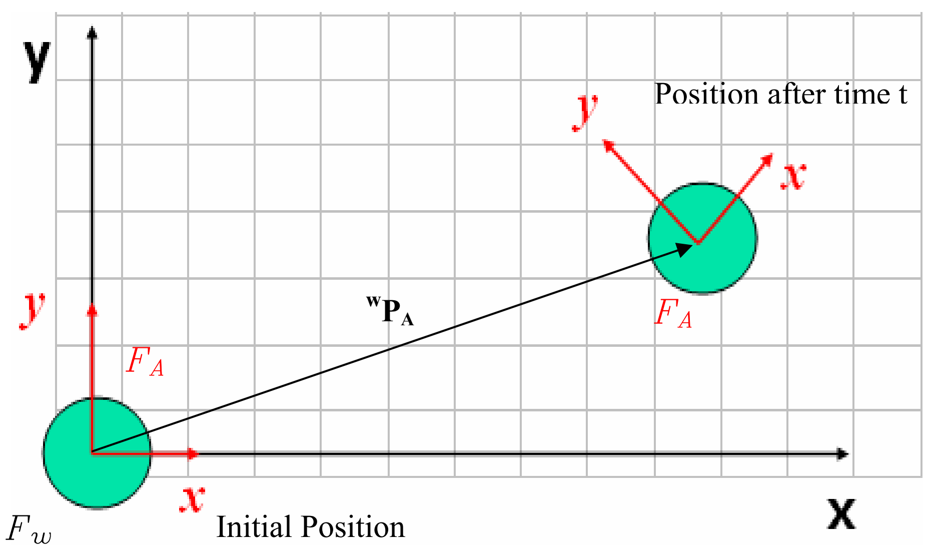


Figure 3.1: Schematic diagram for frame definition

The positions of the grid cells are specified as x and y coordinates with respect to F_W. A rotational matrix ${}^{\mathbf{W}}\mathbf{R}_{\mathbf{A}} \in \Re^{3x3}$ is defined to describe the orientation of F_A with respect to F_w and the position of the origin of F_A with respect to F_w is defined by a positional vector ${}^{\mathbf{W}}\mathbf{P}_{\mathbf{A}} \in \Re^{3x1}$. Figure 3.1 shows a schematic diagram of the frame definition.

The grid cells are classified into three categories namely $gC_{unknown}$ which represents the unexplored cells, gC_{free} which represents the unoccupied cells and $gC_{occupied}$ which represents cells that are occupied by obstacles. The classification into either one of the three classes depends on the certainty values that the cells hold. The following are the threshold ranges determined experimentally to define the certainty values that correspond to $gC_{unknown}$, gC_{free} and $gC_{occupied}$.

- $gC_{unknown}$ – certainty value range [-10, 10]
- gC_{free} – certainty value range [-40, -10)
- $gC_{occupied}$ – certainty value range (10, 40]

3.4 Sensor Interpretation

The laser range finder provides more accurate readings as compared to the ultrasonic sensors. However, the laser range finder is not used alone because laser operates in a two-dimensional plane and any obstacles that are above or below the laser range finder are obscured from its view. In contrast, the ultrasonic sensor projects a three-dimensional cone and hence any object that is obscured from the laser plane would be detected by the ultrasonic sensors. As a result, the ultrasonic reading will be updated if the reading is shorter than the reading returned by the laser range finder in the same direction.

An example would be a table with rectangular flat top that is supported by four legs at each corner. The table top would be obscured from the view of the laser range finder but yet detected by the ultrasonic sensor. In such cases, the robot will only update the grid cell corresponding to the ultrasonic sensor.

It was mentioned in section 3.2 that the 48 ultrasonic sensors are divided into two sets of 24 arranged on the top and bottom perimeters of the robot respectively. This means that a pair of ultrasonic sensors, one at the top and the other at the bottom, is placed at every 15 degrees interval around the robot. The readings that are returned by any pair of ultrasonic sensors scanning in the direction will be compared. The shorter reading among the two will be used for map building. This is because a shorter reading means that the obstacle is nearer to the robot and thus it would be more compelling for the robot to avoid it.

3.5 Updating of Map

The mobile robot does not have any knowledge of the world when it was first placed in an unknown environment. It is therefore intuitive to initialize all the certainty values in the map to 0, which corresponds to $gC_{unknown}$.

The robot starts to update the map by making a 360 degree scan using the laser range finder and the ultrasonic sensors. The certainty value in each gC will then be updated according the sensory data. The position of an obstacle detected by i^{th} sensor reading with respect to F_A is defined as the position vector ${}^{\mathbf{A}}_{\mathbf{i}}\mathbf{P}_{\mathbf{o}} \in \Re^{3x1}$. Hence, the position of the obstacle, detected by i^{th} sensor reading, with respect to F_w, ${}^{\mathbf{w}}_{\mathbf{i}}\mathbf{P}_{\mathbf{o}}$ can be obtained by equation (3.1) with ${}^{\mathbf{w}}\mathbf{P}_{\mathbf{A}}$ and ${}^{\mathbf{w}}\mathbf{R}_{\mathbf{A}}$ known from the odometry readings.

$$ {}^{\mathbf{w}}_{\mathbf{i}}\mathbf{P}_{\mathbf{o}} = {}^{\mathbf{w}}\mathbf{P}_{\mathbf{A}} + {}^{\mathbf{w}}\mathbf{R}_{\mathbf{A}}\,{}^{\mathbf{A}}_{\mathbf{i}}\mathbf{P}_{\mathbf{o}} \tag{3.1} $$

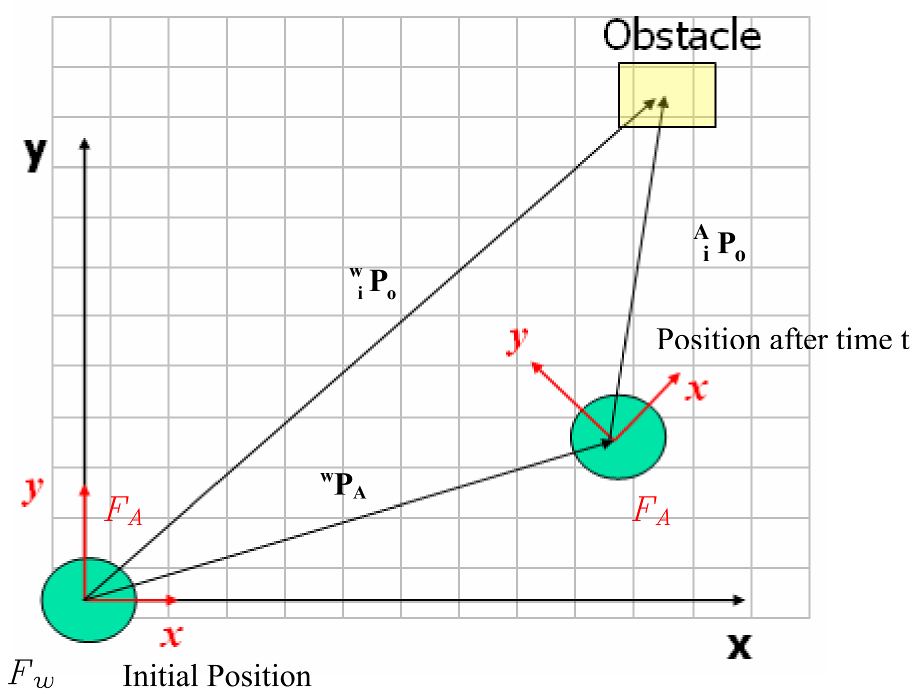


Figure 3.2: Position of obstacle with respect to F_W

Figure 3.2 shows the schematic diagram of the relation between ${}^{w}_{i}\mathbf{P}_o$, ${}^{w}\mathbf{P}_A$ and ${}^{A}_{i}\mathbf{P}_o$. The certainty values of the gC that corresponds to ${}^{w}_{i}\mathbf{P}_o$ would be increased by a value of +3. The other grid cells that intercept the line of sight of the sensors are taken to be free space and their certainty values would be decreased by 1. The increment and decrement values are determined experimentally. Figure 3.3 shows a schematic diagram of how the increment and decrement for certainty values is done.

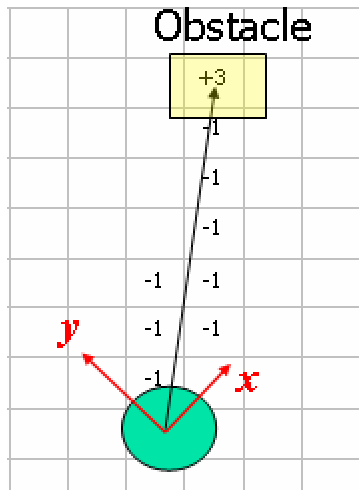


Figure 3.3: Increment and decrement of certainty values for corresponding obstacle

3.6 Implementation and Results

The *grid-based representation* map building process was carried out in an 8m by 6m laboratory. The robot was put in an arbitrary position inside the laboratory. Figure 3.4 shows the result of the map generated. The white cells represent gC_{free}, grey cells represent $gC_{unknown}$ and black cells represent $gC_{occupied}$. Each of the grid cells represents an area of 10cm x 10cm in the real world.

Notice that the outline of the obstacles are not clearly defined. This is due to noises that affect the accuracy of the sensors during the map building process. Nevertheless, the generated map has sufficient accuracy for the robot to rely upon during motion planning and execution which will be discussed in the next chapter.

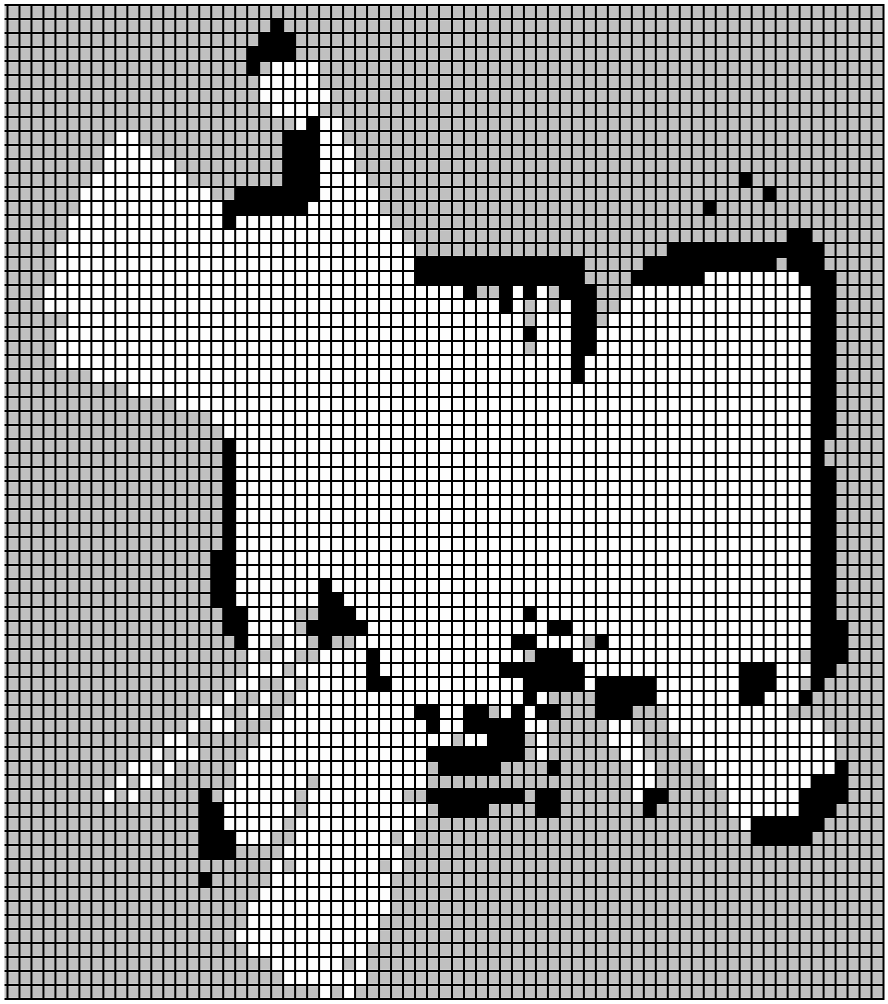


Figure 3.4: Map generated by the grid-based representation

Chapter 4

Path Planning and Execution

4.1 Introduction

The ultimate goal of mobile robotics research is to develop an algorithm that is capable of giving the mobile robots the autonomy to carrying out tasks safely in an unstructured environment. For example, a mobile robot that assists humans in picking up goods in the warehouse should be able to ‘see’ the surrounding, plan a path according to what it perceived and navigate safely to the destination avoiding any obstacles that are in its way.

In the previous chapter, the map building process, which allows the mobile robot to gain perceptions of its surrounding, was discussed and implemented. In this chapter, the generated map will used for path planning and execution.

This chapter starts by introducing the *improved navigation function* which generates a path that prevents grazing of the obstacles in section 4.2. Section 4.3 describes the *potential field* method in detail and section 4.4 will describe the *frontier-based exploration* method. The main thrust of this dissertation will be in section 4.5 where the *integrated algorithm* introduced by the author will be examined. The *integrated algorithm* modifies the *frontier-based exploration* method to incorporate both the *improved navigation function* and the *potential field* method into a single framework where path planning and obstacle avoidance is carried out.

4.2 Improved Navigation Function

A major drawback of the *navigation function* introduced in section 2.2.3 is that it generates a path which grazes the obstacles. In contrast, the *improved navigation function* computes a path that will always maintain a minimum safe distance α away from the obstacles.

The path that is generated by the *improved navigation function* is computed in four steps. First, the coordinates of the free cells at the border of any obstacles gC_{border} in the workspace are extracted and stored into the First-In-First-Out (FIFO) list L_{B}. gC_{border} is defined as any free cells on the map which has at least one $gC_{occupied}$ as its neighbour. Second, the unsafe regions are computed. The unsafe region includes all the cells that are less than α distance away from the border of the obstacle. The unsafe regions are then filled up with occupied cells. This will prevent the generated path from intersecting the unsafe regions. Third, the *navigation function* N is computed in the rest of gC_{free} using the "*wave-front expansion*" algorithm explained in section 2.2.3. Fourth, a path that connects the current position of the robot to the goal is computed based on the *improved navigation function*. The coordinates of all the grid cells that constitute the path are stored in a List L_{p}. The pseudo code for the computation of the path generated by the *improved navigation function* is as follows:

Step 1 – Computing the cells that belong to border of the obstacles

1. begin
2. for every gC_{free} in the map do
3. if there exist a $gC_{occupied}$ neighbour then
4. begin
5. insert the coordinate of gC_{free} to the end of L_{B};
6. end;
7. end;

Step 2 – Computing the unsafe regions

1. begin
2. for *i* = 0,1,2,….., until $L_{\mathrm{B}i}$ is empty do
3. for every neighbour of the cells in L_{B} do
4. if the neighbour is a free cell and less than α distance away then
5. begin
6. neighbour ← occupied (let unsafe cell be occupied);
7. insert coordinate of neighbour at the end of L_{B};
8. end;

end;

Step 3 – Computing the improved navigation function

begin
for every gC_{free} in the map do
    N ← M (large number);
N for goal cell ← 0;
insert coordinates the of goal cell to the end of L_0;
for *i* = 0,1,2,….., until L_{0i} is empty do
    for every 1-neighbour of cells in L_0 do
        if N = M then
            begin
                N ← *i* +1;
                insert coordinate of 1-neighbour at the end of L_0;
            end;
end;

Step 4 – Computing the path joining the current position of the robot to the goal

begin
temp ← coordinates of the cell currently occupied by the robot;
insert temp to the end of L_p;
while N for temp $\neq$ 0 do
    temp ← coordinate of *min*(neighbouring cells of temp)[2];
    insert temp to the end of L_p;
end;

[2] Refer to Appendix 2 for the description of the *min* function.

Figure 4.1 shows a path generated by the *improved navigation function* with α = 1.0. The black cells are the obstacles and the grey cells are the unsafe regions. The path generated by the *improved navigation function* is no longer the shortest path between the starting and the goal point. Nevertheless, the *improved navigation function* always guarantees a path between the starting and the goal point. Figure 4.2 shows a different path generated with a different α for the same workspace as in Figure 4.1.

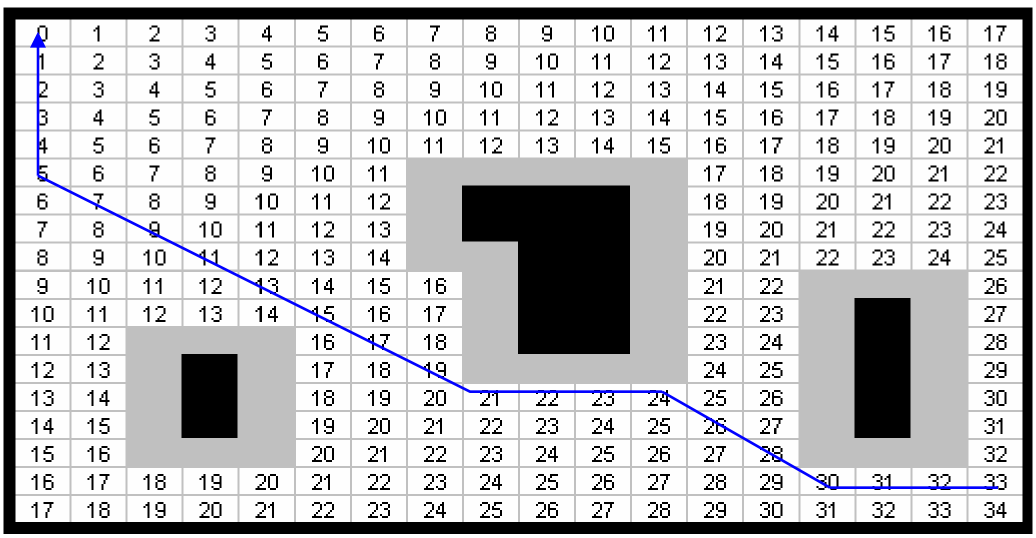


Figure 4.1: Path generated by improved navigation function, α = 1.0

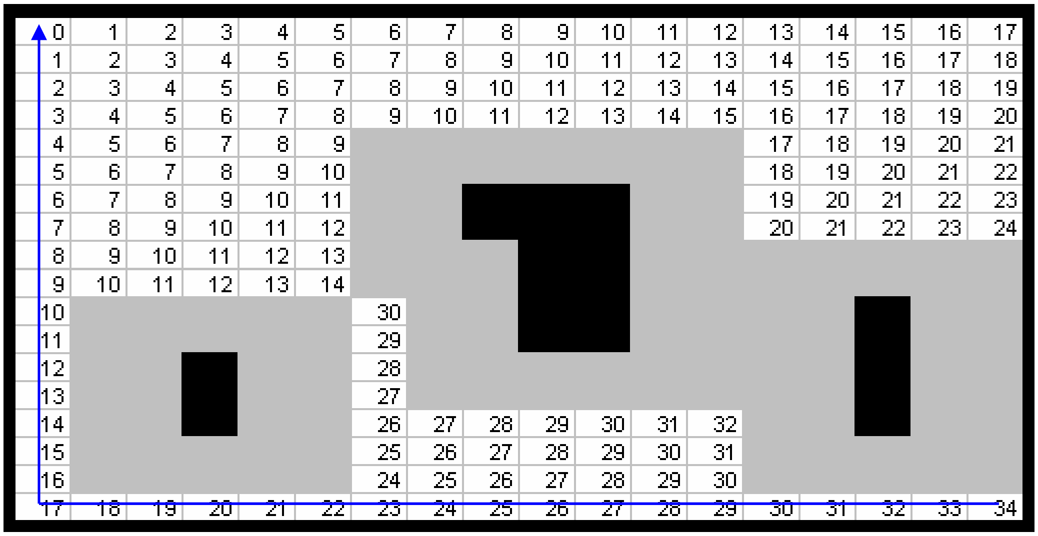


Figure 4.2: Path generated improved navigation function, α = 2.0

4.3 Potential Field Method

As mentioned in section 2.2.5, the robot is represented as a particle in the workspace moving under the influence of an artificial potential produced by the goal configuration and the obstacles. The following sub-sections will give the mathematical equations that describe the *potential field* method.

4.3.1 Potential Functions

The goal generates an attractive potential $U_g({}^{\mathbf{W}}\mathbf{P}_{\mathbf{A}})$ that attracts the robot and the obstacles generate repulsive potential ${}^{i}U_o({}^{\mathbf{W}}\mathbf{P}_{\mathbf{A}})$ that push the robot away. The total potential acting on the robot is thus given by,

$$U({}^{\mathbf{W}}\mathbf{P}_{\mathbf{A}}) = U_g({}^{\mathbf{W}}\mathbf{P}_{\mathbf{A}}) + \sum_{i=0}^{n-1} {}^{i}U_o({}^{\mathbf{W}}\mathbf{P}_{\mathbf{A}}) \tag{4.1}$$

where n equals to 24 which is the number of ultrasonic sensors readings and ${}^{\mathbf{W}}\mathbf{P_A} = (x \quad y \quad 0)^T$ is position of the robot with respect to F_W as mentioned in section 3.3.

Let the goal position be denoted by the vector ${}^{\mathbf{W}}\mathbf{P_g} = (x_g \quad y_g \quad 0)^T$. The attractive potential is therefore defined by,

$$U_g({}^{\mathbf{W}}\mathbf{P_A}) = \frac{1}{2}k_g({}^{\mathbf{W}}\mathbf{P_A} - {}^{\mathbf{W}}\mathbf{P_g})^T({}^{\mathbf{W}}\mathbf{P_A} - {}^{\mathbf{W}}\mathbf{P_g}) \tag{4.2}$$

where k_g is the gain of the attractive potential. Equation (4.3) defines the repulsive potential.

$$ {}^{i}U_o({}^{\mathbf{W}}\mathbf{P_A}) = \begin{cases} \frac{1}{2}k_o(\frac{1}{d_i} - \frac{1}{d_0})^2 & \mathit{if}\ d_i < d_o \\ 0 & \mathit{otherwise} \end{cases} \tag{4.3}$$

where d_i is the scalar distance between the robot and the obstacle read by the i^{th} ultrasonic sensor. The repulsive potential will only have effect on the robot when the robot moves to a distance d_i which is lesser than d_0. This implies that d_0 is the minimum safe distance from the obstacle that the robot tries to maintain. k_o is the gain of the repulsive potential.

4.3.2 Force Functions

The artificial force on the robot is the negated gradient of the total potential function which is given by equation (4.4),

$$\vec{F}({}^{\mathbf{W}}\mathbf{P_A}) = -\nabla U({}^{\mathbf{W}}\mathbf{P_A}) \tag{4.4}$$

Hence, from equation (4.2), the attractive force generated by the goal is given by,

$$\vec{F}_g({}^{\mathbf{W}}\mathbf{P_A}) = -k_g({}^{\mathbf{W}}\mathbf{P_A} - {}^{\mathbf{W}}\mathbf{P_g}) \tag{4.5}$$

Equation (4.5) shows that the attractive force generated by the goal is a function of the difference between the current robot's position and the goal position. This means that the attractive force will be reduced to zero when the robot reaches the goal.

From equation (4.3), the repulsive force generated by the obstacles is given by,

$$ {}^{i}\vec{F}_o({}^{\mathbf{W}}\mathbf{P_A}) = \begin{cases} \left[\ \frac{1}{2}k_o(\frac{1}{d_i} - \frac{1}{d_0})\frac{1}{d_i^2}\cos\theta_i \quad \frac{1}{2}k_o(\frac{1}{d_i} - \frac{1}{d_0})\frac{1}{d_i^2}\sin\theta_i \quad 0\right]^T & \text{if } d_i < d_o \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

where θ_i is the angle that ultrasonic sensor *i* makes with the x-axis of F_W. Figure 4.3 shows the orientation of the reading d_i returned by senor *i*. γ_i is the angle that sensor *i* makes with the robot's frame F_A. β is the angle that F_A makes with F_W. θ_i is therefore the sum of γ_i and β. Equation (4.6) shows that the repulsive force increases in a parabolic way as the robot moves near to an obstacle.

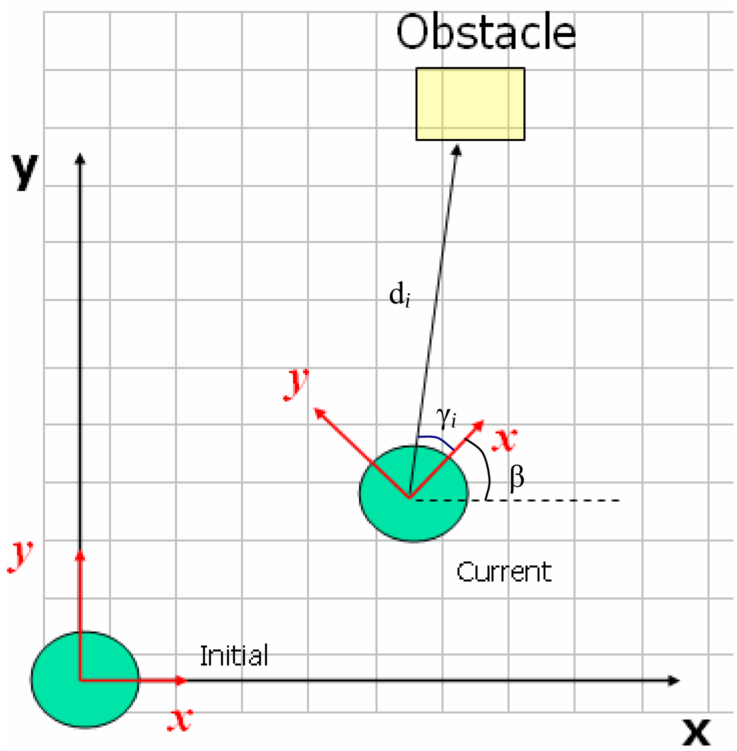


Figure 4.3: Orientation of ultrasonic sensor *i* is given by sum of γ_i and β

The resultant force that will guide the robot through a collision free path to its destination is the sum of the attractive and the repulsive force which is given by,

$$\vec{F}_R({}^{\mathrm{W}}\mathbf{P}_{\mathbf{A}}) = \vec{F}_g({}^{\mathrm{W}}\mathbf{P}_{\mathbf{A}}) + \sum_{i=0}^{n-1} {}^{i}\vec{F}_o({}^{\mathrm{W}}\mathbf{P}_{\mathbf{A}}) \tag{4.7}$$

4.4 Frontier-Based Exploration

The *frontier-based exploration* [9] is a method proposed by Brian Yamauchi to do map building in an unknown environment. In the frontier-based exploration method, the robot first builds a local map of its surrounding in its initial position. The boundary of free space and unknown region is known as the *frontier*. When the robot moves to the nearest frontier, it can see into unexplored space and add the new information to its map. As a result, the mapped territory expands, pushing back the boundary between the known and unknown regions. Hence, the robot can constantly increase its knowledge of the world by moving to successive frontiers.

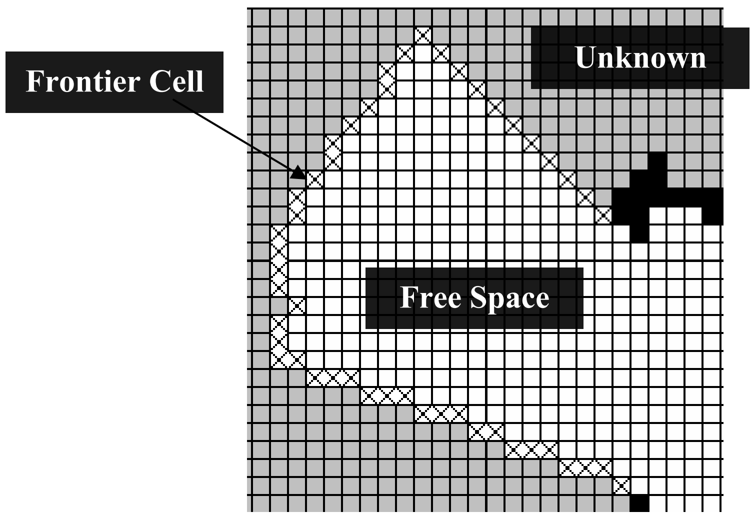


Figure 4.4: A frontier region made up of a group of adjacent frontier cells

A frontier is made up of a group of adjacent frontier cells. The frontier cell is defined as any gC_{free} cell on the map with at least two $gC_{unknown}$ cells as its immediate neighbour. The total number of frontier cells that make up a frontier must be larger than the size of the robot to make that frontier valid. Figure 4.4 shows an example of a valid frontier.

When the robot moves to a valid frontier, it is actually moving towards the centroid of the frontier. The coordinates (${}^{j}x_c, {}^{j}y_c$) of the centroid can be calculated using equation (4.8) and (4.9).

$$ {}^{j}x_c = \frac{\sum_{i=1}^{n_j} x_i}{n_j} \tag{4.8} $$

$$ {}^{j}y_c = \frac{\sum_{i=1}^{n_j} y_i}{n_j} \tag{4.9} $$

Where

n_j = number of frontier cells in frontier j

x_i = x coordinate of i^{th} frontier cell in frontier j

y_i = y coordinate of i^{th} frontier cell in frontier j

4.5 Integrated Algorithm

The *Integrated algorithm* is a method proposed by the author to give the robot all the competencies[3] needed to achieve autonomy in one single framework. The algorithm modifies the *frontier based exploration* method which was originally used for map building[4] into a path planning algorithm. This modified *frontier based exploration* method is then combined with the *improved navigation function* and the *potential field* method into a single framework. The *integrated algorithm* was successfully implemented on the Normad XR4000 mobile robot.

[3] Refer to Section 1.1 for the three competencies needed for mobile robot autonomy.
[4] See section 4.4 for the original frontier-based exploration that is meant for map building.

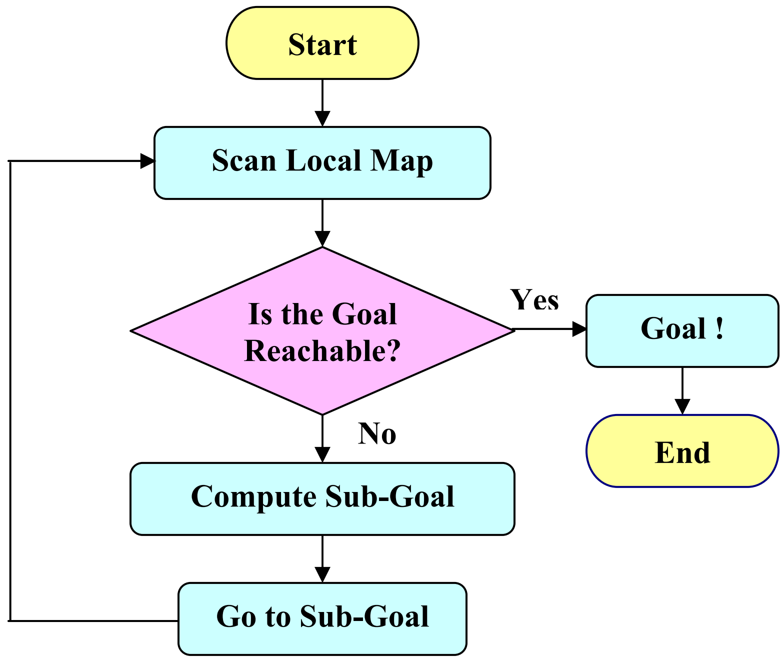
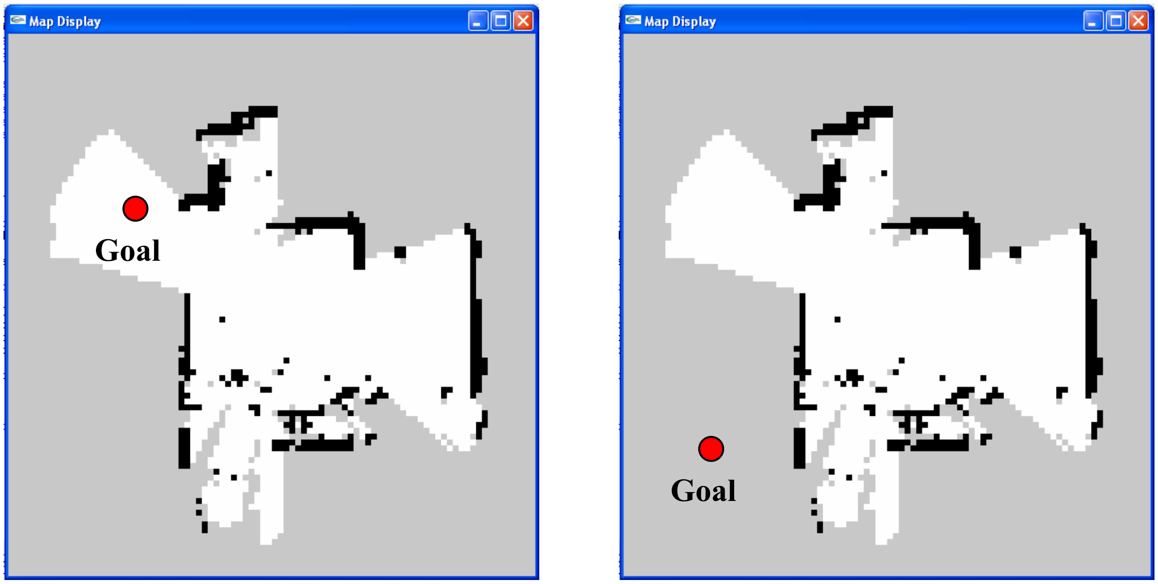


Figure 4.5: The integrated algorithm

Figure 4.5 shows an overview of the *integrated algorithm*. The mobile robot will first build a local map of its surrounding. It then decides whether the goal is reachable. It will advance towards the goal if it is reachable, and to compute for another sub-goal if it is not reachable. The robot will do another local map building after it has reached the sub-goal[5]. This process goes on until the goal is reached. The following sections will discuss the algorithm in detail.

[5] Note that all the local maps built are combined to form a larger map stored in the memory.

4.5.1 Goal Reachability



(a) **(b)**

Figure 4.6: (a) The goal is reachable because it is in the gC_{free} region (b) The goal is unreachable because it is in the $gC_{unknown}$ region

When the robot finished the map building process, it will determine whether the goal is reachable based on the map. The goal is reachable if it is in the gC_{free} region and not reachable if it is in the $gC_{unknown}$ region. Figure 4.6 (a) shows an example of a reachable goal in the gC_{free} region (white region) and (b) shows an example of an unreachable goal in the $gC_{unknown}$ region (grey region).

4.5.2 Computation of Sub-Goal

The robot will compute a sub-goal if the goal is unreachable. This is done in three steps. First, compute the path that joins the robot's current position and the goal using the *improved navigation function*. The unknown cells are taken to be free space in the computation of the *improved navigation function*. Second, all the frontiers in the map are computed. Third, the centroid of the frontier that intersects the NF path will be selected as the sub-goal. The reason for using the *improved navigation function* in computing the sub-goal is that the computed sub-goal will always be the most efficient point in getting to the goal.

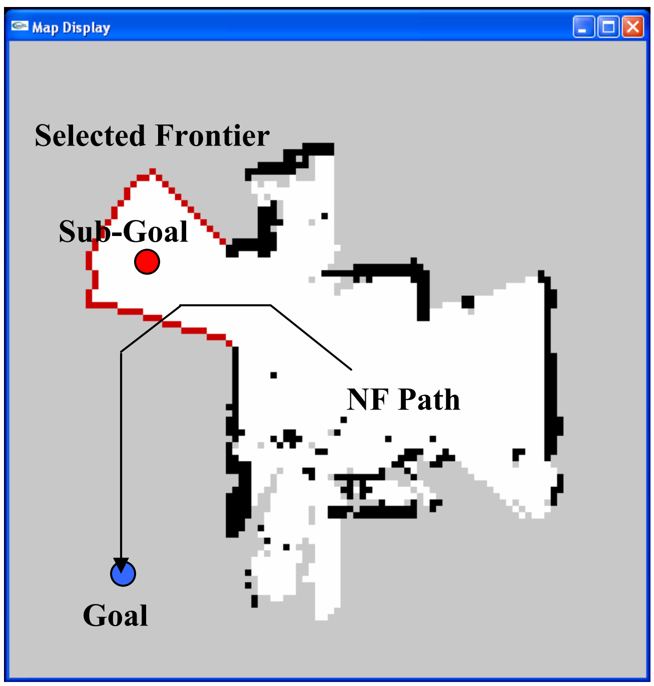


Figure 4.7: Sub-goal is the centroid of the frontier that intersects the NF path

Figure 4.7 shows an example of the computation of the sub-goal. The centroid of the frontier that intersects the NF path is selected as the sub-goal.

4.5.3 Reaching for the Sub-Goal

After computing the sub-goal, the robot will moves towards it using the *hybrid method* [10]. Figure 4.8 shows an illustration of the hybrid method. The robot first computes the path joining its current position to the sub-goal using the *improved navigation function*. The robot then places a circle with an empirical radius centered at its current position. The cell that corresponds to the intersection of the circle with the NF path is known as the attraction point. The attraction point is the cell with the lowest value if there is more than one intersection.

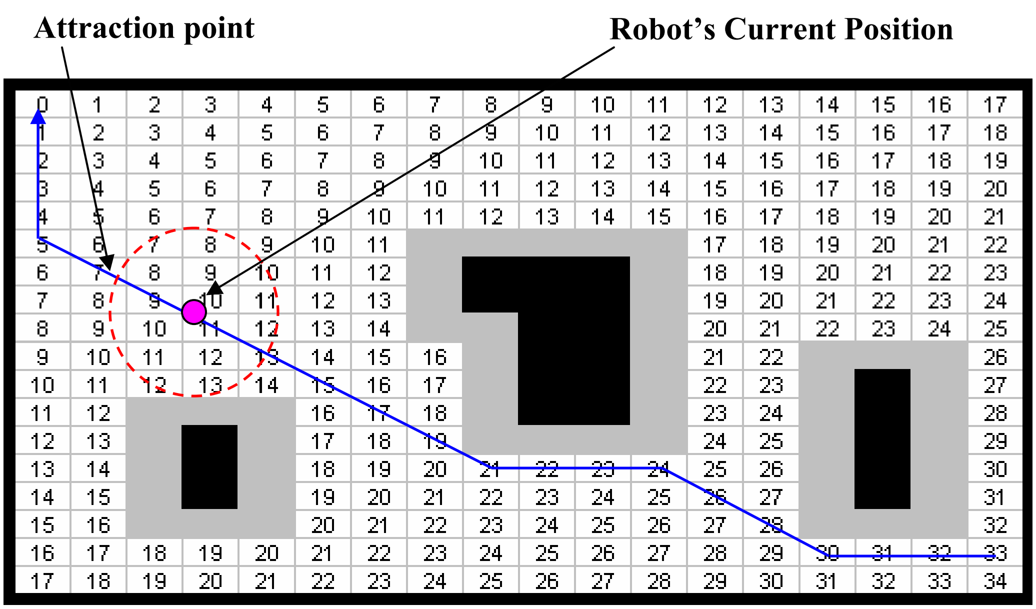


Figure 4.8: Illustration of the hybrid method

The robot advances towards the attraction point using the potential field method. However, the circle moves along with the robot which causes the attraction point to change too. As a result, the robot is always chasing after the dynamic attraction point which will progress towards the sub-goal along the local minima free NF path. The radius of the circle will become larger in cases where no intersections are found. It will however become smaller when the robot is near to the sub-goal.

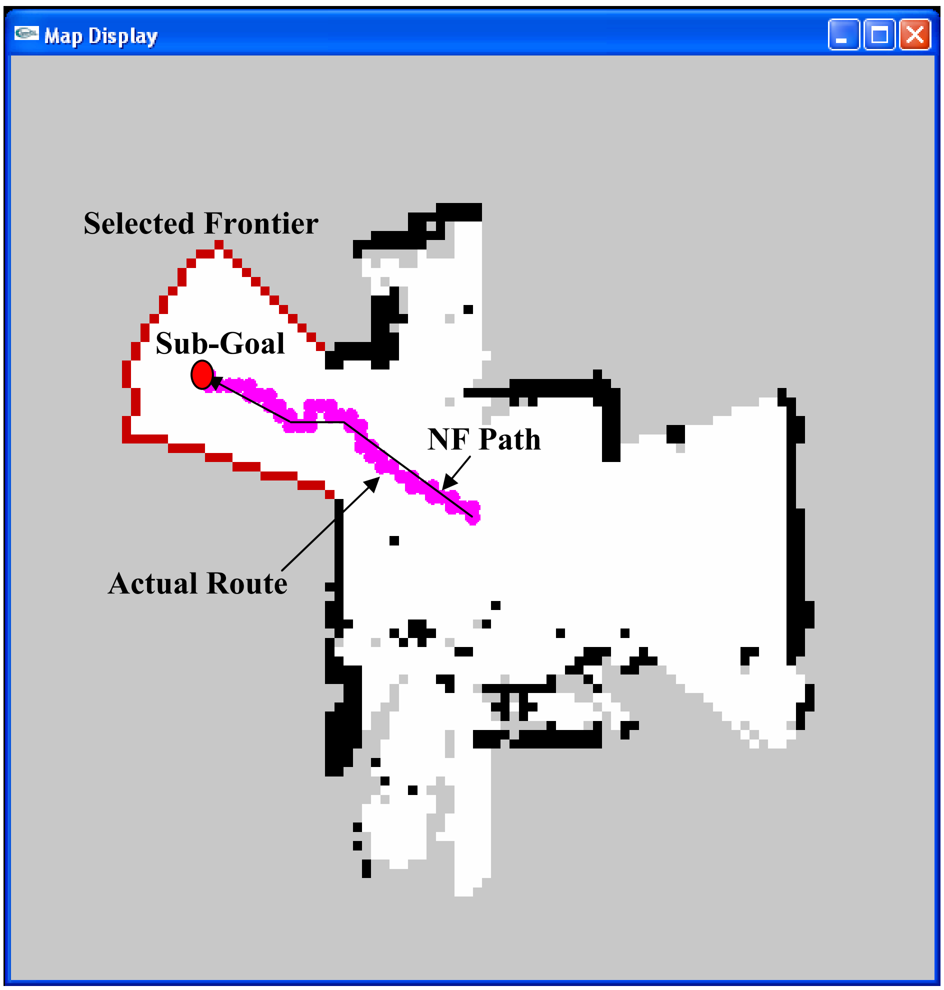


Figure 4.9: Actual route taken by robot during run-time

Figure 4.9 shows the actual route taken by the robot using the hybrid method compared to the NF path.

4.5.4 Reaching for the Goal

As mentioned in section 4.5, the robot will build another local map when it has reached the sub-goal. This local map is added to the previous map to form a larger map.

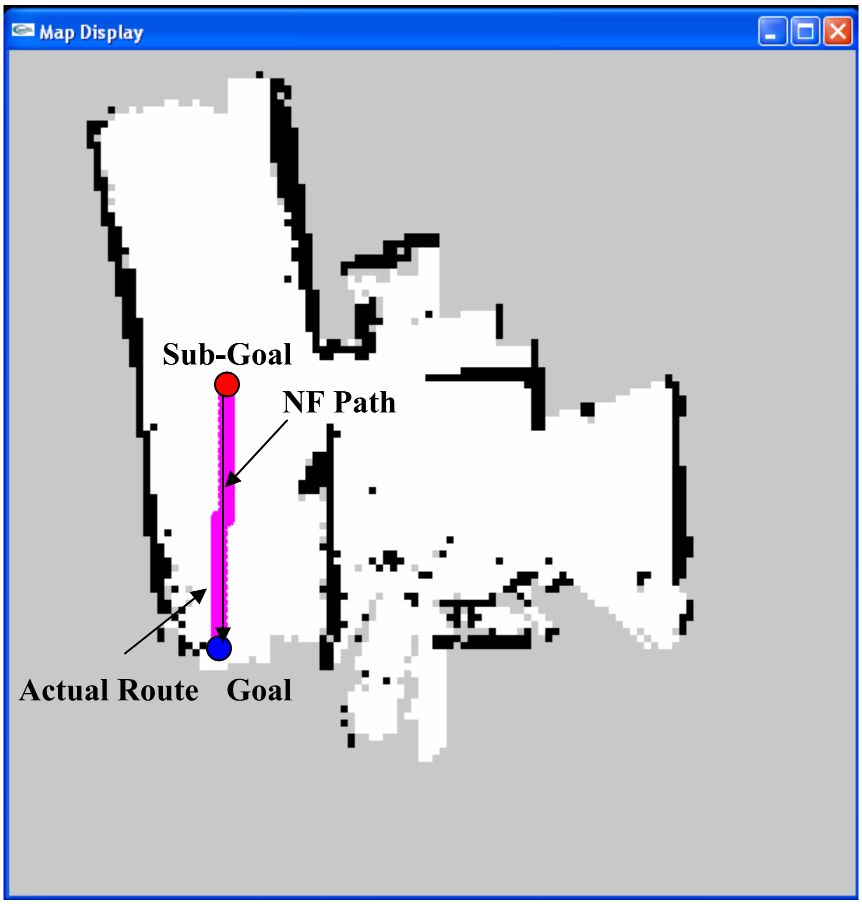


Figure 4.10: The robot moving towards the goal during run-time

Figure 4.10 shows the larger map formed by the robot at the sub-goal during run-time. In this case, the robot deduced that the goal is reachable because the goal is in the gC_{free} region. The robot then plans a path that joins its current position to the goal using the *improved navigation function*. Finally, it will proceed to the goal using the *hybrid method*.

4.6 Implementation and Results

The *integrated algorithm* was successfully implemented on the Nomad XR4000 mobile robot. Video clip of the implementation results can be found from http://guppy.mpe.nus.edu.sg/~mpeangh/lee-gim-hee/integrated1.mpg. The video shows the robot, which is placed in an initial position inside a room, navigating to the given goal along the corridor that is outside the room using the *integrated algorithm*. Notice that the robot has to stop at the various sub-goals to do further mappings into the unknown regions. The video shows that the robot is capable of moving through the narrow door opening of the room and it is also capable of avoiding dynamic obstacles (i.e. human beings) that are blocking its path.

Chapter 5

Conclusion

5.1 Epilogue

The ability to navigate safely from one point to another is the most important part in the development of autonomous mobile robots. A vast number of techniques and methods have been introduced and implemented by many researchers. In this thesis, only a few methods are examined and implemented. They include the *improved navigation function* and the *potential field* method.

The contribution of this thesis is the *integrated algorithm* proposed and implemented on the Nomad XR4000 mobile robot by the author. This method modifies the *frontier based exploration* method, which was originally a map building technique, into a path planning algorithm. In the *integrated algorithm*, the *improved navigation function* and the *potential field* method are fused together with the modified frontier based exploration method into one single framework. The algorithm is capable doing path planning in an unknown environment. The *improved navigation function* is used to plan the path and hence local minima free. The algorithm is also capable of avoiding any dynamic obstacles which were not included in the path planning.

The *grid-based representation* of map building is also discussed and implemented. Laser range finder and ultrasonic sensors were used in the map building process. The map was updated by heuristic method and noises are present. Nevertheless the accuracy of the map is still sufficient for the mobile robot to rely upon for navigation.

5.2 Further Work

In chapter 1, two assumptions for this dissertation have been made. They were:

i. It is assumed that the odometry reading is accurate and hence no localization techniques are required.

ii. It is also assumed that the starting point and the goal belong to free space and a path connecting both positions always exists.

It must be noted that these two assumptions may not be true in the navigation of mobile robot in the real world.

The odometry reading by counting the wheel revolution is not accurate after the robot has traveled a long distance. This is due to wheel slippages and drift that occur in the mobile robot. Figure 5.1 shows a map generated without localization. Notice that the corridor is slanted. This is due to errors in odometry readings in the orientation of the robot. This error will grow as the robot travels a longer distance and the generated map will become less accurate.

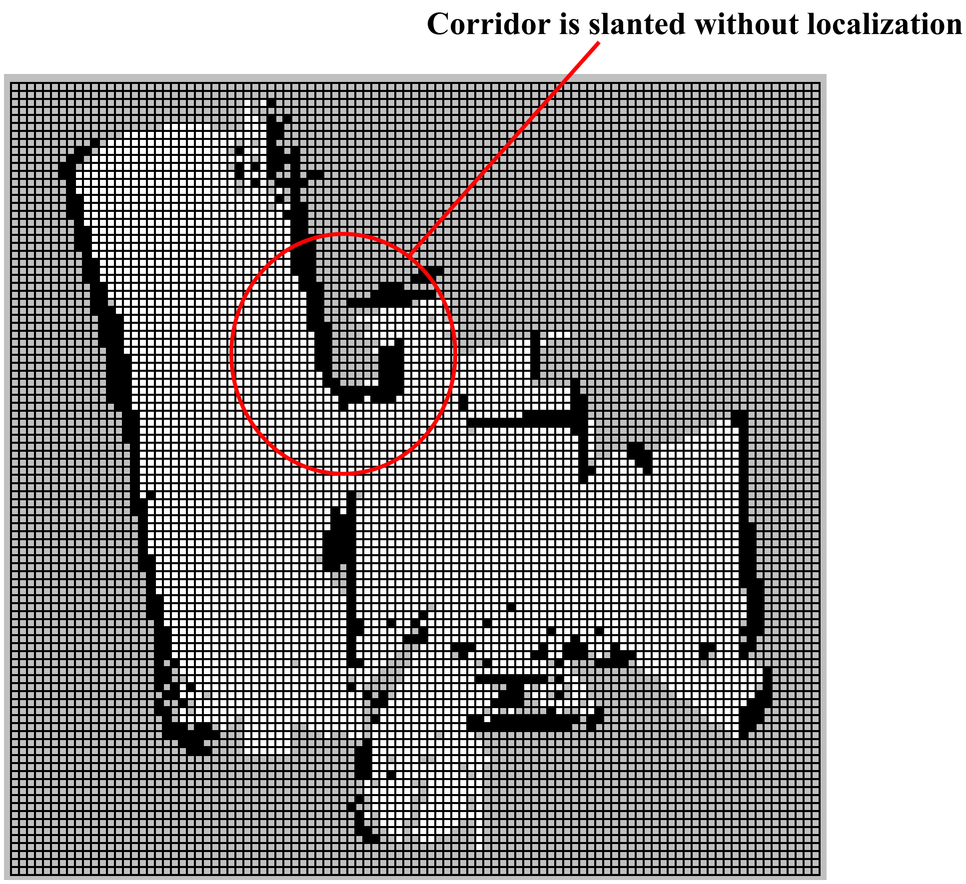


Figure 5.1: Map generated without localization

The solution to the odometry error is to do localization. Various mathematical tools such as *Kalman Filter* [11] can be used.

When a mobile robot is navigating in the real world, a path connecting the starting point and the goal point may not always exist. Further work is needed to be done to solve this problem. The robot must be able to conclude that the goal is unreachable in cases where there is no path connecting the starting point and the goal point.

References

[1] J.-C Latombe, “Robot Motion Planning. Boston”, Kluwer Academic Publishers. 1991.

[2] V. Akman, “Unobstructed Shortest Paths in Polyhedral Environments”, Lecture Notes in Computer Science, Springer-Verlag, 1987.

[3] C. ‘Dnliang and C Yap, “Retraction: A New Approach to Motion Planning”, Proceeding of the 15th ACM Symposium on the Theory of Computing, Boston, pp. 207-220, 1983.

[4] J. Schwartz and M. Sharir, “On the Piano Movers’ Problem: The Case if a Two-Dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers”, Communications on Pure and Applied Mathematics, Vol. 36, pp. 345-398, 1983.

[5] J.Borenstien and Y.Koren, “The Vector Field Histogram – Fast Obstacle Avoidance For Mobile Robots”, IEEE Journal of Robotics and Automation Vol 7, No 3, June 1991, pp. 278-288.

[6] O.Khatib, “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots”, International Journal of Robotic Research, Vol. 5, No. 1, pp.90-98, 1986.

[7] Y.Koren and J.Borenstien, “Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation”, Proceedings of the IEEE Conference on Robotics and Autonomon, Sacramento, California, pp.1398-1404, April 7-12, 1991.

[8] Borenstein, J. and Koren, Y., “Histogram In-motion Mapping for Mobile Robot Obstacle Avoidance”, IEEE Journal of Robotics and Automation, Vol 7, No. 4, 1991, pp. 535-539.

[9] Brian Yamauchi, “A Frontier-Based Approach for Autonomous Exploration”, Proceedings of the IEEE International Symposium on the Computational Intelligence in Robotics and Automation, Monterey, CA, pp. 146-151.

[10] Lim Chee Wang, "Motion Planning for Mobile Robots", Thesis for Master of Engineering, National University of Singapore, 2002.

[11] G. Welch and G.Bishop, "An Introduction to the Kalman Filter", Paper TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175, February 1995.

1. Definition of 1-Neighbour

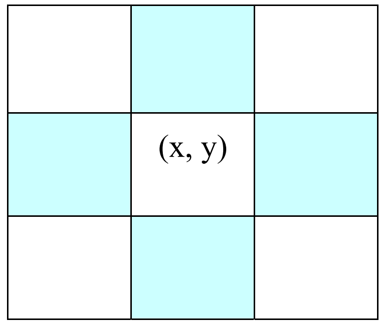


Figure A1.1: The shaded cells are the 1-Neighbour of the cell (x, y)

2. Description of *min* function

2	3	4
1	(x, y)	5
8	7	6

Figure A2.1: Priority of neighbouring cells of (x,y)

The *min* function returns the coordinate of the neighbouring cell of (x, y) that has the lowest N value. However, in cases where there are more than 1 neigbouring cells having the same lowest N value, the cell with the highest priority will be chosen. Figure

A2.1 shows the eight neighbouring cells of (x, y) labeled with their respective priority. The cell with ‘1’ has the highest priority and the cell with ‘8’ has the lowest priority.