

Communication Model and Optical Flow to Aid Collaborative Robotics Search Strategy

by

Swathi Dumpala Basaveswara

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
August 3rd, 2013

Keywords: Collaborative robotics, Manhattan mobility model, Search and rescue,
Communication, Robot navigation, Optical flow

Copyright 2013 by Swathi Dumpala Basaveswara

Approved by

Thaddeus Roppel, Chair, Associate Professor, Electrical and Computer Engineering
Prathima Agrawal, Samuel Ginn Distinguished Professor, Electrical and Computer Engineering
Shiwen Mao, Assistant Professor, Electrical and Computer Engineering

Abstract

This thesis provides an analysis of how communication among robots improves search performance in a multi-robot search and rescue scenario. Mobility models have been used to represent the movement patterns in vehicular and mobile ad-hoc networks. As more and more collaborative robotics applications employ the use of these models, there is a need for them to be better suited for practical applications, and communications is imperative for these operations. Along with exploring the advantages of communications between the mobile robots, this thesis also gives an overview of an optical flow algorithm to aid in robot navigation. This algorithm was designed to be used in a busy environment which holds a lot of moving obstacles. The traditional mobility models have been changed a little with the inclusion of obstacles to suit the experiments and need of collaborative robotics applications. The key results of this thesis are that incorporating communications between the robots reduces the time taken for target detection by a considerable value. The results also explore the change in the time taken to reach the target with change in a set of parameters, like the number robots, number of obstacles, sensor range. The results of the optical flow process show a disparity map generated by the robot, which helps it in analyzing the environment it is a part of, and decide its next move accordingly.

Acknowledgments

Firstly, I would really want to thank my advisor Associate Professor, Dr. Thaddeus A. Roppel, without whose guidance this thesis would not have been possible. I also specially want to thank him for his support throughout my Master's program, and for giving me the confidence to pursue my research interests. I am very fortunate to have him as my advisor. A very special thanks to Dr. Prathima Agrawal for always showing confidence towards my work, for being my support system throughout and for her invaluable advice and financial support provided during my Masters program. I thank Dr. Shiwen Mao for accepting to be a part of my committee and for taking time out from his busy schedules.

I am deeply indebted to my parents, my brother, Santosh, my granddad and family for always being there and motivating me throughout. They are the reason I am what I am today. I would also like to thank Brian Pappas and Abhishek Bichal for helping me with the data needed for my thesis, and also a special thank you to my seniors, Nida Bano and Gopal Iyer for guiding me with my thesis. A big thank you to my friends here: Matthew, Jasma, Sanjay, Vignesh, Rizwan, Poojita, Mradul, Nikhil, Swathi, Vaishnavi, Prardiva, Kalyani, Aditi, Rohit, Bhavadeep, Kartik and many others for keeping me entertained and not miss home much. And a very special thank you to my friends back home: Akila, Dhvani, Sneha, Wilma, Rachna, Divya, Jahnavi, Veda, Amulya, Nitin, Lakshay for being there for me during my tough times and making my life in the USA very easy.

Table of Contents

Abstract	ii
Acknowledgments	iii
List of Figures	vii
List of Tables	ix
Chapter 1: Introduction	1
Chapter 2: Literature Review	5
Part 1: Optical Flow	
2.1) Brightness (grey value) constancy	9
2.2) Gradient Constancy	10
2.3) Smoothness Assumption	10
2.4) Multiscale Approach	12
Part 2: Communication model	
2.5) Co-operative Robotics	12
2.6) Communication in Co-operative Robotics	14
2.7) Search and Rescue operations in Co-operative Robotics	16
2.8) Mobility Models	16
2.9) Simulation Environments	18
Chapter 3: Laboratory Setup and Robot Architecture	20

3.1) Laboratory Setup	20
3.2) Robot Architecture	21
3.2.1) Chassis	22
3.2.2) Drive and Control System	22
3.2.3) Vision System	23
3.2.4) Batteries	24
Chapter 4: Optical flow	25
4.1) Objective	25
4.2) Disparity Map	25
4.2.1) Image rectification	27
4.2.2) Matching and Filtering	28
4.3) Simulation Flowchart	29
Chapter 5: Manhattan Mobility Model and Communication Model	31
5.1) Manhattan Mobility model	31
5.2) Simulation model fundamentals	33
5.2.1) Target Detection	33
5.2.2) Obstacle avoidance	33
5.2.3) Communication	33
5.2.4) Collision avoidance	34
5.2.5) Boundary Conditions	34
5.3) Simulation Model Developed	35
5.4) Communication Model Developed	36
5.4.1) Sensor Range	38

Chapter 6: Results	39
Part 1: Optical flow	
6.1) Simulation setup	39
6.2) Result	39
Part 2: Communication model for Robot Collaboration	
6.3) Simulation setup	42
6.4) Simulation parameters	42
6.5) Results and Analysis	43
6.5.1) Varying the communication intervals.....	43
6.5.2) Varying the number of robots.....	44
6.5.3) Varying the sensor range	45
6.5.4) Varying the number of obstacles	47
6.5.5) Varying the number of robots and obstacles to maximum values.....	51
Chapter 7: Conclusions and future work	54
7.1) Conclusions	54
7.1.1) Optical flow	54
7.1.2) Communication model	54
7.2) Suggestions for future work	56
References	57

List of Figures

Figure 2.1 Image sequence with fragmented occlusion a)first image b) constraint lines.....	11
Figure 2.2 Classification of mobility models.....	18
Figure 3.1 The map a) being generated by the robot, b) inbuilt in the system	21
Figure 3.2 Team of robots exploring 3 rd floor of Broun Hall, Auburn University	21
Figure 3.3 The robot control architecture	22
Figure 3.4 The Drive and Control system	22
Figure 3.5 Vision system, Kinect sensor	23
Figure 4.1 Top: Left and right stereo images Bottom: Disparity map of stereo images.....	26
Figure 4.2 Flow chart for creating a disparity map.....	26
Figure 4.3 Search space before (1) and after (2) the rectification process	27
Figure 4.4 Input images, point correspondences and matching between the images	29
Figure 4.5 Disparity map with window size 5.....	29
Figure 4.6 Simulation flowchart for optical flow process	29
Figure 5.1 Depiction of the movement of the nodes using Manhattan Mobility Model	32
Figure 5.2 Travelling pattern of a robot using Manhattan Mobility Model.....	32
Figure 5.3 Closer look at the travelling pattern of a robot using Manhattan Mobility Model....	35
Figure 5.4 Simulation flowchart for Manhattan Mobility Model used in thesis	36
Figure 5.5 Flowchart of communication model developed	37
Figure 5.6a) Range of view of robot with a regular sensor range.....	38

Figure 5.6b) Range of view of robot with a special sensor range.....	38
Figure 6.1 Input images a) at the starting point b) after progressing a few steps	40
Figure 6.2 Output of the optical flow process, Disparity map.....	40
Figure 6.3 Histogram plotted for number of attempts in running code to the number of steps taken for target detection	44
Figure 6.4 Graph showing the change in the number of steps taken for target detection with change in number of robots with one-step look-ahead	45
Figure 6.5 Graph showing the change in the number of steps taken for target detection with change in number of robots, with sensor range 5	47
Figure 6.6 Graph plotted for number of steps taken to reach target with change in sensor range.....	48
Figure 6.7 Graph plotted for number of steps taken by 5 robots to reach target with varying number of obstacles	50
Figure 6.8 Graph plotted for number of steps taken by 5 robots to reach target with varying number of obstacles	51
Figure 6.9 Graph plotted for number of steps taken to reach target to number of obstacles for extreme values of number of robots and obstacles	52

List of Tables

Table 6.1 Simulation parameters	42
Table 6.2 Simulation parameters set for varying communication intervals	43
Table 6.3 Simulation parameters set for varying the number of robots.....	45
Table 6.4 Simulation parameters for change in sensor range	48
Table 6.5 Simulation parameters for varying the number of obstacles	49

CHAPTER 1

INTRODUCTION

A mobile robot is basically a machine that has an on-board computer, and can move around to accomplish certain tasks. Cooperative mobile robotics involves a group of mobile robots, teaming up to perform and achieve a set of goals.

“The first real research on search and rescue robot began in the aftermath of the Oklahoma City bombing in 1995”[2]. Robots made their first appearance in actual human search and rescue operation on September 12, 2001, “Mankind’s largest push to develop robots arguably began in the days immediately following the 9/11 terrorist attacks, when shoebox-size “PackBots” made by iRobot Corporation (best known for its semi-autonomous Roomba vacuum cleaner) were used to ensure the stability of rubble piles before first responders started their search for survivors at New York’s World Trade Center site” [1]. These robots were of different sizes and performed various different tasks. Sources say that these robots actually discovered more than 2 percent of the victims, on that day [3]. It was then that collaborative robotics became established as a separate field of study.

Collaborative robotics can benefit greatly from the incorporation of wireless communications. It enables collaborative path planning and navigation. The use of a collaborative system constitutes a highly efficient solution, particularly for difficult tasks. Collaborative robotics has many advantages: Costs can be reduced, Throughput times can be reduced and processes can be simplified.

One of the applications of collaborative robotics is load sharing. This involves two robots sharing the load of a certain work. The use of multiple robots can be more economical than deploying a single heavy-duty robot [4]. Various experiments in the past have proved this very point. References [5, 6] describe how a collaborative system of robots outperformed many single robot systems designed to perform the same task. They also state that a multiple robot system also helps in efficient localization if the robots are communicating or exchanging information with each other.

This thesis addresses the importance of collaborative robotics and how communication between the robots can effectively reduce the target detection time, while improving localization and obstacle avoidance. Vision based navigation is also discussed in the thesis.

Applications of vision-based systems widely range from object recognition [7], obstacle avoidance [8], navigation [9], to global localization [10] and SLAM. In the present research, a group of robots are all working together based on a cooperative distributed protocol called SARA, which is an acronym for Search and Rescue Algorithm.

SARA incorporates the concepts of a relatively new area of collaborative robotics [11]. This algorithm works on tackling the several smaller problems like communication

between the robots, path planning, obstacle avoidance, target detection. There has been a constant effort to improve SARA by incorporating vision in the algorithm. This particular protocol has passed through several phases of modifications from the base version, by addition of new features, removing the defective pieces of the system and also enhancing the performance of the existing parts. Currently, the primary focus would be to perform simulations on vision induced and mutually communicating robots. Another very important factor of SARA would be effective exploration of the search area within a shorter span of time. The designed algorithm should have the ability to explore most parts of the search area in as minimum time as possible. This can be achieved if the robots working collaboratively could communicate with each other through a common medium to get a proper knowledge of the region explored by each robot and then decide their next individual move.

The work described in this thesis proposes the use of a specially-tailored communication model in the Manhattan mobility model to perform Search and Rescue operations. It compares the target detection, obstacle avoidance abilities in the Manhattan mobility model for with and without communications, compares the performance of robots with and without sensors. There is also another part of the thesis that proposes an optical flow algorithm to aid in robot navigation.

Mobility models represent the movement of mobile nodes, and how their location, velocity and acceleration change over time. They have been widely implemented in vehicular ad-hoc networks (VANETs) and mobile ad-hoc networks (MANETs). The Manhattan mobility model is a specific model that decides the next move in a

probabilistic pattern and changes the direction of the nodes travelling in a particular direction at the junctions.

The question arises, “Why incorporate communications in an already existing model?” Since the study in the thesis addresses the Manhattan mobility model being used for collaborative robotics, where each robot is performing the task autonomously, communication between the robots serves to reduce the time taken for target detection and also collision avoidance.

Analysis in this work introduces modifications to the standard model including incorporating obstacles and enabling communication between the mobile robots. These changes are made to make the conventional model more suited for the platform of collaborative robotics.

Optical flow is described as the apparent motion of the brightness pattern in the image sequence [12]. To make search algorithms more practical to use, we need to consider a busy environment where there are dynamic obstacles. In an environment with dynamic obstacles, the robot needs to be able to understand the pattern of the motion of the obstacles and should be able to avoid any sort of collisions. Optical flow proves useful in such an environment as it provides the robot with a disparity map, i.e., a 3 dimensional map of the change in the environment from a previously perceived environment. The simulations shown in the thesis show how such an algorithm can be implemented in vision based robotics.

The remainder of the thesis is organized as follows. Chapter 2 is an overview of the relevant literature. Chapter 3 describes the hardware, the experimental setup for running the simulations in the thesis. Chapter 4 explains the implementation and analysis of the

communication model. Chapter 5 describes the process of optical flow, disparity map generation and the algorithm implemented to aid in robot navigation. The results are explained in Chapter 6. Conclusions and scope for future work is discussed in Chapter 7.

CHAPTER 2

LITERATURE REVIEW

Part 1: Optical flow

Optical flow is defined as the apparent motion of brightness pattern in an image sequence [12]. It is also defined as the process of estimating the motion/velocity field from the spatial and temporal variations of image brightness using computer vision techniques [15]. Optical flow deals with the estimation of the displacement field in two subsequent images. Scientists in the field of computer vision have been working for more than two decades with optical flow algorithms. These algorithms are used in applications ranging from simple motion detection in surveillance cameras to complex applications of gesture controlled gaming like Microsoft Kinect, in anti-collision systems built into cars etc. All these systems demand highly resolved accurate data that is processed in real time with real time speeds for quick decision making.

Optical flow theory/algorithms are based on grey value constancy and smoothness assumptions. All previous works show some treatment of these assumptions. The optical flow algorithms define energy associated with each pixel in an image. When two images are given, optical flow algorithms try to find the disturbance associated with second image with reference to the first image by minimizing the energy difference between

corresponding pixels in the two images. Most optical flow algorithms are different with respect to their energy definition.

Lucas-Kanade method [16] and the Horn-Schunck method [12] are among the first methods used for optical flow determination. Lucas-Kanade method assumes that the velocity ($u(u_x, u_y)$) of pixels in a given neighborhood is constant. This assumption is used in minimizing the error associated with the square of the difference between the brightness/pixel values at corresponding neighborhoods in two images. This result in an over-determined system which is iteratively solved to find the displacement associated with the second image compared to the first image. Iteration process is similar to the Newton-Raphson iteration technique.

$$E = \sum_{y=-w_y}^{w_y} \sum_{x=-w_x}^{w_x} [I_2(x + u_x, y + u_y) - I_1(x + y)]^2 \quad (1)$$

where, $I(x,y)$ denotes the distribution of the brightness values in the images and subscripts 1 and 2 denote image 1 and image 2 respectively, $(2w_x + 1)$ and $(2w_y + 1)$ denote the width of the neighborhood in the x and y direction respectively, and $(0,0)$ denote the center of the neighborhood.

Horn et al.[12] introduced a global method to determine the optical flow between two images. They introduced a global smoothing term for the first time. They argued that any object in the image has a finite size, and when moving, all pixels in a neighborhood move with similar velocities. Thus the introduced smoothness term assumes that the flow of the brightness pattern in the whole image is smooth. They showed that to solve for the optical flow in images two constraints are needed, one is the given by the brightness constancy assumption and the second is provided by the smoothness term. The Horn-Schunck method results in Equation 2.

$$E = \int_D \left[(\nabla I \cdot u + I_t)^2 + \lambda^2 (|\nabla u_x|^2 + |\nabla u_y|^2) \right] dx \quad (2)$$

Here, $I(x,y,t)$ denotes the distribution of brightness in the image, ∇ is the spatial differential, I_t is the time derivative of the brightness distribution, u_x and u_y are the velocity components in x and y directions respectively, and λ is the smoothness term.

Though the Horn-Schunck method helped in tackling the aperture problem, the direction and magnitude of motion cannot be determined by a single oriented unit[17]. The use of the smoothness assumption is bound to inaccuracies when there are discontinuities in the flow. Study by Barron et. al.[18] proved that the Horn-Schunck method is sensitive to noise.

Effect of smoothness was later studied by Nagel et al.[19] In their study they agree on the importance of the global smoothness constraint with Horn-Schunck but to counter the effect of discontinuities Nagel introduced an oriented smoothing constraint. Oriented smoothness constraint restricts variation of the displacement vector only when the gray value variation is small.

Researchers have done comparative performance studies on these optical flow techniques. Fuse et al. [20] studied the comparative performance of the gradient-based optical flow techniques and presented their short comings. Faisal et al [21] investigated the implementation of the optical flow technique presented by Brox et al. [22] They concluded that this method gave an accurate result from only 2 input images. In BOS experiments we investigate for instantaneous flow properties which essentially have an instantaneous image to compare with the reference image, thus we chose to work with the method described by Brox et al. [22]

Let the brightness pattern in an image be denoted by $I(x,y,t)$. Optical flow algorithms are based on the following assumptions:

- Surfaces in the image are flat.
- Illumination is uniform over the imaged area.

2.1) Brightness (grey value) constancy:

This states that any given point in the pattern will have the same brightness even when moved from one location to the other in the imaged scene.

$$I(x, y, t) = I(x + u, y + v, t + 1) \quad (3)$$

The above equation can be simplified by expanding the RHS using the Taylor series and ignoring the higher order terms.

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} \quad (4)$$

$$\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} = I_x u + I_y v + I_t = 0 \quad (5)$$

Horn et.al[1] argue that the above equation gives the velocity/movement component in the direction of the brightness gradient, but fails for directions perpendicular to them. This can be tackled using an appropriate background for in the BOS experiments. Lucas and Brox find the assumption good only when image changes linearly along the displacements and Pappenberg et.al[23] argued that the algorithms using the assumption fail when there is a local or global change in the brightness. All insist on other constraints for the optical flow algorithms to applicable to more general situations.

2.2) Gradient constancy:

As mention by Brox and Pappenberg any slight change in the brightness locally or globally can result in large discrepancies. The gradient of the grey level values will remain constant in case of these changes.

$$\nabla I(x, y, t) = \nabla I(x + u, y + v, t + 1) \quad (6)$$

2.3) Smoothness Assumption:

Anandan [24] states that use of smoothness constraint is to minimize the error associated with the velocity field by using the reliable displacements to determine the less reliable neighbors. In a general scenario neighboring points in a scene either move together or have influence on each other except where there are occlusion or sharp discontinuities, like shocks in a flow. Brox suggests use of the piecewise smooth assumption instead.

From the above discussions the energy of the deviation field measured in a small region of interest from the grey value constancy and the gradient constancy is given by:

$$E_d = \int \left(|I(x + u, y + v, t + 1) - I(x + y + t)|^2 + \gamma |\nabla I(x + u, y + v, t + 1) - \nabla I(x + y + t)|^2 \right) \quad (7)$$

where, γ is the weight function. The energy for the smoothness term is given by:

$$E_s = \int |\nabla u|^2 + |\nabla v|^2 dX. \quad (8)$$

where, $\nabla = (\partial_x, \partial_y, \partial_t)$

Black et. al.[26] showed that the use of quadratic function in energy estimation resulted in unreliable results in areas with multiple motions. To show this they considered an example sequence of images with fragmented occlusion (figure 1a). In this example the man is moving behind the plant from right to left resulting in fragmented occlusion.

Figure 1b shows the constraints obtained from solving the brightness constraints in this example using least square regression model. Trying to satisfy the constraints results in wrong motion (white cross), whereas the gray crosses are the correct motion.

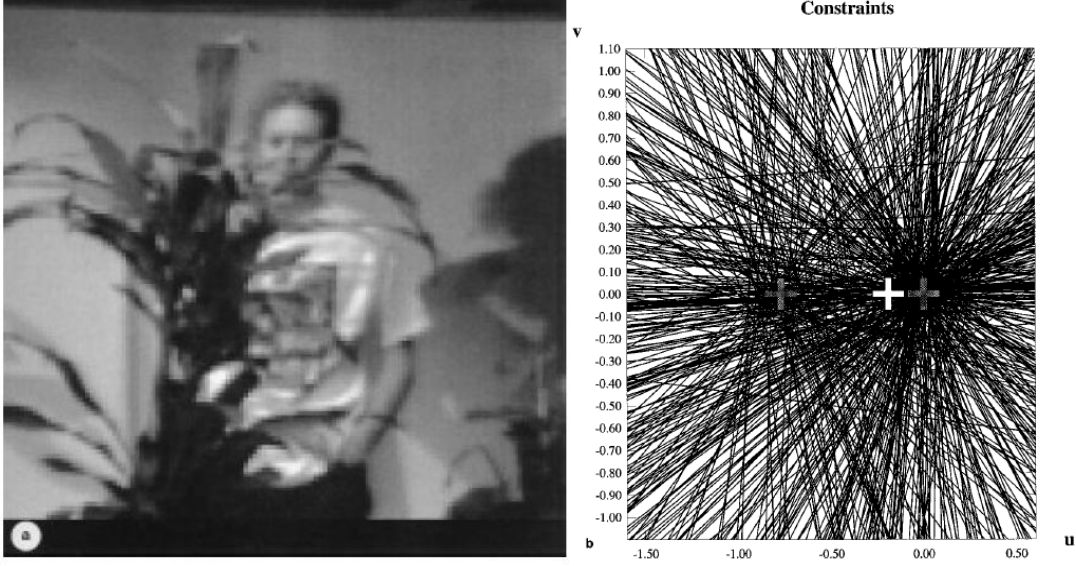


Figure 2.1: Image sequence with fragmented occlusion. (a) First image, (b) Constraint lines.

Instead of a quadratic function a convex function ($\psi(s^2)$) is defined. Where,

$$\psi(s^2) = \sqrt{s^2 + \varepsilon^2} \quad (9)$$

Thus the modified data term is given by:

$$E_d = \int \psi \left(|I(X+w) - I(X)|^2 + \gamma |\nabla I(X+w) - \nabla I(X)|^2 \right) \quad (10)$$

where, $X = (x, y, t)$ and $w = (u, v, 1)$ and the modified smoothness term is given by:

$$E_s = \int \psi \left(|\nabla u|^2 + |\nabla v|^2 \right) dX. \quad (11)$$

Now the total energy is given by:

$$E = E_d + \alpha E_s \quad (12)$$

where, α is the regularizing parameter. Now minimizing the energy terms give the displacement field obtained from subsequent images. A numerical approach to the problem is described in Brox[9]

2.4) Multiscale approach:

Use of the constant gray level assumption and the expansion of the terms using the Taylor expansion is given in Eq. 5:

$$I_x u + I_y v + I_t = 0 \quad (5)$$

The small displacements assumption in expanding the gray value constancy results in the above equation. Thus use of optical flow algorithms for the flow analysis in a sequence of images has a limitation on the maximum measurable displacement. To cope with this limitation Anandan [24] suggested a multiscale/hierarchical approach to the problem. The method suggests to start with the lowest resolution images and then go to increasingly resolved images. At the lowest resolution displacements are calculated and these are used at the next higher resolution to warp the first image and then calculate the displacements again. This is done till the highest resolved images and the displacements from all states are added to get the final displacement values. For many down-sampling approaches it is customary to use a factor 0.5, but efficient results are found with higher factors.

Part 2: Communication model

2.5) Co-operative robotics:

Cooperative robotics is that field of mobile robotics that deals with the study of multiple autonomous robots “working together” to perform some task that is either too difficult or impossible for a single robot to perform while acting alone. It combines the disciplines of computer science, electrical engineering, and artificial intelligence. These

tasks can be anything from a simple task to box-pushing [27], exploration [28], area mapping [29], fire-fighting [30]. Fully autonomous and cooperative robots remain an unrealized goal for researchers worldwide [3].

Liu and Wu [31] put forward that using multiple cooperating robots has several advantages over single-robot systems. These advantages include greater efficiency, a wider defined task domain, inherent parallelism, and distributed sensing. Also, investing on a number of cheap “off-the-shelf” robots can be less expensive than one large, custom-made robot [32]. Since these robots would be used in several search tasks, such a tactic would be beneficial when a failed robot can never be retrieved.

The major development in the field of cooperative robotics began in 1989. Asama et. al. [33] present ACTor-based Robots and Equipments Synthetic System (ACTRESS) which is a distributed multi-robot system designed for maintenance tasks in nuclear power plants. The autonomous components of this system are termed “robotors” and can be mobile robots or any component that has at least two basic functions: 1) the ability to sense surroundings, make decisions, and act on these decisions and 2) the ability to communicate to other robotors for purposes of cooperation and interference avoidance. ACTRESS is shown in simulation with the cooperative task of two mobile robots pushing boxes to the sides of a room.

Cooperative robotics can be divided into two broad areas of research, Swarm robotics and Intentional cooperation [34]. The swarm-type approach deals with a large number of lower-level robots that are performing tasks independently and are normally unaware of each other’s actions. Tasks proposed for swarm robots usually have a parallel nature such as collecting rock samples on Mars and sorting mail. Cooperation in this type

of robotics occurs due to the statistical result of a large number of repeated actions. The goal of swarm robotics is to design the control laws of each robot such that, many simple interactions with the environment produces a globally desirable behavior. On the other hand, intentional co-operation deals with a limited number of higher-level robots that are aware of the actions of the other robots in the same network. Each robot also uses the information received from the neighboring robots to make its own best decision for its next move [35]. Task examples for this type of cooperative robotics are moving furniture, building space stations. In this approach, cooperation can be imbued in the robots both locally and globally, unlike just globally for swarm robotics.

2.6) Communication in Cooperative robotics:

Communication among the robots is essential for cooperative behavior among the robots. It is that area of research in robotics which is developing extensively. Performing experiments with inter-robot communication is a common topic of research in cooperative robotics [27, 28].

Fukuda and Nakagawa [36] introduced the idea of a dynamically reconfigurable robotic system (DRRS) which allows a robot to autonomously reconfigure its parts based on the goals of a specific task while communicating with its neighbors. DRRS consists of robotic “cells” which communicate with each other and can approach, detach, and combine themselves in different ways depending on task definition and allowable workspace.

Communication in cooperative robotics can basically be divided into two types, Explicit communication and implicit communication [28]. Explicit communication is when the robot is trying to communicate to an external agent, or the environment.

Example of such a communication would be radio transmissions. Implicit communication is when the robots are communicating mutually to understand the progress of the task or to understand the factors affecting the task. Example of implicit communication would be when two robots are performing a task together, and one robot realizes the task is done and stops, because the last bit of the task was performed by the other robot. This mutual understanding between the robots is implicit communication.

Dadios and Maravillas [35] use implicit communication in a team of two soccer playing robots. Fuzzy logic and an overhead camera are used for navigation. Fuzzy logic helps in the decision making for the robots. The overhead camera is used by both robots for implicit communication, but the robots are not allowed to explicitly communicate with each other. In simulation the robots are able to effectively pass and shoot the ball into a goal in the presence of opposing team members represented by stationary obstacles.

Asama et. al. [33] use explicit communication by putting laptops, equipped with wireless modems, on each mobile robot. This allows large amount of information like global maps, to be processed. Simsarian and Matarić [27] use explicit communication by equipping two box-pushing robots with radio communication.

After deciding what kind of communication to use, the next question would be, “what is the optimum amount of communication for a particular task?” Arkin et. al. [37], Matarić et. al. [27] in their experiments address this statement. They conduct a set of experiments for no communication and simple communication and compare it to see the behavioral state of robotic system. Similar experiments are performed in this thesis, results of which are shown in Chapter 6.

2.7) Search and Rescue operations in Cooperative robotics

Search and rescue robotics has been a very big part of the applications of cooperative robotics. As already discussed in Chapter 1, “The first real research on search and rescue robot began in the aftermath of the Oklahoma City bombing in 1995” [2]. Robots made their first appearance in actual human search and rescue operation on September 12, 2001, “Mankind’s largest push to develop robots arguably began in the days immediately following the 9/11 terrorist attacks, when shoebox-size “PackBots” made by iRobot Corporation (best known for its semi-autonomous Roomba vacuum cleaner) were used to ensure the stability of rubble piles before first responders started their search for survivors at New York’s World Trade Center site” [1]. These robots were of different sizes and performed various different tasks. Sources say that these robots actually discovered more than 2 percent of the victims, on that day [3]. Even after this, there have been a lot of attempts made in the area of search and rescue. Jennings et. al. [38] present two large robots searching for and “rescuing” warehouse-like boxes. Here, the robots are randomly looking for the target in an obstacle free environment. Vainio et. al. [39] present a control architecture for a group of underwater search-and-destroy robots. In all these experiments, we see that time is a very important factor in search and rescue operations. The results of this thesis shows an analysis of how time taken to complete the search task is influenced by change in other simulation parameters in the experiments.

2.8) Mobility models

In this thesis we discuss how communication can affect the performance of a single mobility model. Mobility models are typically used to simulate and analyze routing

protocols and network analysis in VANETs and MANETs and to represent the movement of mobile users, and how their location, velocity and acceleration change over time. Such models are frequently used for simulation purposes when new communication or navigation techniques are investigated. A **mobile ad-hoc network (MANET)** is a self-configuring infrastructure-less network of mobile devices connected by wireless. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. Such networks may operate by themselves or may be connected to the larger Internet.

A **Vehicular Ad-Hoc Network (VANET)** is a technology that uses moving cars (robots in our case) as nodes in a network to create a mobile network. VANET turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 metres of each other to connect and, in turn, create a network with a wide range. As cars fall out of the signal range and drop out of the network, other cars can join in, connecting vehicles to one another so that a mobile Internet is created.

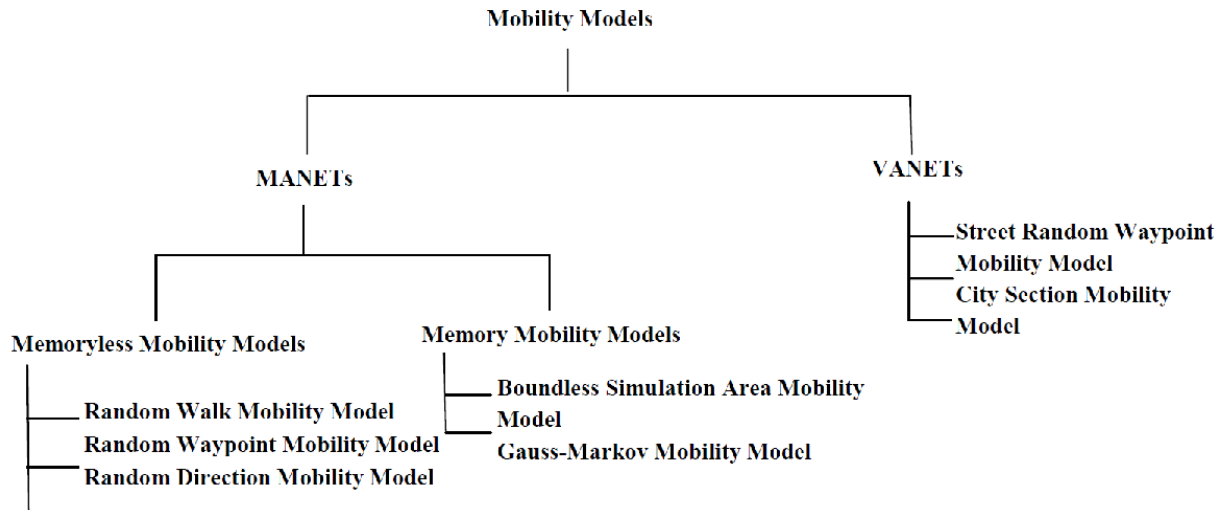


Figure 2.2) Classification of mobility models

The Manhattan mobility model is a common type of VANET. These mobility models when incorporated in robotics can be used for various cooperative tasks like search and rescue. The mobility model that is most important in this thesis is the Manhattan mobility model. In this model, the direction of nodes is decided probabilistically at every street intersection. In the Manhattan model, the mobile node is allowed to move along the horizontal or vertical streets on the urban map. More about this model is discussed in Chapter 5.

2.9) Simulation Environments

Simulations are the most important tools in any wireless networks research or robotics. Simulations are highly preferred in this type of study as the models are easier to replicate and analyze with simulations than with actual experiments. Current simulators are lagging in modeling essential characteristics of the real world [40], [41]. When deploying mobile nodes, real-world experiments are rare.

For most simulations testbeds are used for wireless research. Testbeds need to be able to provide means for programming the devices, they need to be able to house actual fully operating robots, they should be able to allow accurate motion, should be available for access online to remote users and also help in easy collection of experimental data. The simulation tools used for most simulations are MATLAB or the NS2-network simulator 2. NS2 is a very prominent simulation tool which also allows for the design of testbeds and helps simulate a very high number of nodes. It provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks [42].

Another simulation testbed available is MiNT[43] which is a miniature 802.11 testbed. It focuses on reducing the area required for a multihop 802.11 testbed. It achieves mobility through the use of antennas mounted on Lego Mindstorm robots, tethered to a PC running the applications where the robot mobility is limited. The ORBIT [44] testbed provides a remotely accessible 64-node indoor grid with 1m spacing, that can emulate mobility. User code is run on PCs, which are dynamically bound to radios. The Mirage testbed [45] is different from all the other testbeds. It uses resource allocation in a sensor net testbed as a target to explore new models of resource allocation.

CHAPTER 3

LABORATORY SETUP AND ROBOT ARCHITECTURE

The work presented in this thesis is one of the many collaborative robotics projects done in the CRR (Cooperative Robotics Research) lab at Auburn University. This chapter provides a detailed explanation of both, the hardware and software parts of the robots and the experimental setup used to run the simulations. It gives a brief overview of the robot architecture and explains the functionality of the robotic nodes. The robots described in this chapter have already been used to demonstrate the use of ad-hoc routing networks and are a part of few other projects that can be found on [13]

3.1) Laboratory Setup:

The laboratory setup includes a team of three identical robots working in a collaborative manner to explore the search area which is the third floor of the Broun Hall building at Auburn University. The functionality includes on-board processing, mobility and wireless capability. These robots are all operating on ROS [14] and are moving around this particular search area using a map to find the target while avoiding any static or dynamic obstacles that may be encountered along the run. The robots have the capability of building the map as they progress in the search environment, or an already updated map can be inbuilt in the system.

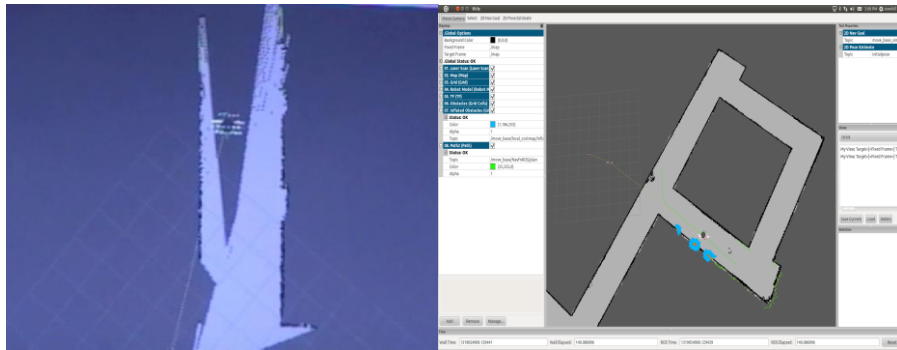


Figure 3.1: The map (a) being generated by the robot, the map (b) inbuilt in the system.

The stationary target is set through the software for every run, and the robots all start out from the same point in search of the target. These robots use 12 V sealed lead acid batteries for operation and are capable of motion at the maximum rate of 0.5 m/sec.



Figure 3.2: Team of robots exploring 3rd floor of Broun Hall, Auburn University

3.2) Robot Architecture:

The architecture of the robot is constituted of four main parts: chassis, drive and control system, vision system and power system.

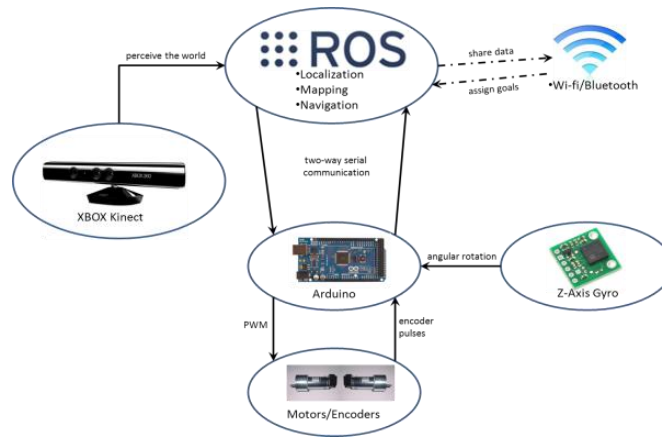


Figure 3.3: The robot control architecture

3.2.1) Chassis:

The chosen build for the bots is from Zagros Microcontrollers. It consists of 3 plastic platforms each having a diameter of 16 inches to house the various components that go on the bot. The first level houses the laptop which is basically the brain of the robot. The second platform is the vision sensor platform, which houses a Kinect™ sensor. The third level is the base platform that provides housing for the motors, gyro and all the other components that help in processing the motion of the robot.

3.2.2) Drive and control system:



Figure 3.4: The drive and control system

The drive system consists first of the wheels. The robot has 4 wheels, 2 castor wheels for stability and 2 main differential drive wheels for movement. There are 2

differential drive motors attached to each wheel and 2 encoders for the motors to determining the speed of the robot by counting the number of ticks per second. Next is the Arduino Mega microcontroller, it handles all of the low level drive train operations and is responsible for communicating with the laptop over the USB port. It relays the encoder and gyro raw signals to the laptop. The laptop is then continuously running a dual PID loop (one for each motor) to drive the motors at a specified linear and angular velocity. The control loop calculates PWM values that are serially sent back to the Arduino to then control the speed of the motors. It also consists of an ADXRS 300 degrees/sec Gyro sensor which measures the angular velocity around an axis. There is also a L298 N based motor driver board which is responsible for driving the motors. It has a 12 V direct connection to it.

3.2.3) Vision system:



Figure 3.5: Vision system, Kinect™ sensor

The vision system used in these bots, is the XBOX 360 Kinect™. It has an RGB camera and an infrared camera. The infrared projector projects a grid-like infrared pattern. The infrared camera detects the distortion in the pattern and generates a depth image to see how far the objects are from the Kinect™. The laptop has the ability to make a point cloud from the depth image generated using the software ROS. A horizontal slice

of the point cloud is then taken, which would almost resemble a laser scan to help with the mapping and navigation for the robot. The laptop also has a node that uses the wheel encoder counts and the rate of the gyro to keep track of the odometry measurement. Using a known map of the 3rd floor of Broun Hall, the point cloud of what the XBOX Kinect™ sees, and the odometry measurement, the robot can pretty accurately estimate its location on the 3rd floor of Broun Hall. Once the robot localizes itself, it can autonomously drive to another point in the hallway by continuing to combine the odometry and point cloud inputs.

The ROS (robot operating system) [14] allows the robot software to be designed in a modular fashion. The computation is divided into a set of independently running nodes, where each node is like a separate application on the laptop. ROS also allows for the reuse of code in the research community by providing a standard for how robots are to be defined such as; is left the positive or negative turn direction. In this way, newly developed navigation algorithms also can be easily ported to a variety of robots without having to make the code specific to each robot.

3.2.4) Power System:

Each robot derives its main power from a 7 AH, 12 V sealed lead-acid battery. The raw battery voltage is distributed to the motors and electronics with switching, fusing, and regulation as needed. The laptop uses its own internal battery for power.

CHAPTER 4

OPTICAL FLOW

4.1) Objectives:

- 1) Implement a new obstacle avoidance method for mobile robot navigation in a busy environment using optical flow.
- 2) Locate survivors and victims using specialized sensors, provide 2-way audio/visual communication for rescue teams, cooperatively plan search paths.
- 3) Enhance the ability of a team of wirelessly-connected mobile robots to find a target, using vision as a primary sensory modality

The simulation environment is same as stated before, and the robots are assumed to be moving in a staccato fashion, i.e. each robot first takes a picture of the environment, then moves a few steps, examines the environment, and takes a new picture. It then processes both the pictures taken through the Brox algorithm to create a disparity map.

4.2) Disparity Map:

A disparity map shows the apparent pixel difference or motion between a pair of stereo images. When we close our eyes, and open each eye once at a time, we notice that the objects that are close to us will appear to jump a significant distance while objects further away will move very little. That motion is called the disparity. In a pair of images derived from stereo cameras, we can measure the apparent motion in pixels for every point and make an intensity image out of the measurements.

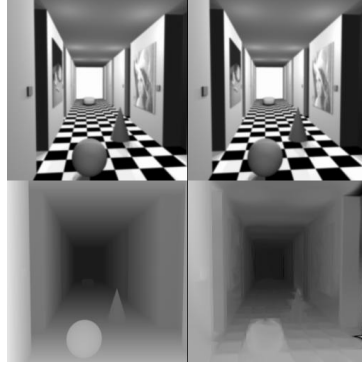


Figure 4.1: Top: Left and Right stereo images Bottom: Disparity map of stereo images

The above picture implies that the objects in the foreground are brighter, denoting greater motion and lesser distance.

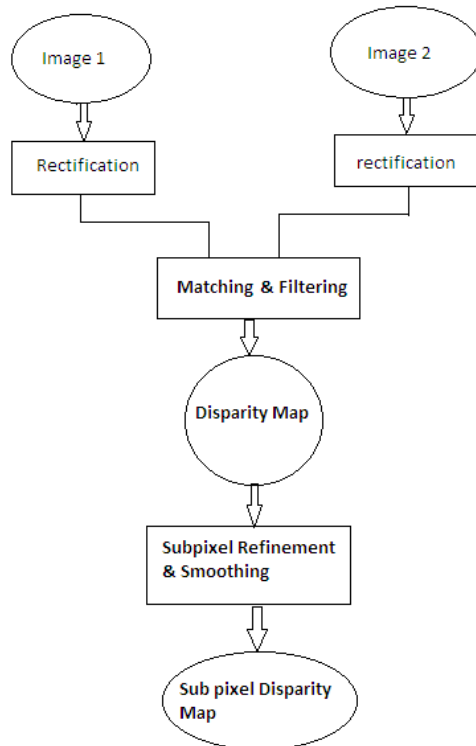


Figure 4.2: The flowchart for creating a disparity map

Image 1 is the left image and Image 2 is the right image.

4.2.1) Image Rectification:

Image rectification is a transformation process used to project two-or-more images onto a common image plane. It corrects image distortion by transforming the image into a standard coordinate system. It is used in stereo vision to simplify the problem of finding matching points between images. In stereo image processing, a problem that must be addressed is the correspondence problem. This is the difficulty in finding the corresponding point related to the first image in the second image in the stereo image pair. This can be solved, if the line of sight of the two images is aligned to be coplanar, so that the search is simplified to one dimension - a horizontal line parallel to the baseline between the cameras. Furthermore, if the location of a point in the left image is known, it can be searched for in the right image by searching left of this location along the line, and vice versa. Image rectification is an equivalent alternative to perfect camera alignment. Image rectification is usually performed regardless of camera precision due to impracticality or impossibility of perfectly aligning cameras or perfectly aligned cameras may become misaligned over time.

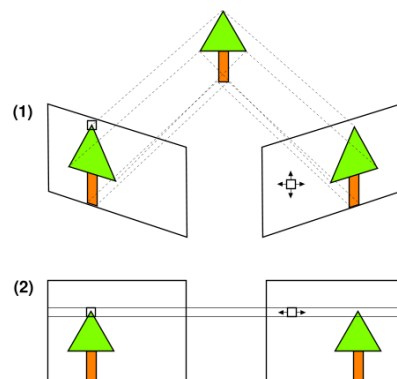


Figure 4.3: Search space before (1) and after (2), the rectification process

Steps involved in the process of image rectification:

- 1) Read Stereo image pair.
- 2) Collect interest points from each image.
- 3) Find putative point correspondences
- 4) Remove outliers using geometric constraint
- 5) Remove outliers using epipolar constraint.
- 6) Rectify images.
- 7) Generalize the rectification process.

4.2.2) Matching and Filtering:

The two rectified images are then matched and filtered to find the area of movement. Matching of images is done by first obtaining the pixel wise differences in both the images and then finding the point correspondences between both the images along with the corner detection. Then, the correspondences between the points are selected and the matched points are retrieved from both the images. Image filtering is useful for many applications, including smoothing, sharpening, removing noise, and edge detection. A filter is defined by a kernel, which is a small array applied to each pixel and its neighbors within an image. Filters are applied in either the spatial domain or the frequency domain. Within the frequency domain, a filter is applied to an image by multiplying the FFT of that image by the FFT of the filter. When the FFT of an image is multiplied by the FFT of a filter to perform convolution, this process is known as windowing. This process described above helps reduce the noise in the image. And finally the rectified images without the noise are ready for the disparity function

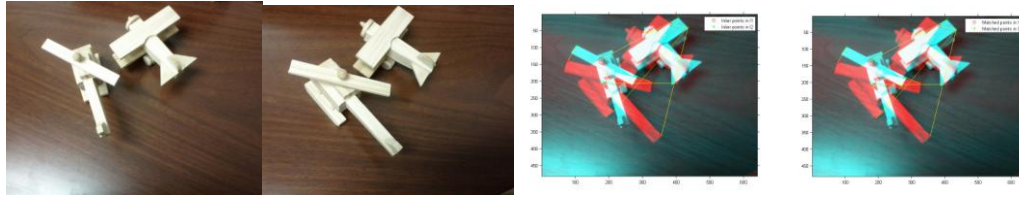


Figure 4.4: Input images, Point correspondences and matching between the images.

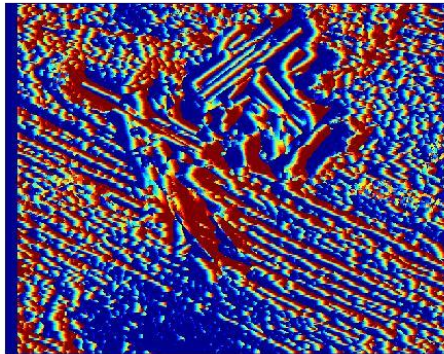


Figure 4.5: Disparity Map, with window size 5. The darker areas show maximum movement

4.3) Simulation flowchart :

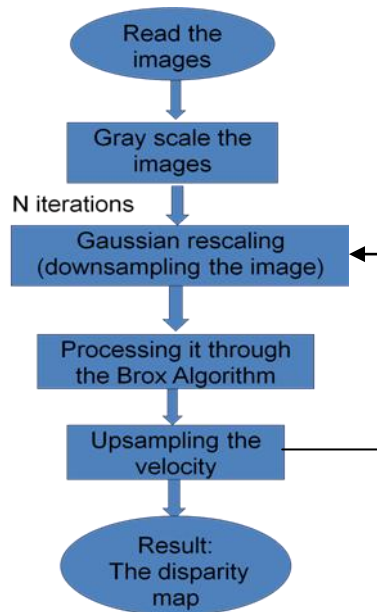


Figure 4.6: Simulation Flowchart for the optical flow process

The above flowchart describes the optical flow process implemented in the thesis. Gray scaling of the input images is done to reduce the pixel values of the images captured. This helps in faster simulation of data, and also helps eliminating the unwanted data in the images. The number of iterations is decided on the size of the images being processed. Gaussian rescaling is done to eliminate all the noise in the images and to eliminate most of the unwanted data before processing it through the Brox algorithm. Up-sampling the velocity facilitates the study of movement in the stereo images captured.

CHAPTER 5

MANHATTAN MOBILITY MODEL AND COMMUNICATION MODEL

Due to extensive research in the field of wireless ad-hoc networks, there is a lot of importance being given to mobility models in wireless networks. Mobility models are of two types: traces and synthetic models [46]. Traces are real life models which have a number of nodes and really long observation periods, and are very tough to be modeled. Synthetic models on the other hand attempt to realistically represent mobile nodes without using traces. In these models changes in the parameters can occur in realistic time frames.

5.1) Manhattan mobility model:

A very important part of this thesis is the Manhattan mobility model. The communication model designed in this thesis is suited for the Manhattan mobility model and analyzes the increase in the efficiency of the performance of this particular model after communications are incorporated in the model. This model works on the principle that follows: The direction of nodes is decided probabilistically at every street intersection. In the Manhattan model, the mobile node is allowed to move along the horizontal or vertical streets on the urban map.

Junction is the point of intersection of two perpendicular grid lines. In this model, each node, in our case robot, which is moving in a particular direction, chooses a random direction along the grid when a junction is encountered and keeps moving in the chosen

direction until it reaches the next junction. Figure 5.1 shows an example topography of the movement of nodes for Manhattan Mobility Model with nine nodes. The map is used to define the path of movement for the robots. Figure 5.2 shows us the movement pattern of the robots in this model, where, each line represents a single-lane road. Robot movement occurs along the direction shown by the arrows.

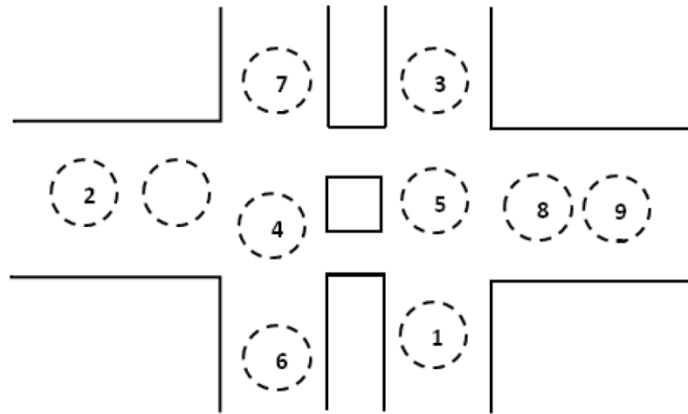


Figure 5.1: Depiction of the movement of the nodes using Manhattan mobility model

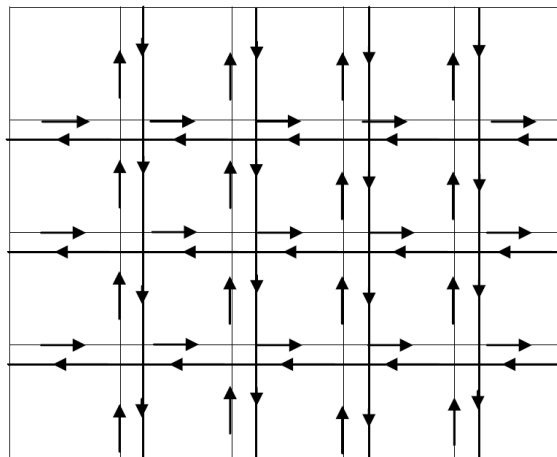


Figure 5.2: Travelling pattern of a robot using Manhattan mobility model

From Figure 5.2, it can be seen that the robots can move along the vertical and horizontal streets, and when the junction is encountered, i.e., the intersection of a horizontal and vertical street, the robot can choose to go either right, left, up or down

with a certain probability. The probability of going straight is 0.5 and the probability of going either left or right is 0.25. The robots are initially assumed to be placed randomly at the junctions and the movement of a single robot, is decided one street at a time. A few changes have been made to the conventional model to well suit the search and rescue operation experiments. The robots all start at the same point from the middle of the search grid, and obstacles, sensor range and communication between the robots have been added to the model.

5.2) Simulation model fundamentals:

5.2.1) Target Detection: The robots are all made to start from the mid-point of the search grid, and are exploring the search area using the basic Manhattan mobility model.

A random target is generated at every run. A simple target detection technique is employed where the robot matches the coordinates of the next step to the target coordinates and the target is detected. When expanded sensor range is employed, the robots match the coordinate of the target in the window of the sensor range and if the coordinates happen to be in that window range, the target is detected.

5.2.2) Obstacle avoidance: A basic obstacle avoidance algorithm is used. If a node encounters an obstacle, along its path to the next coordinate, the robot retreats back to its previous position and a new coordinate and speed is assigned to it. The same procedure is employed even with the use of expanded sensor range. The robots also communicate the positions of the obstacles encountered along their path to the neighboring robots when the simulation is run with the communication model.

5.2.3) Communication: The robots are allowed to communicate with each other.

Communication involves a broadcast of the explored area of a single robot i.e., each grid

point that has been visited by each robot is broadcast to every other robot in the network. This greatly reduces redundancy of the search area. This information can be used to build a global map of the environment from each robot's local map.

5.2.4) Collision avoidance: For the practical application of the simulations designed, the robots need to be able to avoid any form of collision with either the obstacles or with neighboring robots. The nodes need to be aware of the positions of the neighboring robots and avoid collisions while staying within the communication range of the repelling robot. This also helps distinguish between obstacles and other robots. This is done with the help of attractive and repulsive forces between the nodes. Each robot is simulated to be circular in shape and has a circular repulsion zone around it. For a robot not in the repulsion zone of other robots, the next coordinate and speed is randomly generated coordinate and assigned. For those robots that are lying within each others' repulsion zones, the speed and direction are calculated as follows:

A circular region centered on a robot with a radius of 'r' is considered as the repulsion region of that robot. If a robot is within one or more repulsion regions of other robots, then it will calculate its movement vector based on the repulsion force(s), instead of based on the random movement vector.

Let $\vec{f}(i, j, t)$ be the repulsive force induced from i to j iteration t and $\vec{m}_j(t-1)$ be the movement of robot j at time $(t-1)$, regardless of whether it is due to random movement or repulsion forces. The movement vector of j at iteration t is:

$$\vec{m}_j(t) = \vec{m}_j(t-1) + \sum_{i \in I_j} \vec{f}(i, j, t)$$

5.2.5) Boundary conditions: the behavior of the robots at the boundaries is also a very important part of the thesis. The boundaries are blocked in the initial part of the

simulation code for the robot to avoid it. A robot which can go outside the set boundary is probabilistically restricted from making a movement during that iteration thus keeping it at the same position.

5.3) Simulation model developed

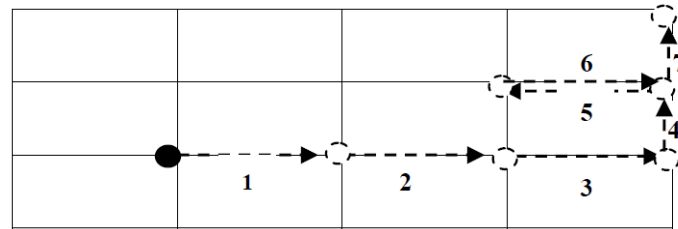


Figure 5.3: Closer look at the travelling pattern of a robot using Manhattan model

From figure 5.3, it is seen that in this model the robot chooses its next direction probabilistically when a junction is encountered. The nodes are initially assumed to be randomly placed at street intersections. The movement of a single robot is decided one street at a time. To start with, each node has an equal probability of choosing any of the streets leading from its initial location. After a node begins to move in the chosen direction and reaches the next street intersection, the subsequent street in which the node will move is chosen probabilistically. The robot is allowed to move in the same direction or change directions at the junctions. If the robot can continue to move in the same direction, the probability is 0.5, the probability is 0.25 when the robot has the chance of turning to the east/north, west/south, depending on the direction of the previous movement. If the robot has only one option to move (this occurs when the robot reaches any of the four corners of the network), then the node has no other choice except to explore that option.

The simulation flowchart for the model designed in this thesis is shown in figure

5.4

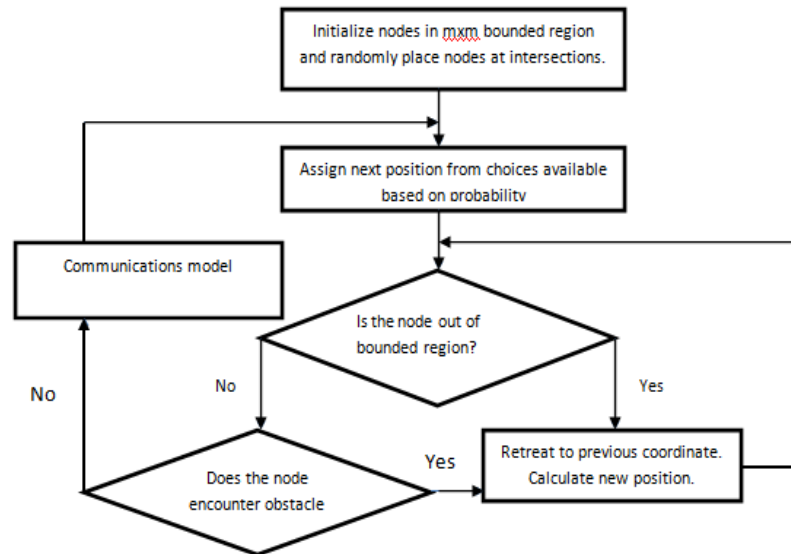


Figure 5.4: Simulation flowchart for the Manhattan mobility model used in the thesis

5.4) Communication model developed:

For the basic communication model developed, the search grid is developed, and the number of robots and obstacles are initialized. A random target is placed in the grid, and the robots all start from a common point to explore the search area. The individual robots travelling in a particular direction look one step ahead in the direction of travel and if that coordinate is not the target, not an explored point, and not an obstacle, it continues moving in that particular direction, and sets the flag to 0.5, which specifies an explored coordinate according to our simulation. Otherwise, when it has to change the direction, it looks for free neighbors in another direction, if there are none, it goes to a previously explored neighbor. It checks all neighbors in clockwise succession for first free neighbor,

if none is free, takes a random previously explored coordinate. There must be at least one of these, or else robot is trapped, but it can't be since at worst it could back track its path. This way we avoid the robot getting trapped within its own explored neighbors. Different flag values are assigned for target detection, explored cell, communication and obstacle. The flowchart for the simulation model is shown in figure 5.5:

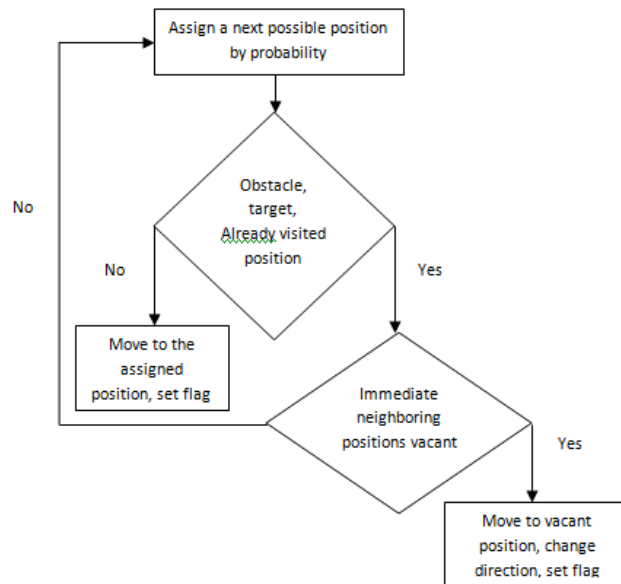


Figure 5.5: Flowchart of the communication model developed.

The figure 5.5 shows the basic flowchart of the communication model developed, but there are many other conditions that are to be considered along the path of travel of the robot. For example, the robot also checks for the boundary condition every iteration and plans its next step accordingly. Also, the checking of the neighboring vacant cells is done in a clockwise pattern, if all the immediate neighboring cells are occupied.

5.4.1) Sensor range:

The range of view of a single robot without the incorporation of expanded sensor range would be the next immediate cell in its direction of movement. When expanded sensor range is applied, the field of view of the robot changes to all the 8 neighboring grid cells of the robot. The range of view is shown in Figure 5.6

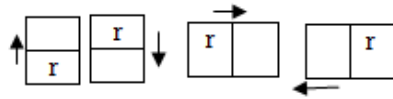


Figure 5.6a): Range of view of the robot with one-step look-ahead

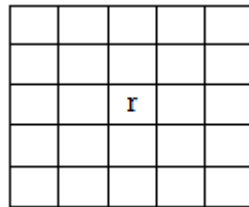


Figure 5.6b) Range of view of the robot with expanded sensor range of 2

From Figure 5.6b) we see that the robot's view with expanded sensor range is the same irrespective of the direction in which the robot is moving. Also with a small sensor range value of 2, we see the difference in the search area covered by each robot, therefore giving a clear picture of how the efficiency of the simulation increases with sensor range and communication.

CHAPTER 6

RESULTS

Part 1: Optical Flow

6.1) Simulation setup:

As discussed in Chapter 5, the output for the optical flow process is a disparity map of stereo images, captured by the robot moving in a staccato manner. We simulated our results using the MATLAB R2011B simulator. The simulation environment is the same as described in Chapter 3 of the thesis. The robot uses an inbuilt map to move around in the environment. Once it is on the experimental field, it captures the images through a Microsoft Kinect installed on it, or through an inbuilt camera of the laptop mounted on the bot. These images are immediately processed through the algorithm, and the disparity map is created, through which the robot can decide its next move.

6.2) Result:

Example input images are shown in Figure 6.1:



Figure 6.1: The input images a) at the starting point, b) after progressing a few steps.

In the above images the distance moved by the robot is clearly noticeable. After processing the images through the optical flow process and the Brox algorithm, the output observed is shown in figure 6.2:

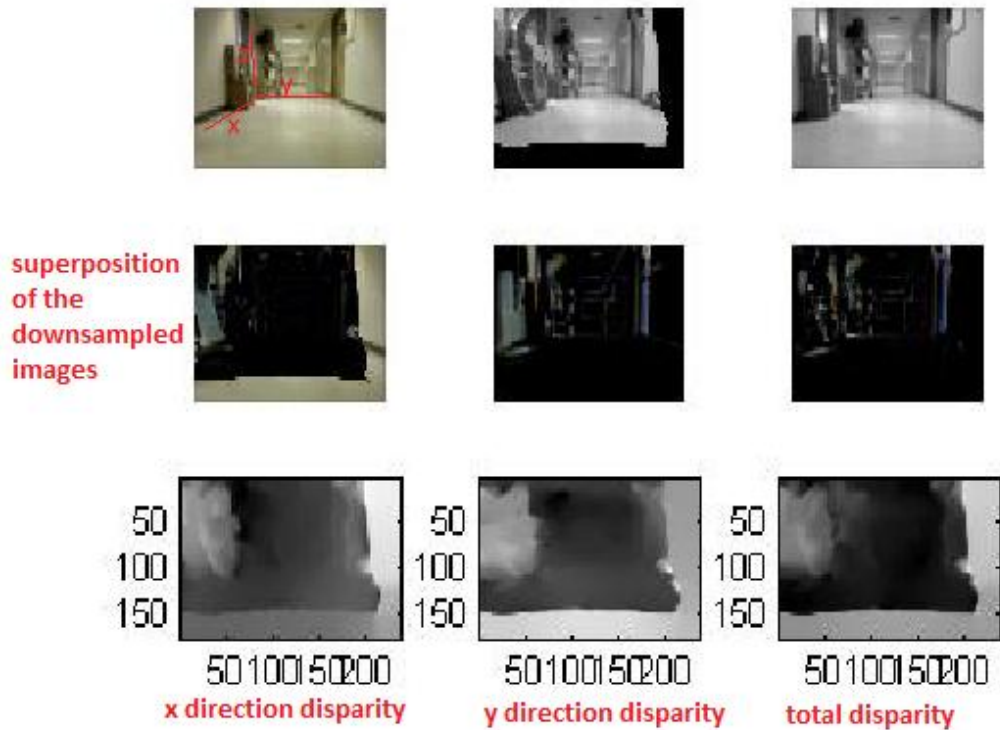


Figure 6.2: The output of optical flow, the disparity map

1) The first image in the first row of the result shows the first input/reference image captured by the robot. The second image in the first row shows the gray scale image of the second input image. The blacked out portion of the image highlights the area not common to the reference image. The third image in the first row is just the gray scale image of the second input image.

2) The second row of the image show the superposition of the downsampled images at various stages of the optical flow process. Downsampling of the velocity in both the images helps in understanding the flow of motion in the environment.

3) Results/disparity maps from the optical flow process are shown in the third row. Disparity measured in the x direction and the y direction are shown in the first two images of the third row. The lighter gray areas are the areas where there is minimum or no motion. Gray levels increase with the amount of motion in the scene, i.e., black means the maximum motion has occurred. The third image shows the total measured disparity. Timing the measured disparity gives the speed at which the robot sees these changes. Having the disparity in these three different domains helps the robot in getting a proper three dimensional picture of the environment and the relative movement occurring in its field of view, thereby enabling it to avoid static and dynamic obstacles.

Part 2: Communication Model for Robot Collaboration

In most of the applications of Collaborative Robotics, communication between the robots is imperative. When the Manhattan mobility model, which is usually used in VANET's, is applied to robotic nodes, slight variation in it is needed to ensure practical application. Since this thesis aims at implementing the particular mobility model in a

search and rescue environment, three very important features are introduced in the simulations: obstacles, inter-robot communication and expanded sensor range. A detailed explanation of the sensor range is given in Chapter 5.

This part of the chapter provides an analysis of how communication affects the performance of the robots using MATLAB simulations. Communication of the already visited coordinates and position of obstacles on the search grid are the most essential data communicated between the robots.

6.3) Simulation Setup

The simulator used to perform the simulations is MATLAB R2011b. The robots used for the practical application of these simulations are same as described in Chapter 3 of the thesis. The grid size for the simulations can be altered according to the requirements. A number of robots ranging from 1-10, are deployed on the search area/experimental grid of size 51 by 51, and the searching algorithm is used to detect the target, while avoiding any obstacles encountered along the search path. The number of obstacles can be varied from 0-600. The simulation is halted when one robot reaches the target or if the number of iterations reaches a maximum count.

6.4) Simulation Parameters:

Parameters	Value
Simulator	MATLAB R2011b
Area/Grid size	51 x 51, 10x10
Number of Robots	1-15
Speed	5cm/sec
Number of obstacles	0-600

Sensor Range	1-10
Number of Runs	100

Table 6.1: Simulation Parameters

The parameters varied for the simulations are:

- 1) Continuous communication vs No communication
- 2) Number of nodes
- 3) Grid Size
- 4) Number of obstacles
- 5) Sensor range

6.5) Results and Analysis:

All the above parameters are varied for different experiments and the graphs are plotted to show the change in the characteristics.

6.5.1) Varying the communication intervals:

All of the simulation results are a comparative study for “with communication” vs “without communications.” The simulation is carried out for 100 runs, and for every 10 runs, the comparative graphs are plotted.

The following parameters were set to get the results.

Parameter	Value
Number of robots	3
Number of obstacles	0
Grid size/area of exploration	10x10

Table 6.2: Simulation parameters set for varying communication intervals.

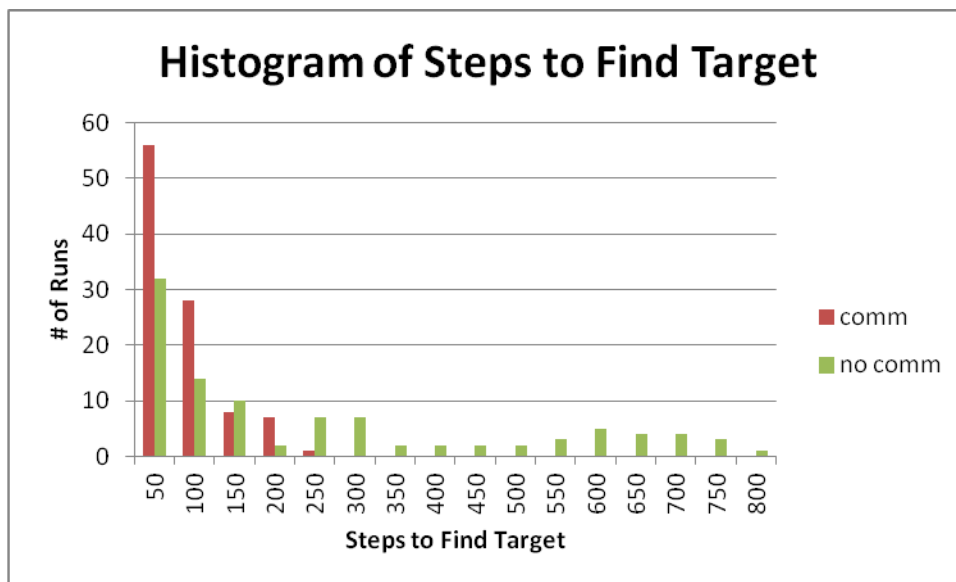


Figure 6.3: Histogram plotted for number of attempts in running the code to the number of steps taken for target detection.

The intervals of communication are varied to see how much communication is needed between the robots to efficiently avoid obstacles and detect the target within a minimum amount of time. From the literature and practical applications so far, ideally, this should help reduce the time to detect the target, as lesser time is spent in exploring the area, and the same search area is repeated a fewer number of times by each robot.

From figure 6.3 it is seen that the average time taken for target detection is reduced by about 80% with continuous communication.

6.5.2) Varying the number of robots:

From the previous section, the effect communication has on the time taken to detect the target, can be noted. The next step would be incorporation of obstacles.

The parameters changed for part a and part b are:

Parameter	Value
Grid size/Area of exploration	51x51
Number of obstacles	20, 40
Number of robots	1-10
Sensor range	5

Table 6.3: Simulation parameters set for varying the number of robots

a) With One-step look-ahead:

From the literature, it is known that having obstacles in the search environment has a significant effect on the number of steps taken to find the target. The time taken to reach the target increases, since in this scenario the robot retreats to a previous position, and recalculates a new position once it encounters an obstacle.

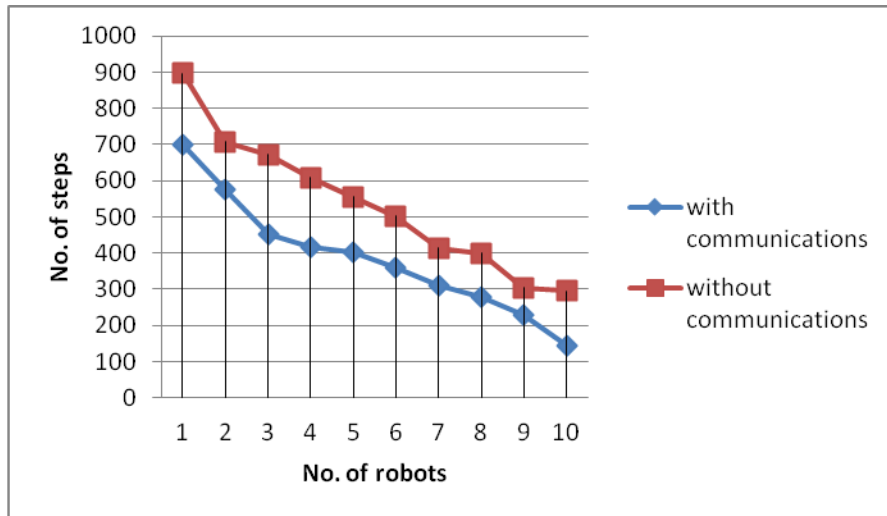


Figure 6.4: Graph showing the change in the number of steps taken for target detection with change in the number of robots with one-step look-ahead.

From Figure 6.4, the number of obstacles is kept constant at 20. It can be seen that as the number of robots is increasing the time taken to detect the target decreases both with communications and without communications. This is because the search area is now being explored by many robots, and therefore the area of search for each individual robot decreases along the way. It is also noted that the number of steps taken to locate the target is reduced by a narrow margin when the simulations are run with continuous communication among the robots. This is because, with continuous communication, the robots are aware of the position of obstacles and also the position of the other robots in the network, therefore avoiding repetition of searching the same grid cells.

Since there is not much difference in the time taken to detect the target, between continuous communication and no communication, another very important factor, 'sensor range' is introduced in our simulations.

b) Expanded Sensor range:

Vision sensors in robots have always been used for collaborative robotics applications. In the simulations that follow, sensor range is shown to have a significant effect on the performance of the system. The robots are all assumed to have a 360 degree vision, i.e., a sensor range of 5, means the robot can see 5 grid cells ahead from all 4 sides and corners of itself.

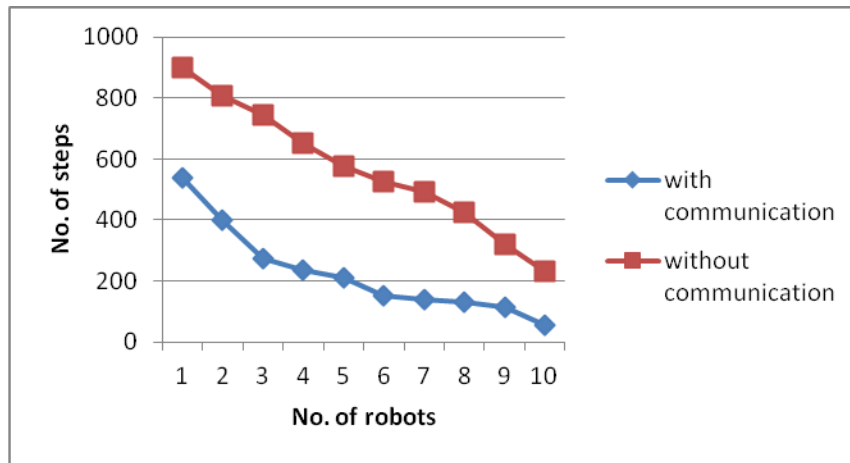


Figure 6.5: Graph showing the change in the number of steps taken for target detection with change in the number of robots, with sensor range 5.

The number of obstacles while performing this simulation is set to 40. From figure 6.5, the significant reduction in the time taken for target detection can be noticed, when communication between robots is employed. When sensor range is increased in this simulation, a larger number of grid cells is being searched by an individual robot at a particular time. Therefore, this ensures more information being shared over the robotic network for each step taken by individual robots. With a 360 degree vision in the robot, the robot can locate the position of obstacles and neighboring robots, and also communicate this information over the network all at once. All this information when

communicated in the network ensures the robot can understand the search area better and faster therefore, enabling the robots to reach the target faster.

6.5.3) Varying the sensor range:

From the above section, it can be deduced that when the simulations are run with the increased sensor range, there is a significant effect on the time taken to detect the target.

The parameters changed for this simulation are:

Parameter	Value
Area/grid size	51x51
Number of robots	5
Number of obstacles	20

Table 6.4: Simulation parameters for change in sensor range

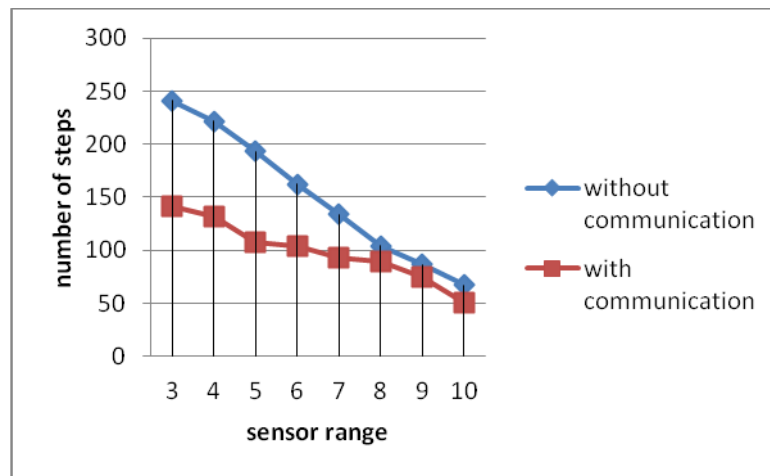


Figure 6.6) Graph plotted for number of steps taken to reach target with change in sensor range.

From figure 6.6, it can be noted that the number of steps taken to detect the target is reduced when the sensor range of the robot is increased by a smaller value. This is because more amount of information about the search area is being shared by the robotic network. But, it is also noticeable that, once the value of the sensor range increases to a higher value, there is not much difference in the number of steps taken by the robot with communication and without communication-this can be explained by understanding that as the ratio of the grid size or the area of exploration to the ratio of the sensor range of each individual robot decreases each of the robots understands the search environment by themselves, irrespective of the communications between them.

A very important conclusion that can be made from this simulation experiment is that the performance of the robots when communicating among them depends significantly on the ratio of sensor range to exploration area.

6.5.4) Varying the number of obstacles:

From section 6.5.2, it is presumed that the incorporation of obstacles has a weighty effect on the number of steps taken to complete the search task. From figure 6.4 and 6.5, it can be seen; that as the number of robots increases, the number of steps to find the target both with no communication and with continuous communication between nodes significantly decreases. This can be credited to the fact that every time an obstacle is encountered, the robot moves back to its original position and recalculates a new coordinate, as already discussed in the previous section.

The parameters set for this simulation are:

Parameter	Value
Grid size/area of exploration	51x51
Number of robots	5,10

Table 6.5: Simulation parameters set for varying the number of obstacles

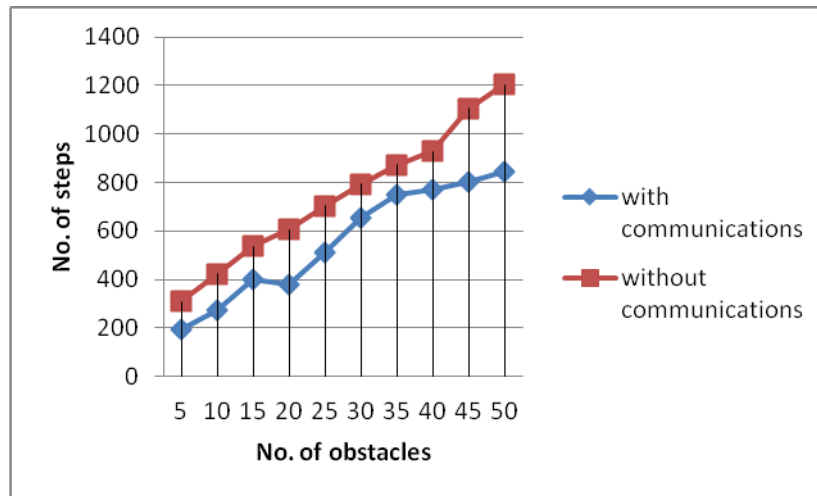


Figure 6.7) Graph plotted for number of steps taken by 5 robots to reach target with varying the number of obstacles.

For this simulation, the number of robots was kept constant at 5, and the number of obstacles is increased from 5 to 50 on steps of 5. In this section we try to look at how the communication model performs when the obstacles are increased. As the number of obstacles increase, when no communication is employed between the robots, the steps to find the target increase linearly. This is due to the fact that the other robots are also seen as obstacles and thus there is a lot of redundancy in search area. With communication between nodes, the time to finish the search task increases initially and then there is a sudden drop at 20 obstacles, and again a linear increase when more than 20 obstacles are

introduced. This is endorsed to the fact that more obstacles mean more reiterations and thus more steps taken before the target is found.

To examine the sudden dip in the graph at 20 obstacles, another set of simulations were run with the same ratio of obstacles to robots, and similar results were obtained.

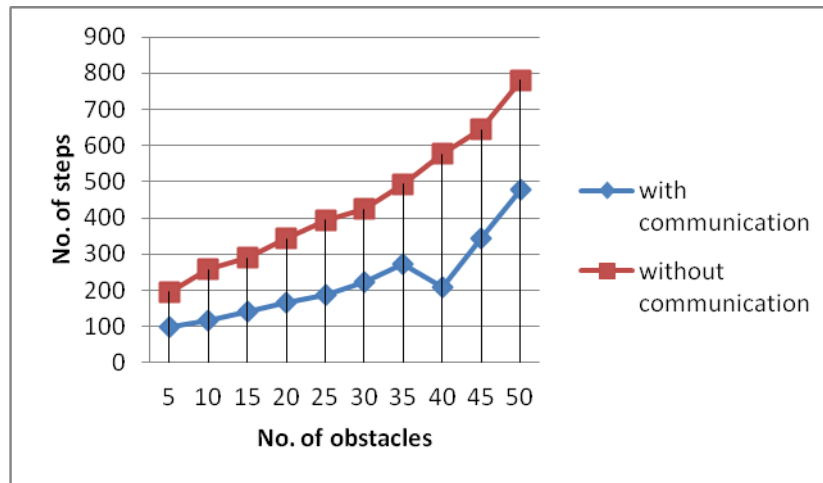


Figure 6.8: Graph plotted for number of steps taken by 10 robots to reach target by varying the number of obstacles.

Owing to the sequential search nature of the Manhattan mobility model, the sudden dip in the graph can be attributed to the fact that each search grid has its optimum ratio of number of robots to the number of obstacles, where it is ideal for the particular communications model implemented in the Manhattan Mobility model to perform its best.

For the above simulation, the number of robots is kept constant at 10, therefore we see a dip at 40 obstacles. Thus we see, From Figure 6.7 and figure 6.8, that the optimum ratio of number of robots to the number of obstacles is 0.25 for the grid size of 51x51.

6.5.5) Varying the number of robots and the number of obstacles to maximum values

From the previously discussed sections, the effects that varying the number of robots, number of obstacles, sensor range has on the performance of the communication model are understood clearly. But, the maximum number of robots, maximum number of obstacles, maximum sensor range that a particular grid size can handle would be a critical finding in this thesis.

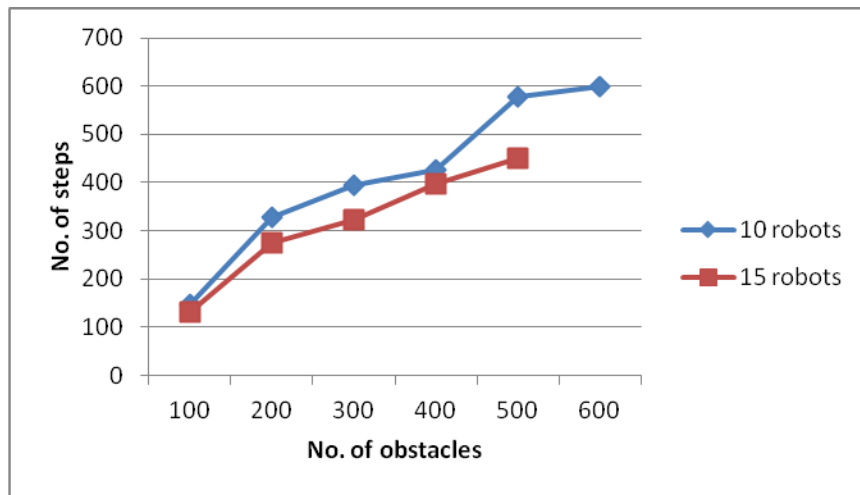


Figure 6.9: Graph plotted for number of steps to number of obstacles for extreme values of number of robots, obstacles

The above simulation was run in a grid size of 51x51, and ‘with communications’ between the robots. For a grid size of 51x51, from the above simulation it can be seen that the maximum number of obstacles the grid can handle for the performance of the communication model is 600. Once the number of obstacles is increased for more than 600, it is seen that for most runs, the robots are trapped by the obstacles, Therefore, making it impossible for the robots to find the target. The simulation halts when such a condition is encountered.

The case is similar for the number of robots too. Once the number of robots is increased to more than 15, most of the robots do not get to move after the initial step, as there are already other robots in the calculated positions, or they are surrounded by obstacles which obstruct their search path. From figure 6.6, it is already concluded that the maximum sensor range this particular communication model can handle would be 10. Once the sensor range is increased to a higher value, there wouldn't be any change in the number of steps taken regardless of the amount of inter-robot communication.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

This thesis demonstrates the working of an optical flow algorithm as well as proposes the opportunity of applying communications to the conventional Manhattan mobility model for cooperative search and rescue using robots. The experiments involve a group of collaborative robots searching for a target. Various parameters are varied: the number of robots, inter-robot communication interval, grid size, the number of obstacles and the sensor range of the robots.

7.1 Conclusions:

7.1.1) Optical flow:

The optical flow algorithm implemented shows us how the robot can perform search and rescue operations in a busy environment with mobile targets and obstacles. From the results it can be deduced that the output of the optical flow process is a disparity map that helps analyze the absolute change in the environment by giving the ‘move’ in the environment along the initial velocity, final velocity and the time axis. This gives the robot a three dimensional view of the scene, which it can use to decide its next move efficiently.

7.1.2) Communication model:

The communication model implemented on the famous Manhattan mobility model is simulated with and without communication and obstacle avoidance.

Simulation results presented in this work suggest that with a simple obstacle avoidance algorithm and inter-robot communication, the robots demonstrate major improvement in completing the task of locating the target compared with no inter-robot communication.

Increasing the number of robots displayed obvious results. There is an inverse relationship between the number of robots and the number of steps for completion of the task. Results indicate that inter-robot communication considerably reduces the time taken to complete a cooperative search task in the same model. Also, in the absence of communication, the number of steps taken to reach the target is considerably reduced when sensor range is increased. And, with communications, increasing the sensor range further reduces the number of steps taken to reach the target. Also since power for the robots is a major restriction for the robots, the results obtained from the simulations serve to be of great use.

In the communication experiments, the communication interval was varied from continuous to none. In all cases, the sets of experiments using communication, and therefore cooperation, outperformed the no communication runs in terms of time for task completion.

The results also show that, as obstacles are introduced in the search environment the time taken for completing the search task increases, as compared to the simulation result of no obstacles. Also, as the number of obstacles increase, the time taken for target detection also increases, but there is an optimum ratio of number of robots to the number of obstacles for which the communication model performs its best. With the help of the simulations this ratio for a 51x51 grid size was found to be 0.25.

From the simulation results it is also noted that the maximum number of obstacles a 51x51 sized area of exploration can handle, is 600, after which the robots get stuck among obstacles and the simulation is halted. The maximum sensor range the grid can handle was known to be 10, after which it wouldn't make a difference in the performance irrespective of the communication between the robots. The maximum number of robots the search area can handle was found to be 15, after which most of the robots are not able to move after their initial position, and the simulation is halted.

The next section presents some suggestions for future work.

7.2 Suggestions for Future Work:

- 1) Implement the mobility model, the communication model, optical flow algorithm on CRR laboratory's mobile robots.
- 2) Consider the use of cognitive radios for inter-robot communication.
- 3) Implement the communication model for mobile targets and mobile obstacles.
- 4) Implement the optical flow process using input images of a higher pixel range as compared to the VGA input images used.

REFERENCES

- [1] Douglas Main, “Robots to the rescue[sponsored post]”. Internet.
<http://www.popsci.com/trp-sponsored-article-slideshow>, April 8th 2013
- [2] Albert Ko, Henry Y. K. Lau, “Robot Assisted Emergency Search and Rescue System With a Wireless Sensor Network”, *International Journal of Advanced Science and Technology*, Vol. 3, February, 2009
- [3] A. Jacoff, et. al, “Test arenas and performance metrics for urban search and rescue robots,” in *Proceedings of the IEEE International Conference on Intelligent Robots and systems*, vol. 4, 2003, pp. 3396-3403.
- [4] Y.U. Cao, AS. Fukunaga, and A.B. Khang.” Cooperative mobile robotics: Antecedents and directions.” *Autonomous Robots*, 4, 1997
- [5] G. Dudek, M.. Jenkin, E.E. Milios, and D. Wilkes. “A taxonomy for multi-agent robotics.” *Autonomous Robots*, 3(4), 1996.
- [6] D. Guzzoni, A. Cheyer, L. Julia, and K. Konolige. “Many robots make short work.” *AIMagazine*, 18(1):55-64, 1997.
- [7] S. Belongie, J. Malik, J. Puzicha, “Shape matching and object recognition using shape contexts.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (4) (2002) 509-522.
- [8] K. Sabe, M. Fukuchi, J.S. Gutmann, T. Ohashi, K. Kawamoto, T. Yoshigahara, “Obstacle avoidance and path planning for humanoid robots using stereo vision.” *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, vol. 1, 2004
- [9] D. Murray, J.J. Little, “Using real-time stereo vision for mobile robot navigation.” *Autonomous Robots* 8 (2) (2000) 161-171.
- [10] I. Ulrich, I.R. Nourbakhsh, “Appearance-Based Place Recognition for Topological Localization”, in: *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2000, pp. 1023-1029.
- [11] A. Ray, “Cooperative Robotics using Wireless Communication,” Master’s thesis, Auburn University, 2000

- [12] Horn, B. K. P., Schunck, B. G., “Determining Optical Flow”, Artificial Intelligence, Vol. 16, pp 185-203, 1981.
- [13] Cooperative Robotics Research Team’s Home Page: <http://crr.eng.auburn.edu>
- [14] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Ng, “ROS: an open-source Robot Operating System”, Internet.
<http://pub1.willowgarage.com/~konolige/cs225B/docs/quigley-icra2009-ros.pdf>
- [15] Simoncelli, E. P., “Course-to-fine Estimation of Visual Motion”, IEEE Eighth Workshop on Image and Multidimensional Signal Processing, Cannes France, September 1993.
- [16] Lucas, B. D., Kanade, T., “An Iterative Image Registration Technique with an Application to Stereo Vision”, Proceedings of Image Understanding Workshop, pp. 121-130, 1981.
- [17] Nakayama, K., Silverman G.H., “The Aperture Problem-I, Perception of nonrigidity and Motion Direction in Translating Sinusoidal Lines”, Vision Res, Vol. 28., pp-739-746, 1988.
- [18] Barron, J. L., Fleet, D. J., Beauchemin, “Performance of Optical Flow Techniques”, International Journal of Computer Vision, 12(1), pp. 43-77, 1994.
- [19] Nagel, H., Enkelmann, W., “An Investigation of Smoothness Constraints for the Estimation of Displacement Vector Fields from Image Sequences”, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 5, September 1986.
- [20] Fuse, T., Shimizu, E., Tsutsumi, M., “A Comparative Study on Gradient-Based Approaches for Optical Flow Estimation”, International Archives of Photogrammetry and Remote Sensing, Vol. XXXIII, Part B5, Amsterdam, 2000.
- [21] Faisal, M., Barron, J., “High Accuracy Optical Flow Method Based on a Theory of Warping: Implementation and Qualitative/Quantitative Evaluation”, ICIAR 2007, LNCS 4633, pp. 513-525, 2007.
- [22] Brox, T., Bruhn, A., Papenberger, N., Weickert, J., “High Accuracy Optical Flow Estimation Based on a Theory for Warping”, Proceedings of 8th European Conference on Computer Vision, Springer LNCS 3024, T. Pajdla and J. Matas (Eds), Vol. 4, pp 25-36, Prague, Czech Republic, May 2004.
- [23] Papenberger, N., Bruhn, A., Brox, T., Didas, S., Weickert, J., “Highly Accurate Optical Flow Computation with Theoretically Justified Warping”, International

Journal of Computer Vision, 67(2), pp 141-158, 2006.

- [24] Anandan, P., "A Computational Framework and an Algorithm for the Measurement of Visual Motion", *International Journal of Computer Vision*, 2, pp. 283-310, 1989.
- [25] Amiaz, T., Lubetzky, E., Kiryati, N., "Coarse to Over-Fine Optical Flow Estimation", *Pattern Recognition*, 40(9), pp-2496-2503, 2007.
- [26] Black, M. J., Anandan, P., "The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields", *Computer Vision and Image Understanding*, Vol. 63, No. 1, pp. 75-104, January, 1996.
- [27] M. J. Matarić, M. Nilsson, and K. T. Simsarian, "Cooperative multi-robot boxpushing," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 1995, pp 556-561.
- [28] R. C. Arkin and J. Diaz, "Line-of-sight constrained exploration for reactive multiagent robotic teams," in *7th International Workshop on Advanced Motion Control*, 2002, pp 455-461.
- [29] R. Simmons, et. al, "Coordination for multi-robot exploration and mapping," in *Proceedings of the National conference on Artificial Intelligence*, 2000, pp. 852-858.
- [30] C. R. Kube and H. Zhang, "Collective Robotic Intelligence," in *Second International conference on Simulation of Adaptive Behavior*, MIT Press, 1992, pp. 460-468.
- [31] J. Liu and J. Wu. "Multi-Agent Robotic Systems" CRC Press: London, 2001.
- [32] S. Bergbreiter and K. S. J. Pister, "CotsBots: An off-the-shelf platform for distributed robotics," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2, 2003, pp. 1632-1637.
- [33] H. Asama, A. Matsumoto, and Y. Ishida, "Design of an autonomous and distributed robot system – ACTRESS," in *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems*, 1989, pp. 283-290.
- [34] L. E. Parker, "ALLIANCE: An architecture for fault tolerant multi-robot cooperation," in *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, 1998, pp 220-240.
- [35] E. P. Dadios and O. A. Maravillas Jr., "Cooperative mobile robots with obstacle and collision avoidance using fuzzy logic," in *Proceedings of IEEE International*

Symposium on Intelligent Control, 2002, pp 75-80.

- [36] T. Fukuda and S. Nakagawa, "A dynamically reconfigurable robotic system," in Proceedings of the International Conference on Industrial Electronics, Controls, and Instrumentation, vol. 2, 1987, pp. 588-595.
- [37] R. C. Arkin, T. Balch, and E. Nitz, "Communication of behavioral state in multiagent retrieval tasks," in Proceedings of IEEE International Conference on Robotics and Automation, vol. 3, 1993, pp 588-594.
- [38] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," in Proceedings of ICAR International Conference on Advanced Robotics, 1997, pp. 193-200.
- [39] M. Vainio, P. Appelqvist, and A. Halme, "Generic control architecture for a cooperative robot system," in Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, 1998, pp 1119-1125.
- [40] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. "Link-level Measurements from an 802.11b Mesh Network." In Proceedings of SIGCOMM, Aug. 2004.
- [41] J. Heidemann, N. Bulusu, and J. Elson. "Effects of Detail in Wireless Network Simulation." In Proceedings of the SCS Multiconference on Distributed Simulation, Jan. 2001.
- [42] P. De, A. Raniwala, S. Sharma, and T. Chiueh. "MiNT: A Miniaturized Network Testbed for Mobile Wireless Research." In Proceedings of IEEE INFOCOM, Mar. 2005.
- [43] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H.Liu, and M. Singh. "Overview of the ORBIT Radio Grid Testbed for Evaluation of Next Generation Wireless Network Protocols." In Proceedings of the IEEE Wireless Communications and Networking Conference, Mar. 2005.
- [44] B. N. Chun, P. Buonadonna, A. AuYoung, C. Ng, D. C. Parkes, J. Shneidman, A. C.Snoeren, and A. Vahdat. "Mirage: A Microeconomic Resource Allocation System for SensorNet Testbeds." In Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors, Sydney, Australia, May 2005.
- [45] Amir Reza Momen, Jahangir Dadkhah Chime and Rey Branch. "A New Analytical Method in User Mobility Modeling in Wireless Network." In Proceedings of the International Conference on Information and Communication Technologies: From theory to applications, 2004.

- [46] J. Broch, D. A. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva. “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols.” In Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), pages 85–97, Dallas, Texas, October 1998.