Universidade de
Aveiro
2002

Departamento de Engenharia Mecanica

Caetano Filipe Costa de
Noronha Ferreira

Utilização de Padrões de Medidas Laser para Localização de um Robot Móvel

Mobile Robot Localisation using Laser Range Patterns

DOCUMENTO PROVISÓRIO

Universidade de Aveiro
2002

Departamento de Engenharia Mecanica

Caetano Filipe Costa de Noronha Ferreira

Utilização de Padrões de Medidas Laser para Localização de um Robot Móvel

Mobile Robot Localisation using Laser Range Patterns

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecanica, realizada sob a orientação científica do Dr. Vítor Manuel Ferreira dos Santos, Professor Auxiliar do Departamento de Engenharia Mecanica da Universidade de Aveiro

O júri

presidente

Prof. Dr. José Joaquim de Almeida Grácio
professor associado com agregação da Universidade de Aveiro

Prof. Dra. Maria Isabel Lobato de Faria Ribeiro
professora associada com agregação do Instituto Superior Técnico da Universidade Técnica de Lisboa

Prof. Dr. Vítor Manuel Ferreira dos Santos
professor auxiliar da Universidade de Aveiro

agradecimentos

Quero em primeiro lugar agradecer ao Professor Vítor Santos, meu orientador de tese, pelo apoio que me deu e pela paciencia que teve comigo ao longo deste último ano e meio. Presente sempre que necessitei, insistiu, no entanto, para que fosse eu a arranjar soluções para os problemas que foram aprecendo ao longo do trabalho. Foi por esta razão entre outras, a tese uma experiencia muito interessante e agradável para mim.

Quero ainda agradecer à Universidade de Aveiro pela bolsa de acolhimento que me foi atribuída enquanto decorreu o mestrado.

Ao Professor José Grácio e à Doutora Susana Sardo um obrigado especial por me terem trazido até Aveiro, onde hoje me considero em casa.

Aos colegas do Departamento de Engenharia Mecânica, das àreas de Tecnologia Mecânica e da Automação, pela amizade e camaradagem que me ofereceram desde a minha chegada a Aveiro.

Por fim, não posso terminar sem dirigir um obrigado muito especial à minha familia e amigos pelo apoio, amizade e tambem paciência!

resumo

O objectivo deste trabalho foi o desenvolvimento de um algoritmo para a localização de um robot móvel equipado com um medidor de distâncias laser. O processo devia integrar-se facilmente na linguagem para a definição de missões de navegação, LAMP, previamente implementada no robô.
O primeiro passo deste procedimento é a procura nos dados laser de alguns objectos especificos. Estes objectos (faróis), inseridos expressamente no ambiente para o processo de localisação estão arranjados num certo padrão geométrico que permite a sua identificação. Esta identificação é seguida por um processo de confirmação onde os dados laser são examinados novamente para a existência de outra formação da sala cuja posição é descrita a partir dos potenciais faróis.
A estimativa de localização é obtida através de “trilateração” e do conhecimento da posição dos faróis no sistema de coordenadas global.
Numa fase final do trabalho, foram introduzidos simultaneamente vários faróis, arranjados de formas diferentes.
Os resultados indicam um desempenho bom do algoritmo com erros na estimativa de posição de menos de cinco centímetros e erros de orientação de menos de um grau. A integração simples no sistema de navegação existente permite a utilização do método para um arranque fiável de uma missão de navegação.

abstract

Mobile Robot Localisation using Laser Range Patterns: The work presented here involved the development of an algorithm to utilise data obtained from a Laser Range Finder -mounted atop a 4-wheeled robot - to estimate the pose of the robot (Coordinates x, y, and θ). The complexity and concept behind the method were to be kept in line with the LAMP navigation language already implemented on the same robot.

A two stage procedure was designed by which the algorithm searched for points ranged on particular objects in the scan. These objects (beacons), planted specifically for localisation, are arranged in a certain geometric pattern which enables their identification. This identification is then followed up with a confirmatory procedure in which the scan is probed for the existence of some known permanent feature whose position is described relative to the beacons. The location estimate is then obtained using trilateration and a knowledge of the beacon position in the global coordinate system.

An extension of the procedure saw the introduction of more than one set of beacons into the environment with the algorithm searching the laser scan for each possible configuration.

The results indicate good performance with position estimate errors of less than five cm and orientation errors of less than one degree.A simple integration into the existing navigation system allows the method to be utilised to provide for a purposeful and abreviated start to a LAMP navigation mission.

Contents

Figures

1. Introduction

Today, as some professionals in the field of mobile robotics will testify, many of the problems in the field of localisation and navigation of Mobile Robots have been solved. Although there still does not exist one unique solution to satisfy the vast gamut of problems, several techniques and approaches showing robust performance have emerged. Nonetheless, the quest for increased robustness and versatility continues.

Localisation is the process by which the robot control system makes use of sensors or some other means to estimate its position in the real world. Various ideas have been put into practice over the years in an attempt to obtain satisfactory estimates of the location of robots at various instances. With the continuous improvement in sensor technology and their ever-reducing cost, more ambitious implementations of localisation algorithms have appeared. Research work in localisation is developed with particular operating conditions and specific environments or types of environments in mind. Tailoring the localisation procedure to the level of hardware and software sophistication of a particular situation is essential.

1.1. Framework

Activity on robotics at the Department of Mechanical Engineering, University of Aveiro, began in early 1997 when a mobile robotics platform (Robuter III from Robosoft) was purchased by the department. The main thrust has been to develop techniques for autonomous and semi-autonomous navigation, which obviously must rely on more or less developed perception. Early activities created a framework for robot higher-level control, by developing a modular navigation architecture where perception consisted only of odometry and ultrasonic sensors [Oliveira et al., 1999]. A higher level of control was achieved with the introduction of the concept of navigation mission made up of navigation tasks, intended to define a more robust navigation application. After a study on gyroscopes [Oliveira and Santos, 2000], a gyroscope from KVH was added to overcome some limitations imposed by odometry, and even to allow for enhanced navigation [Oliveira and Santos, 2000b]. In addition, a laser range finder from SICK was installed to overcome some of the limitations of the ultrasound, and navigation reached a new level of sophistication allowing automatic crossing of doors with much more efficiency than with ultrasonic sensors. The navigation mission concept had reached an interesting level to evolve towards robust autonomy [Santos and Oliveira, 2001].

In this context, from the remaining issues necessary to address for a more complete navigation system, robot self-localisation was the one with no dedicated activity so far. The presence of the laser range finder upon the robot allowed for an implementation of a laser-based localisation.

1.2. The robot platform

The robot platform utilised in this work is the, four wheeled Robuter III from Robosoft SA, France equipped with twenty-four Polaroid ultrasound (US) sensors, an eye-safe Laser Range Finder (LRF) from SICK, a system for odometry and a fiber-optic gyroscope. The independently powered rear wheels allow for velocity and directional control. The odometry is updated every 50ms and has a resolution of 0.1mm.

The US sensors have a range of around 8 meters with uncertainties of under 1cm under optimum conditions with low measurement rates of around 4-5 Hz. These were included in the original platform and are controlled by the robot hardware.

The LRF is a SICK LMS200 laser range finder that provides a 180º-wide depth scan. The resolution, maximum measurable depth and other parameters of the range scanner can be set up as required.

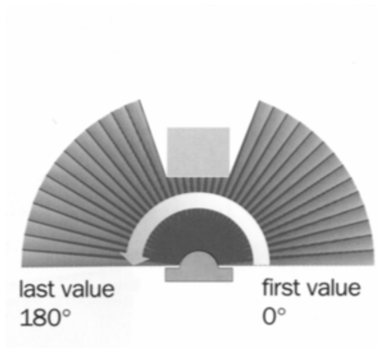


Figure 1-1. SICK laser range finder LMS 200 with direction of scan

An IBM PC clone, running Linux, and referred to as the 'onboard computer', is placed on the robot. This machine, at present, is used primarily to control the communication link between the robot and the stationary remote computer. It is connected via serial and parallel ports to the robot CPU and by means of an Ethernet card to a station of a wireless Ethernet modem. The role of this computer, however, is set to grow in importance, as it shall host some navigational functions in the near future. The connections to the robot are made through the serial and parallel ports. One serial and the parallel port are used for the transfer of commands and programs to the robot CPU while the other serial port is exclusively used to monitor processes running on the Robuter. The ports on the robot CPU are utilized for the purpose of data exchange with the onboard computer and for communication with the LRF and the gyroscope.

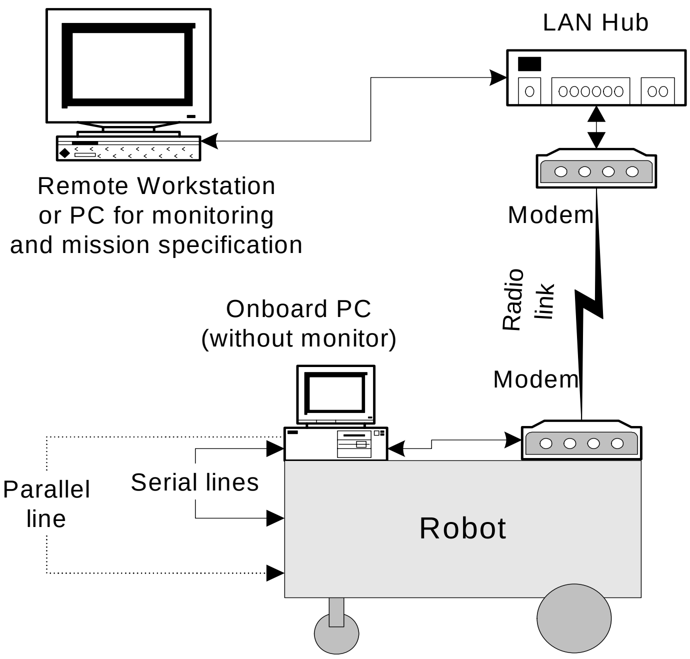


Figure 1-2. Communications set-up

The fixed station of the modem is connected to the Department of Mechanical Engineering network through a LAN hub. This arrangement allows any computer having access to the LAN to act as the remote workstation in order to 'telnet' the onboard PC, execute commands and transfer programs and data to and from the robot CPU.

The robot runs a real-time operating system called Albatros^{tm}, created specially to run multi-axis or multi-sensor machines efficiently. The system has a real-time kernel, I/O drivers, a generalized PID trajectory generator, sensor read modules and a command interpreter. The system can also run application programs developed by users.

1.3. Navigation and mission control

In earlier projects of the mobile robotics and automation laboratory (LAR) of the Department of Mechanical Engineering, a mission programming language has been developed for use on the robot platform. The Language for Autonomous Mission Planning (LAMP) is a tool that can be used by an operator to set up a mission, using a qualitative description of the topology of the environment [Santos et al., 2001]. The kernel feature of LAMP is that the robot mission may be planned and executed by utilizing the approximate position and layout of entities to trigger the beginning and the end of individual phases of the mission.

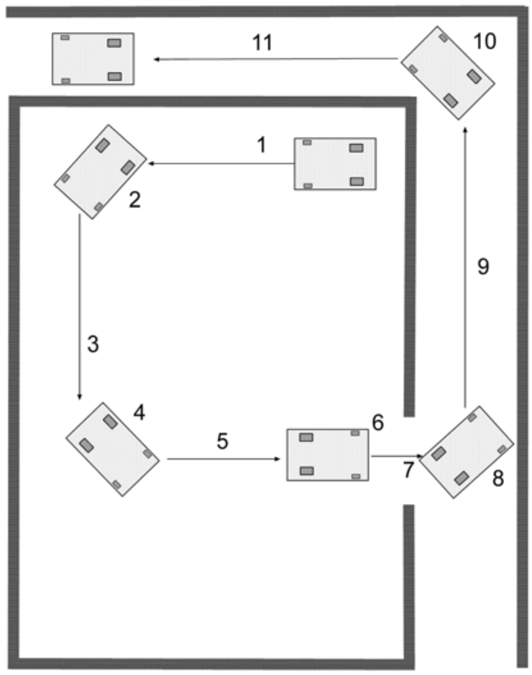
Although the absolute position of the robot through odometry is maintained, the same is not utilised and the detection of objects in front of the US sensors provides a sort of external referencing. Individual stages are set up either as closed loop feedback or as open loops. This method partially obviates the necessity of the robot knowing where exactly it is in the environment. Setting up one or more initial stages allows the robot to take a particular position and/or orientation, and thus work without an initial location estimate.

MOVE LV –20 AV 0 USG 800 SEN 13	Move backwards (straight down, linear velocity 20, angular velocity 0) till sensor 13 measures less than 800 mm.
MOVE LV 0 AV –10 ANL 90	Turn in same place, Clockwise Through 90º
MOVE MV 25 PS 2 USG 700 SEN 1 DIL 1000 OR	Follow parallel wall on side 2 at a mean velocity of 25, till sensor 1 meets obstacle at 700mm or till at least 1 meter of distance is covered
SETP PS2 SD 0	Set parallel to side 2 (distance no issue)
CROS SD 900	Cross doorway and travel 900 mm
MOVE LV 15 DIL 700	Go 700 mm in straight line

Table 1-1. Some typical LAMP commands

By combining the low-level instruction sequences that the Albatross[tm] provides, with the data from the sensors (Ultrasound sensors, LRF, Odometry and Gyroscope), LAMP allows the robot to follow flat walls, stop upon encountering a feature somewhere around the robot, cross doorways, etc. An illustrative mission is shown in Figure 1-3.

While the method has proved to be quite successful for the purpose for which it was intended, it still has undeveloped features in the areas of obstacle avoidance and a means for absolute localisation. The latter would allow a mission to be less specific and less dependant on the robot's initial position and orientation. It would also save on mission execution time. The present work aims to help fill the void left by the lack of such a means for initial localisation of the robot.



Sequence of 11 steps to execute the mission presented in the figure at left

1. MOVP MV 20 PS 2 USL 500 SEN 1
2. MOVE LV 0 AV 10 ANL 90
3. MOVP MV 20 USL 1000 SEN 1
4. MOVE LV 0 AV –10 ANL 90
5. MOVE LV 20 DIL 1500
6. CROSS
7. MOVE LV 10 USL 300 SEN 1
8. MOVE AV 20 ANL 90
9. MOVP MV 20 PS 2 USL 500 SEN 1
10. MOVE AV 20 ANL 90
11. MOVP MV 20 PS 1 USL 300 SEN 1

Figure 1-3. A navigation mission with associated LAMP code.

1.4. Interfacing with the FMS

As part of an ongoing project, there is the need to interface the mobile autonomous robot with the Flexible Manufacturing system installed in the CIM laboratory at the Department of Mechanical Engineering. In this situation, the robot will play the role of a flexible AGV.

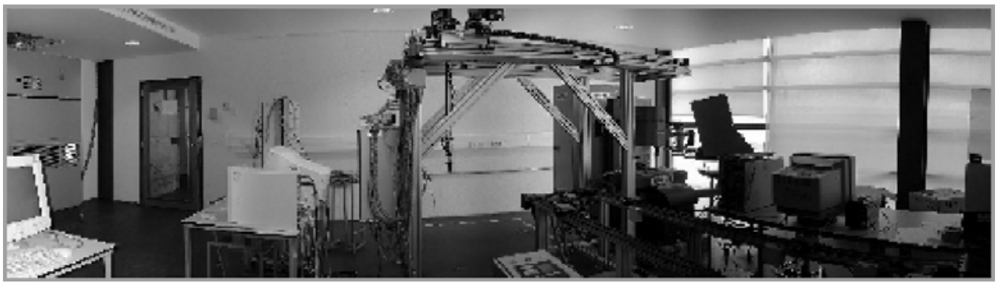


Figure 1-4. View of the CIM with the gantry robot and door connecting to the LAR

More specifically, the Robuter must interface with a gantry robot, which at present simulates an Automated Storage and Retrieval System (ASTR). For this purpose, the Robuter will have to approach the gantry robot correctly, position itself beneath it and relay the host computer the exact position of the part pallet with respect to a coordinate system that the gantry robot can also reference. The precision of positioning required for this task is relatively high, typically of the order of one or two centimetres.

The FMS runs using a proprietary ASCII-based messaging protocol implemented on a common Ethernet network. Using DDE transfer between the host computer and the various workstation controllers, ASCII messages are exchanged, through the Ethernet network of the department, between the host and remote stations with instructions, acknowledgements and combinations thereof [SantosJ et al., 2000]. Since the robot is accessible to the FMS controller, the robot shall be requested to approach the gantry robot, and queried as to its present position.

1.5. The Problem to be solved

The Figure 1-5 shows the workspace of the robot without any furniture, a region that spans two laboratories; the Laboratory for Automation and Robotics (LAR) and the Centre for Intelligent Manufacturing (CIM). A typical robot mission might involve starting up from its parking position, moving towards the door that joins the two laboratories, negotiating the narrow passage, heading for the gantry robot and positioning itself after which a part transfer might be executed.

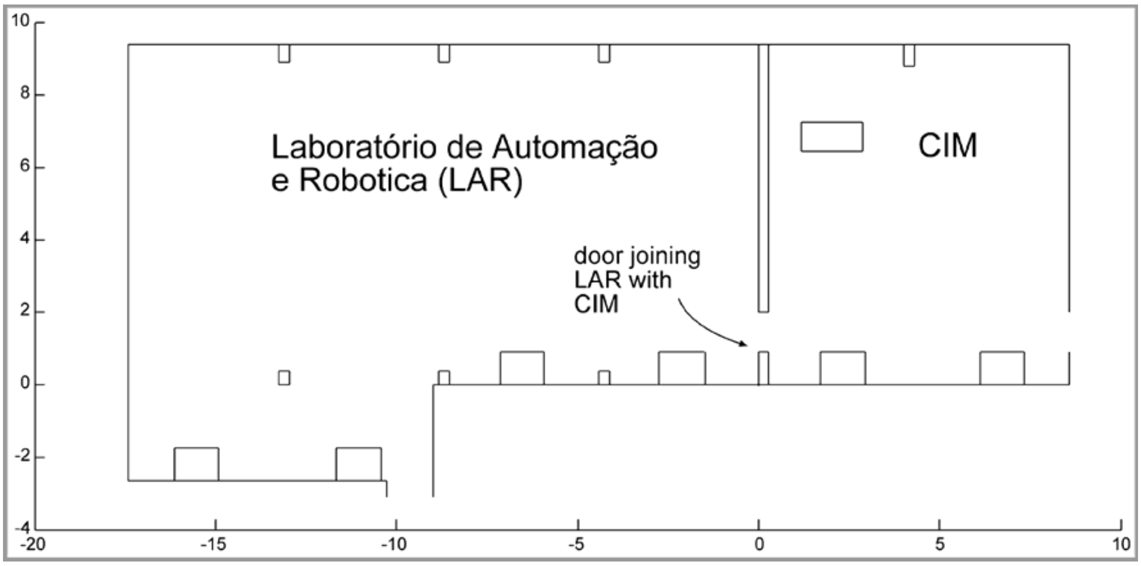


Figure 1-5. View of the robot workspace, laboratories LAR and CIM

Localisation of the robot was defined as being necessary for two distinct stages of the robot mission. The first is at the time of initiating a mission and the second at the instance of interaction with the gantry robot of the FMS. These two stages of the robot mission require that the robot rely on some means of position estimation in order to be able to perform its tasks.

1.6. Initial localisation

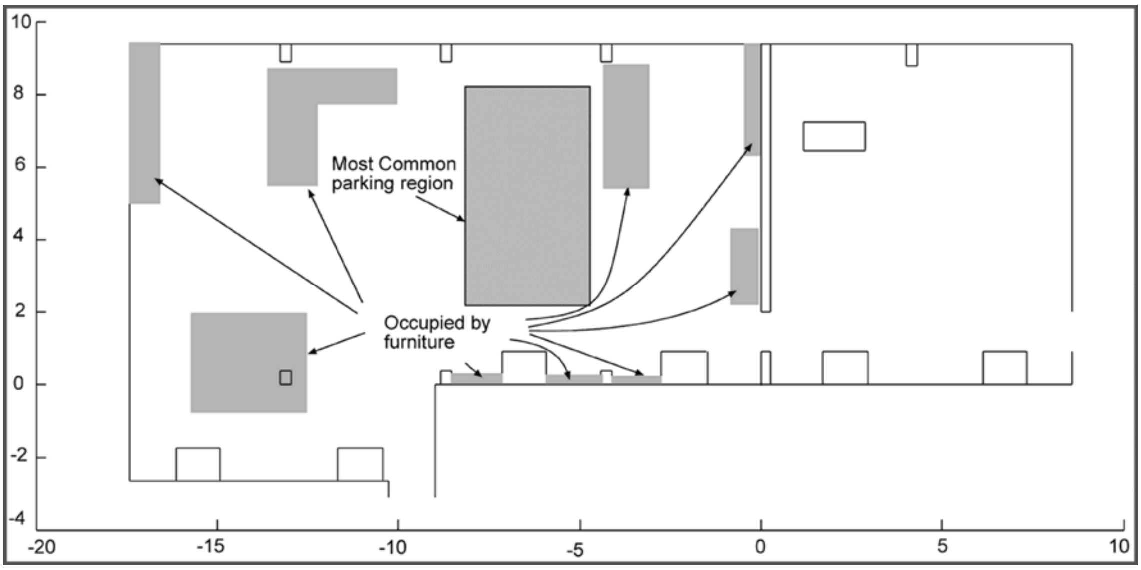


Figure 1-6. View of the LAR with depiction of typical parking area of the robot

This procedure can be characterized by two assumptions:

1. No estimate of the robot position is available.

2. The need for localisation is expected to occur within the confines of a section of a hall, in which the robot is usually parked.

It is pertinent to note that while the X and Y starting coordinates of the robot are coordinates of interest, the initial orientation usually has a greater utility for the LAMP mission definition.

It is sought to automate the initial localisation procedure so that it can be seamlessly integrated within the existing mission definition. Since the system at the time of start-up has no other way of verifying the location estimate as provided by the localisation algorithm, the automation of this procedure requires that the probability of erroneous detection be kept to a minimum.

1.7. Final localisation

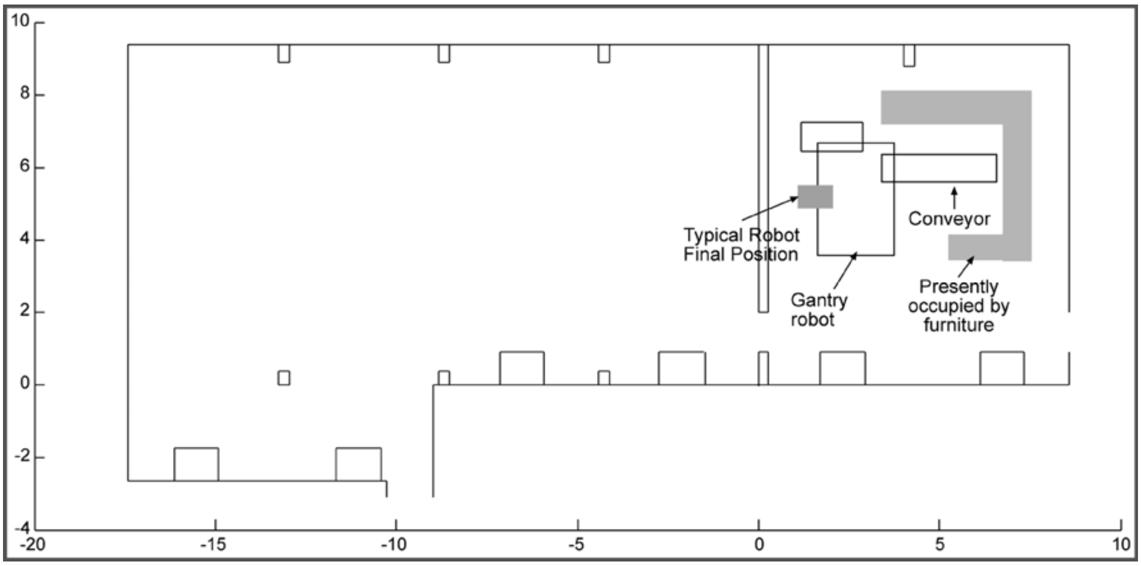


Figure 1-7. View of the CIM showing gantry robot with robot final position

The robot must perform some of its functions in conjunction with the Flexible Manufacturing System (FMS) installed in the CIM laboratory. More specifically, the robot is to behave like an Automated Guided Vehicle (AGV) interacting with the computer controlling the FMS and transferring parts to and removing parts from the FMS with the aid of the gantry robot. To succeed in this role of AGV the robot has some position estimate needs.

It must obtain a location estimate other than the one provided by an odometry update of its initial position. This allows for a relatively more precise localisation next to the FMS, the culmination of a robot mission. It also signifies that the error of the position estimate at the gantry robot is independent of the length and nature of the path that the robot followed, allowing the system mission capabilities to be augmented.

In its role as an AGV, the interaction with the gantry robot of the FMS shall be utilized to transfer objects from the robot to the gantry robot and vice-versa. The upper bound on the

estimation error of the location estimate must be evaluated. In addition, a communications protocol must be implemented for the transfer of position requests, and confirmation commands.
It was decided to utilize the same method developed for initial localisation to allow the robot to estimate its position at the gantry robot. An evaluation of the suitability of the algorithm to perform localisation with sufficient accuracy is necessary.

1.8. A Brief description of the proposed solution

If localisation is to be done with the aid of landmarks, the most attractive choices for use, in the interests of keeping the environment untouched, are natural features such as furniture, pillars or other structural features of the building.
However, the absence of prominent similar features throughout the extents of the laboratories prompted the use of artificial features inserted in the environment to aid in the localisation process. The solution that was adopted, in an attempt to emulate pillars or similar natural features, was the erection of multiple poles (PVC pipes and vertical strips of paper were utilised), arranged in a regular pattern and oriented in some known way with respect to some other, permanent, feature of the environment. The process of localisation attempts to locate any feature that might be present (the individual distinct beacons that make up the 'landmark' together with the additional information as regards the relation of the feature with regard to the environment) and match it with the list of features that it has stored within the program. The program contains some information about the features, including its absolute position, allowing for an estimation of the location of the robot.
The process of identification of the artificial landmarks has been broken up into two-stages:
In the first stage it is attempted to glean out 'possible' individual beacons from the data obtained from the laser range scan. Differences in the pattern and in the arrangement of the beacons allow more than one landmark to be set up at any time.
The second stage involves confirming the hypothesis generated in the earlier stage. For each of the hypothesis generated, additional available information is utilised in order to confirm the identification of the beacons. This additional data is chosen in such a way that it is as unique as possible for each feature and identifies a landmark with a high probability.
A pre-processing stage has also been incorporated to improve computational efficiency and increase algorithm efficacy.
While initial algorithm development was attempted in C++, frequent alteration and a greater reliance on simulation to need to aid algorithm development led to the adoption of Matlab for Windows® environment. The program developed in this environment was then ported without much difficulty into the open source Matlab-like program Octave, which is provided along with most installations of Linux. Some functions that exist in Matlab had to

be re-written for Octave. Computationally intensive functions were precompiled using the ‘wrapper’ classes provided in Octave.

1.9. Outline of this document

Chapter 2 attempts to place the present work in the framework of research that has been carried out in the field of localisation. It presents the state of the art, focussing on localisation methods that specifically utilise LRF scanning. Chapter 3 describes the work undertaken, the physical characteristics of the landmarks as well as the software developed for the purpose of the algorithm. Chapter 4 presents the results obtained from the implementation of the algorithm. It presents details of the performance of the algorithm in different settings. In chapter 5 the way to set up the program for a particular beacon configuration and positioning is explained. Chapter 6 summarises the conclusions.

2. The State of the Art

2.1. Introduction

The problem of robot navigation in mobile robotics can be thought of, as having three broad, and often overlapping, sub-categories:

1. Robot Localisation
2. Environment mapping and
3. Robot planning and control

Most research in the field of mobile robotics is carried out in these three areas. They have been treated either as separate areas or as interdependent facets of the same problem. The solutions that have emerged either group together independent problem resolution techniques or result in new solutions that try to solve more than one problem simultaneously. The complexity of the problem at hand and the limitations imposed by the computing and sensing hardware has resulted in the fact that, despite there being many techniques that solve the problem of robot navigation most of these either impose severe constraints on the environment or attempt to solve a sub-problem in the field.

While there have been quite a few relatively elegant solutions presented, to aid in the resolution of the problems of localisation and mapping taken individually (the location being known and the map unknown or vice versa), it has been only over the last decade that the resolution of the first and second problem, taken simultaneously, has had some success. Concurrent Mapping And Localisation (CML) or Simultaneous Localisation And Mapping (SLAM) are terms used to refer to algorithms that attempt to solve the problem of robot localisation and environment mapping and updating at the same time [Thrun, 2002].

Over the next few sections a brief examination of the current state of the art in the field of localisation is carried out. While issues in environment mapping are beyond the scope of this work, they will be mentioned in the course of the description of techniques in which localisation is inextricably linked to mapping.

The next section attempts to situate current efforts in the history and evolution of research in the area of mobile robots, followed by a section that takes a look at the big picture in the area of localisation. A range of techniques that are current topics of research are mentioned here in brief. The last section introduces some of the work carried out with laser range finders by researchers in the field as a precursor to the presentation of the details of this work in chapter 3.

2.2. General trends in robotics research

The following was compiled based on a presentation by W Burgard at a summer school at the EPFL, Laussane in the summer of 2001[Burgard, 2001].

2.2.1. Classical robotics

With the onset of widespread automation in automotive plants in European and Japan, in the mid–70's, similar technologies began to be applied to the field of mobile robotics. Namely, these techniques involved the resolution of tasks that could be defined with exact models. Sensing was kept to a minimum or altogether dispensed with. This initial phase was marked by the desire to show the usefulness of robots at repetitive, unchanging tasks. Versatility was still relatively unimportant at this stage.

2.2.2. Reactive paradigm

In an attempt to allow robots to live in the Real-world and work with others of their kind or with other types of entities, such as humans, robots began to be equipped with sensors, typically ultrasound sensors and, to some extent, light based devices. This phase can be characterised by the absence of an overall model that could utilise sensor inputs to attempt to convincingly produce an answer to the question "Where am I?" except in the case of a few extremely simple situations. Sensors were thus utilised to enhance safety, detect end of path and path interference. The emphasis here seemed to be on the completion of repetitive tasks with the added advantage of obstacle avoidance and some other interaction with the environment. In addition, robots began to be equipped with some computing power that could be programmed and re-programmed to deal with varying tasks, instead of the hardwired machines of the earlier generation.

2.2.3. Hybrids

With the development of reliable sensor technology and computers, robots began to experiment doing non-standard tasks. Equipped with the ability to better read their environment and a high-level model that tried to estimate the position after having accumulated and integrated the various sensor inputs, robots could now begin to attempt to answer the question "Where am I?" in a number of situations. Also, the robot tasks began to be separated out in levels. By organising the various tasks in a hierarchical fashion with basic tasks left to be implemented by independent modules at a lower levels and comparatively complex tasks being carried out by successive higher levels designers could now concentrate on the 'whole picture' and better implement models.

2.2.4. Probabilistic approaches to robotics

Robots soon came to be equipped with sophisticated models and better sensorial capabilities, developed so as to allow them to better respond to solicitations made by 'perfect' or laboratory-controlled conditions. Researchers now turned their attention to less than perfect environments in which the earlier class of methods would lead a robot to get lost either because the robot's knowledge of the environment was imperfect (or the environment itself had changed with time), because the robot's sensors were not able to give a sufficiently clear and accurate idea of the environment around it, or because the actual environment was simply too complex for any exact model to handle.

In short, robots have to work in a world of noisy sensors and incorrectly represented environments that often change with time. In other words, both, the location of the robot and the map are uncertain. The answer here is to attempt to develop methods that account for inaccurate sensors, inaccurate models of the environment and attempt to compensate for this inaccuracy by relying on other information, typically from the past. This generation of robots, which have been around the mid-90's, has also seen a better integration of environment models and sensing.

2.3. Probabilistic techniques to handle sensorial inputs

As mentioned above, sensor noise is an ever-present problem in mobile robotics. A study of the behaviour and limitations of the sensors employed usually allows for a development of a "sensor model" which attempts to relate the output provided by the sensor with various input parameters or factors that might affect sensor performance. By exploiting the uniqueness, or rather, the relative uniqueness of the sensor readings obtained under certain conditions, non-plausible possibilities can be eliminated and a measure of sensor reliability can be obtained. In mathematical terms, the concept of probability is used to quantify the chances of obtaining a particular set of sensor readings or, of being at a particular place. The principals of conditional probability are made use of in order to combine data from many sources to evaluate the probability of the outcomes. Baye's rule of conditional probability is the cornerstone of methods for this purpose.

2.4. The basics of localisation

Localisation is the process by which the robot control system makes use of sensors to estimate its position in the real world. With the continuous improvement and reducing cost of sensors, more ambitious forms of implementation have appeared. Most of these methods utilise one or more of the following basic methods and are utilised both for absolute localisation and pose tracking.

2.4.1. Dead-Reckoning methods

These are associated with the integration of incremental-motion information over time and are essentially pose tracking methods. Almost all commercially available robots come equipped with encoders that enable the robot control system to perform odometry. The easy access to this source of data means that most localisation systems make use of odometry data in their localisation algorithms. Another group of sensors that allow dead reckoning are heading sensors. Gyroscopes and accelerometers fall under this class of devices. These devices were quite expensive in the past but there has been a fall in the price of fibre-optic gyros, making them a very attractive proposition [Oliveira et al., 2000b].

The advantages of dead-reckoning methods derive from the fact that they are completely self-contained (do not require interaction with the environment) and allow high sampling rates. Their principle of operation, however, results in their continuously accumulating errors in proportion to the passage of time and distance covered. This means that besides the position estimate fed to the robot at start-up, correction of the odometry error is required if reliability is to be maintained over an extended distance and time. The following is quoted from [Borenstein et al., 1996] as being the reason for the continued popularity of dead-reckoning methods:

> "Odometry data can be fused with absolute position measurements to provide better and more reliable position estimation.
>
> Odometry can be used in between absolute position updates with landmarks. Given a required positioning accuracy, increased accuracy in odometry allows for less frequent absolute position updates. As a result, fewer landmarks are needed for a given travel distance.
>
> Many mapping and landmark matching algorithms assume that the robot can maintain its position well enough to allow the robot to look for landmarks in a limited area and to match features in that limited area to achieve short processing time and to improve matching correctness.
>
> In some cases, odometry is the only navigation information available; for example: when no external reference is available, when circumstances preclude the placing or selection of landmarks in the environment, or when another sensor subsystem fails to provide usable data."

2.4.2. Active beacon navigation systems

In this class of methods, beacons emitting or receiving some energy, designed so that they can be easily picked out, are inserted in the environment. The robot is usually equipped with some means of picking up the angle that the beacons make with the robot axis and/or their distance from the robot. These are absolute positioning methods in which the absolute position of the beacons has to be known. The absolute position of the robot is then estimated using trilateration or triangulation. Three or more landmarks must be used in order to solve a typical pose problem (x, y, θ), unless some additional constraint is placed upon the robot's operating space with respect to the beacons. The equipment is based either on radio, ultrasound, laser or infrared transducers. Outdoor systems utilising GPS have recently proved to be accurate and relatively inexpensive given their performance [Gray, 2000]. In certain cases an indoor GPS system is set up, though this method is expensive and does not lend itself to use in many indoor environments. Wulf [Wulf, 2001] describes a method by which a robot is led to a visible charging station with the aid of an IR beacon.

2.4.3. Landmark navigation

This class of methods utilise beacons that reflect/return ambient or directed energy onto transducers that allow the robot control system to detect them from the input data. Landmarks can be natural or artificial. Artificial landmarks are those that are introduced in the environment for easy recognition by the robot sensory input device. Additionally, they might carry additional information such as a bar code or some special unique geometric characteristic. Methods utilising ‘natural landmarks’ usually work well in highly structured environments and are usually man-made [Borenstein et al., 1996]. The sensor of choice in this case is usually vision [Madsen and Andersen, 1998].

Usually the various methods work using features that contrast to the rest of the sensed environment. Many techniques get around the difficulty of choosing good landmarks by designing algorithms that, automatically, select landmarks during an initial training phase and utilise a similar method to pick out image features during actual localisation. The criteria for selection are the reliability (read uniqueness and high contrast) of a landmark and a low probability of occlusion. The performance of these methods is usually affected by various conditions including lighting, angle at which the image is taken and distance of the landmark from the robot [Sim and Dudek, 1999], [Thompson and Zelinsky, 2000].

While it is uncommon to see ranging devices being used without vision in such methods, features with distinct signatures such as edges, long walls might be chosen as landmarks to be extracted using range finders [Santos and Oliveira, 2001].

The detection of artificial landmarks is usually much easier as these are optimised in terms of uniqueness, position and ease of separation (contrast) from the rest of the image. Variously shaped figures and objects have been placed, by different researchers, in the environment in attempts to come up with a fast and reliable method of detection. Neural networks have also been utilised in the extraction of artificial landmarks from images. In the case of outdoor navigation, especially, the above methods have shown great effectiveness.

Another point of interest in the case of continuous pose tracking using Landmark detection (both natural and artificial) is that the robot should usually know its approximate position, since this allows it to search for landmarks faster.

2.4.4. Map-based positioning

This class, if they can be considered so, of methods work by using onboard sensors to draw up a local map of the environment after a pre-processing stage in which data from the sensors are filtered, evaluated and finally integrated. The system then compares the local map obtained at the current point with an appropriately modified image of a stored global map. Different procedures might be used to match the global map with the local map [Cox, 1991], [Drumheller, 1987], upon which, an estimate of the position of the robot is obtained. Recently, a lot of research is being carried out in this field and some very interesting results have been obtained. The sensors of choice are ranging sensors (formerly

Ultrasound and more recently Laser Range finders), though the algorithms sometimes are provided with vision and dead reckoning capabilities.
The global map can be obtained and represented in a number of ways. Two representations are frequently seen: geometric and topological representations, though it difficult to clearly classify the approach utilised by most methods.
A geometric map represents subjects according to their absolute geometric position. It can be represented using a grid structure –a discrete representation, or using lines and polygons in a continuous representation of the environment. Using data from the robot's current and past sensor measurements the existence and position of the various features are evaluated and updated. Topological maps on the other hand focus on the geometric relationship between features rather than on their absolute location in a coordinate of reference [Pradalier & Sekhavat, 2002], [Vale et al., 2001].

2.4.5. Vision based positioning

The large processing power now available on personal computers, together with reduction in the cost of digital cameras has resulted in a focus of attention on the possibility of localising a robot using a very intuitive method -stereovision. Over the last two decades research in vision has resulted in big strides being made in the development of projective theory and practical methods to be applied in the case of vision sensors for quick image tracking and interpretation.
In the case of stereovision, the problem essentially consists of trying to interpret the 3-D surroundings utilising two cameras aimed at the same scene but separated by some distance and angle. The problem boils down to identifying correspondent features in the two images. An extension of the problem involves calculating the position of the point in the environment that results in the two correspondent images. This might subsequently be used in order to reconstruct the surrounding environment.
Digital cameras have also been utilised as sensors in order to extract certain environment features that might be easily reflected in a video image of the environment. For example, a large number of methods utilise a single camera, as the sole sensor, [Thompson and Zelinsky, 2000] or together with some other sensor such as a range finder [Arras and Tomatis, 1999], in order to extract edges. Many indoor environments abound with such edge data, commonly generated by wall edges, door, window and wall-picture frames, furniture outlines etc. This data is then integrated with the data obtained from other devices or from a possible position estimate on a previously constructed map and utilised for localisation.

2.5. A word on probabilistic localisation

The navigational requirements of robots, in most cases, require relatively precise position estimates. The problem of sensorial noise coupled with imprecise environment models work to induce uncertainty in the estimate (after obtaining one). At any instance, a sensor-equipped robot is armed with more than simply the readings taken by its sensors at that point. Namely, it might also possess sensor readings from (a) previous position(s), an estimate of the last position it occupied, two or more independent forms of estimating the current position, and/or an estimate of the relative location and state of other objects or entities in the environment.

Some or all of this data and information might be utilised in order to improve the estimate of the localisation process. Through a variety of implementation schemes, past observations might be taken into account and provisions allowing the state of some entities to change over time (transitions) might be added. Bayesian filters allow the incorporation of new information obtained at each step into a robust estimate.

The localisation problem involves the resolution of one or more of the following tasks [Burgard, 2001]:

- Position tracking
- Global Localisation
- Recovery from kidnapping

Some of the methods implement in the resolution of these tasks and which explicitly use probabilistic techniques are:

- Kalman Filter- based systems
- Multi-Hypothesis Tracking systems
- Grid- based Systems
- Topological maps
- Methods that make use of particle filters

2.6. Localisation using laser ranging devices

With the reduction in the price of laser ranging devices, the high sampling rates possible, and their clear superior performance when compared to ultrasound, they appear to have become the ranging device of preference in mobile robotics. In many cases, range finders are used as the principal sensor, in conjunction either with odometry, vision and ultrasound or a combination thereof. Most of the methods mentioned below are essentially pose tracking algorithms, though some also include an initial localisation method.

In [Cox, 1991] readings from a LRF are subjected to an iterative procedure of rotation and translation till ‘convergence’ results in the inverse transformation that produces the robot position estimate. Here the objective function to be minimised is the distance of each scanned point from the nearest line (as represented in a line model of the environment). The iterative procedure works so as to translate and rotate the scan till the sum of squares of these distances is minimised. Odometry data assists in reducing the iterations and in situations in which matching fails to yield an estimate.

Ribeiro and Gonçalves in [Ribeiro and Gonçalves, 1996] utilise pairs of vertical edges to obtain a localisation estimate. The environment is scanned selectively in the direction in which the edges are expected. An environment model is utilised in order to obtain the absolute position of the edges and to choose from among various possible edges. An initial position estimate, obtained, in case of a moving robot, from odometry is utilised to aid the search procedure and to strengthen robustness of the estimate.

In [Arsénio, 1997], a LRF is mounted atop a Pan and Tilt Unit (PTU) and the laser beam is utilised to obtain a depth picture of the robots surroundings. This comprehensive work includes an algorithm that has access to a 3-D representation of the hall. In this map, objects with vertical edges, together with the sides that make up the edge, are represented, chosen on the basis of contrast and probability of being observed. After acquisition of the laser scan, vertical edges are extracted and an attempt is made to match them with the vertical edges of objects in the map. To simplify computation and improve results, some pre-processing of the data is carried out. In addition, the scanning for features based upon an initial estimate of the robot's current position is performed in order to identify what objects might appear in the laser scan and what objects might be partially or fully occluded. Occlusion effects and the range of angles through which each edge is visible from each cell are taken into account. In order to account for the presence of more than one landmark a probabilistic method, Hough clustering, is utilised in which clusters of transformation parameters are obtained. While the work described is essentially used for pose tracking, whenever the error in positioning is very large a variant of the method is utilised to localize without a previous estimate.

Another example of a feature-based system that utilises a LRF in conjunction with odometry is [Jensfelt and Christensen, 1999]. Here, a very reduced representation of the world model is utilised, usually limited to four walls representing the outer extents of the laboratory. Armed with the assumption of orthogonal walls, the values of the distance to the walls are used to update the position estimate, which, otherwise is updated by an odometry reading. The X and Y Cartesian coordinates are updated alternatively using different walls. The error in the angular orientation, a more serious problem in odometry-based systems is corrected at every update.

In [Sequeira and Gonçalves, 1993], a sweep of the LRF is utilised to give a scan, which is then matched with a line-representation of the hall in which the robot finds itself. Utilising a modified version of Cox's [Cox, 1991] algorithm, an attempt is made to match the scan obtained with the line representation. The paper reported very good precision and a good performance, this considering the bulky hardware and slow processing power then available. However, from the scan representations presented, the hall seems to be quite bare and contains few objects that significantly alter the rectangular space.

While this section deals with laser range finder–based localization, it is important to mention a specific work that, although it utilised ultrasound, is very relevant in the context of distance ranging methods. Drumheller in [Drumheller, 1987] describes a multistage algorithm is utilised in which data from US sensors from all around the robot are used. From data obtained from an US scan, an attempt is made to extract line segments. Using interpretation trees, the lines scanned are then matched with actual walls (and null walls). Special techniques for refinement and ambiguity handling are also included. In a similar vein [Dudek and Paul MacKenzie 1993] provide another method by which the scan data is viewed as being essentially lines. An iterative matching scheme is then devised that attempts to fit the scanned lines to actual lines existing in the model.

[Arras and Tomatis, 1999] attempt to introduce a vision sensor, a CCD camera, to a robot already having a localisation system based on a LRF. The stand-alone LRF-equipped system achieves good performance in rooms in which the environment is made up of distinct features. The performance of the system undergoes a drastic reduction in efficacy when presented with long corridors and situations in which the laser beam is subject to specular reflection. With the aid of a vertical-edge extraction procedure the method seeks to present data that are more reliable in situations in which the LRF is prone to provide unpredictable or highly ambiguous data. Pose estimation in the problematic environments mentioned improved with the addition of the CCD camera.

[Vale et al., 2001] describe a robot navigation system for a structured environment consisting of corridors and crossings. Ranged measurements below a threshold value only, are utilised (to simplify data processing and to handle diffraction effects). The data is grouped and lines are fitted to the groups. The algorithm provides the possibility of using more than one line to characterize points in the same group. While the principal method of localisation is odometry, a filter is utilized that updates the odometry-provided estimate using laser range data. A lower level of control utilizes ranged measurements to correct the robot orientation and lateral position within the corridors, and a higher level of control is utilized to build the map, explore as yet un-revealed portions of the map, arrive at pre-programmed destinations and avoid obstacles (stops and retreats in the face of objects in its path). The environment is defined in terms of nodes and arcs. Nodes are represented with their position, the number of times traversed and the arcs leading from them.

[Pradalier and Sekhavat, 2002] present a method for localisation using a laser range finder and artificial landmarks. They call the method SMLAM for Simultaneous Matching, Localization and mapping since it involves a matching phase for localisation and a map-building phase. The distinguishing feature of the method is the creation, from a map of features already mapped (and subsequently updated, the ALM), two properties of the layout of the map features. Termed ‘invariant’ properties, the authors have constructed triangles from the landmarks taken three at a time. Another set of ‘invariant’ properties, utilized when the triangles cannot be used, consists of distance segments between pair of landmarks. The ‘invariants’ are held in two different databases, (RDB) s ordered by area (or length). From the laser scan, possible landmarks are extracted. These then become the observed landmarks (O) s. Invariant properties are then calculated for these same. A matching is attempted with the list of invariants in the RDBs depending on the number of

Os and their layout (position estimate is utilized). Triangles are matched using area and subsequently superimposition matching. By the application of matched landmarks to the ALM, the rotation and translation transform required to obtain the position estimate are obtained.

3. The Proposed Localisation Method

3.1. Introduction

The environment in which this work has been carried out consists of a hall containing a large number of objects (tables, laboratory-stools, cables, boxes, window-frames, etc) that appeared at the height of the laser scan, which means that the laser-scan is heavily laden with information. This information consists of points reflected off smaller features such as the legs of tables and chairs as well as other points reflected off larger planar surfaces. The proposed method for localisation in this work envisaged the utilisation of the presence of an artificial landmark placed somewhere in the environment at the height of the laser scan to perform localisation. This landmark would be effective for a certain set of pose variables, i.e. over a certain region of the environment and with the robot oriented through particular directions. Although only one landmark is used to localise at any given moment, more than one landmark might be set up in order to obtain robot localisation through a larger set of pose variables. While it seems a relatively easy task to use trilateration and geometric transformations (rotation and translation), after the identification of the landmark, to estimate the robot's position, it was not certain if a two–dimensional laser range scan to detect landmarks would work consistently and with a sufficiently high rate of successful detections. The extraction, with the aid of a signature of a particular landmark from among all this information, with a low margin of error, seemed not to be a trivial task.

Figure 3-1. A 360º-wide image of the LAR

It was decided to utilize '*composite*' landmarks, or landmarks made up of two or more individual and separable components. This was done for two main reasons:

1. To better control the detection of landmark's signature in the laser scan. In this way bounded regions, some containing points and others without points can be anticipated. The signature of a landmark would then consist of an organized pattern of regions some of which would include a certain number (or proportion) of ranged points while others would be noted for the absence of points.
2. To make the procedure less dependent on the distance of the robot from the landmark. It is known that due to the fixed angular resolution of the LRF, the points obtained on the surface of an object become further spaced out as the distance of the object from the LRF increases. This means that the information about objects ranged at a larger distance is less than that in the case of closer objects. Since an object identification approach (and not a data-map matching approach) is sought, a

trace of the surface of the object is not required. By using distinct separable components, that, as a whole, represent a landmark, checks can be made for the presence or absence of surfaces rather than the shape of the surface.

At this point, it might be mentioned that, in order to aid the task of landmark extraction, additional information such as the presence of natural (more permanent) features and open spaces and their distribution with respect to the objects inserted in the environment might be utilized. Attempts aimed at detecting and eliminating features such as walls, table-legs and others, allow for significant improvements in the performance of the algorithm. Since the absolute position of the landmarks is known beforehand to the application, it will then be able to calculate the position of the robot and its orientation using simple algebraic expressions. In addition, the same program might be utilized to pick out one among many landmarks, making use of known variations in geometry to accomplish this task.

This project involves two main aspects:

1. Choosing a good template for the landmark configuration so that the pattern detected by the LRF is as unique as possible (i.e. the intersection of the laser beam with the other features in the environment should not result in erroneous identification of the landmark).
2. Designing an algorithm that, though simple to implement, is effective in the recognition of the landmarks. While it is possible to implement complex algorithms, the aim was to design and utilize a set of simple algorithms that might be easily run on the onboard computer or on the Robuter processor itself. The challenge is to discard points that are thought not to belong to the landmarks, while avoiding the discarding of those that might actually represent the landmarks.

Since the robot has no estimate of the position it occupies at the start of the mission, the algorithm must work without any previous position estimate. Thus, the signature of the landmarks must be very distinct if a positive identification of the landmark is to be made with a high confidence.

3.2. Setting up the landmarks

Multiple vertical poles (hereafter referred to, interchangeably, as markers or poles) are arranged along a straight line with a constant lateral separation. Sets of three cylindrical plastic pipes were utilized in the exercise, though the program can easily accommodate a greater number or poles (a smaller number of poles would not work as there would be no pattern to confirm in the alignment of the poles). A greater number of poles might be thought to be desirable for reasons of greater reliability in the identification phase

However, a greater number of poles presents other problems, mainly as a result of a larger landmark (which reduces the region over which the landmark is visible in its entirety and without suffering occlusion.

In Figure 3-2 a set of three poles is shown arranged against a wall, the region around the poles and between the poles and wall being free from other objects. The ‘wall’, in this case, is actually a glazed surface (covered by a curtain to prevent the beam from passing) that is largely free of furniture and other objects. The pipes are common, commercial-grade PVC drainage pipes, with the external diameter measuring ca. 13 cm. The space between the poles, in the above case, was fixed at 0.5 m, giving us a total landmark width of ca. 1m. The parameters such as the distance of the landmark from the wall, the length of the wall, the open spaces in front of and behind the landmark can be configured and adjusted for other landmark configurations and environments.

Figure 3-2. An example of a typical landmark set-up.

The broad ideas behind setting up a landmark are as follows

1. Poles are set up along a straight line that is either parallel or perpendicular to a major surface
2. The poles must be uniformly spaced out. Additionally, the distance from any one of the extreme poles to the nearest object, other than an adjacent pole, must be superior to the sum of the pole spacing and the pole width.
3. Any landmark will be effective over a region of the work environment. The choice of where the poles are located and their orientation should reflect not only the general orientation and position usually taken by the robot but also the path taken by the robot.

By varying the factor variables mentioned above such as the orientation of the poles, the spacing and their position in the hall, the algorithm shall distinguish between different landmarks. The algorithm must be able to distinguish a landmark from among background information and from among other landmarks. The signature of the landmarks as they appear in the ranged data must not only be distinct from one another, they must be distinguishable from other spurious landmarks. The task of identifying the poles from among the other data in the laser scan is left to the algorithm described in the next section.

3.3. Processing the laser range data

Whenever localisation is required, a procedure is run to fetch a data string from the LRF, process the laser range scan (as shown in Figure 3-3) and attempt a positive recognition of the previously defined landmarks in the environment.

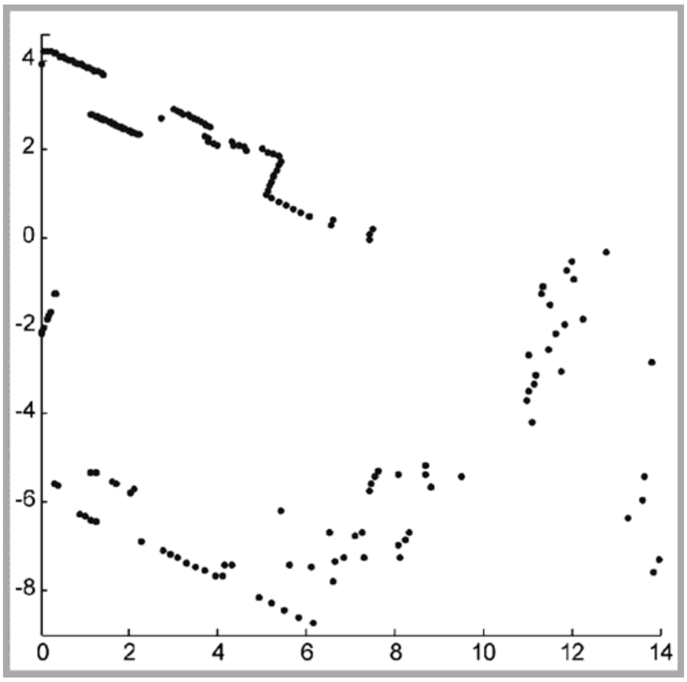


Figure 3-3. Example of a raw laser scan (180º wide with 1º of resolution)

In all the laser scans the rotating mirror of the LRF is located at [0,0], Cartesian coordinates. The X axis is defined by the intersection of the horizontal plane at which the range is obtained and the vertical plane of symmetry of the range finder. The Y-axis is defined by applying the right-hand rule to the scan as viewed from above the LRF.

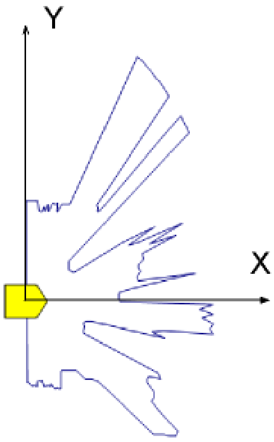


Figure 3-4. Radial plot of laser range data

The range measurements in radial (r,θ) coordinates are transformed into Cartesian (x,y) coordinates using:

$$x_{laser\_data} = r_{laser\_data} \times \cos(\theta_{laser\_data})$$
$$y_{laser\_data} = r_{laser\_data} \times \sin(\theta_{laser\_data})$$

The Laser Range Finder (LRF) was set up to obtain scans with angular range of 180º and a resolution of 1º. System error is put at a typical ± 4 cm (for a distance of less than 20 m) and the standard deviation is typically 5 mm. The LRF is set up to detect the emitted laser beams that are reflected by features in its path; the distance to the reflected surface is obtained through 'time of flight' (TOF) calculations. In some cases, the beam passes through windows and open doors giving large range values, the limiting case being the value corresponding to a 'time-out' of the LRF. The LRF is set to standard sensitivity, which is adequate for the indoor conditions expected.

The data processing consists of subjecting the ranged points to agglomerative clustering procedures and a series of heuristic eliminatory tests. The purpose of these tests is to recognize and extract points that represent the landmarks and the other features. Successive tests serve to steadily eliminate walls, other large surfaces, table legs, cable etc and finally the landmark. The agglomerative clustering procedures group the points obtained on the same feature and also features that are related to their neighbours.

3.4. The principle of the data-processing algorithm

The algorithm has passed through at least two evolutionary stages. In a first version, a procedure in which range discontinuities aided the identification of salient features was utilized. The following procedures were undertaken:

1. A check for discontinuities in the laser scan data was performed. A central differences procedure was utilized on the radial data as a measure of the point to point-to-point discontinuity for each ranged point. Only those points that were sufficiently separated from their neighbours (after partially accounting for the distance of the robot from the surface) were subsequently processed.
2. These were then subjected to checks for the pole-width and pole-separation criteria.

Results showed, however, that too many groups of points passed the test resulting in a rather large number of false detections together with the correct detections. The differences procedure that was employed to make the points stand out was found to work unsatisfactorily in some cases as the data points on a pole were sometimes discarded (this situation often occurred whenever the robot was quite close to the poles). Another problem was the setting-up of the algorithm for the identification of multiple landmarks, a task that proved to be computationally very time-consuming.

To get around the shortcomings of the earlier approach a new procedure was implemented. Here,

1. Points are grouped based on their nearness to others. The *nearest neighbour* agglomerative clustering technique is used with a particular value of the distance as

the threshold. Clustering stops when the distance between the nearest neighbours is greater than this threshold. The resulting groups are then represented by their area centroid. Thus, the points lying on the same pole are now represented by a single new point.

2. The separation between these groups of points is tested against a pole separation parameter. In other words, the distance between the representative centroids are tested. The occurrence of points in this space would signify that the centroids in question do not represent the poles.

Once again, however, it was seen that too many groups of points passed the test. The actual procedure is a modification of the two-step approach mentioned above. However, before proceeding with the search for landmarks, the algorithm adds a data processing phase to eliminate points in the LRF scan that *cannot* possibly belong to the landmarks. The remainder points are now grouped and attempts are made to identify the poles from among the groups. In addition, as explained further, a final confirmatory step is performed in which checks are carried out for the presence of permanent features and unoccupied spaces at some position defined relative to the landmark

The stages of the algorithm are as follows:

1. Data Preparation and Pre-processing
   1. Grouping of points.
   2. Elimination of large groups and limiting number of members in a group.
2. Beacon Extraction.
3. Landmark confirmation.

At this point, it might be useful to formalize the schematic of the set-up of the coordinate system and the nomenclature that shall be used in the description of the algorithm.

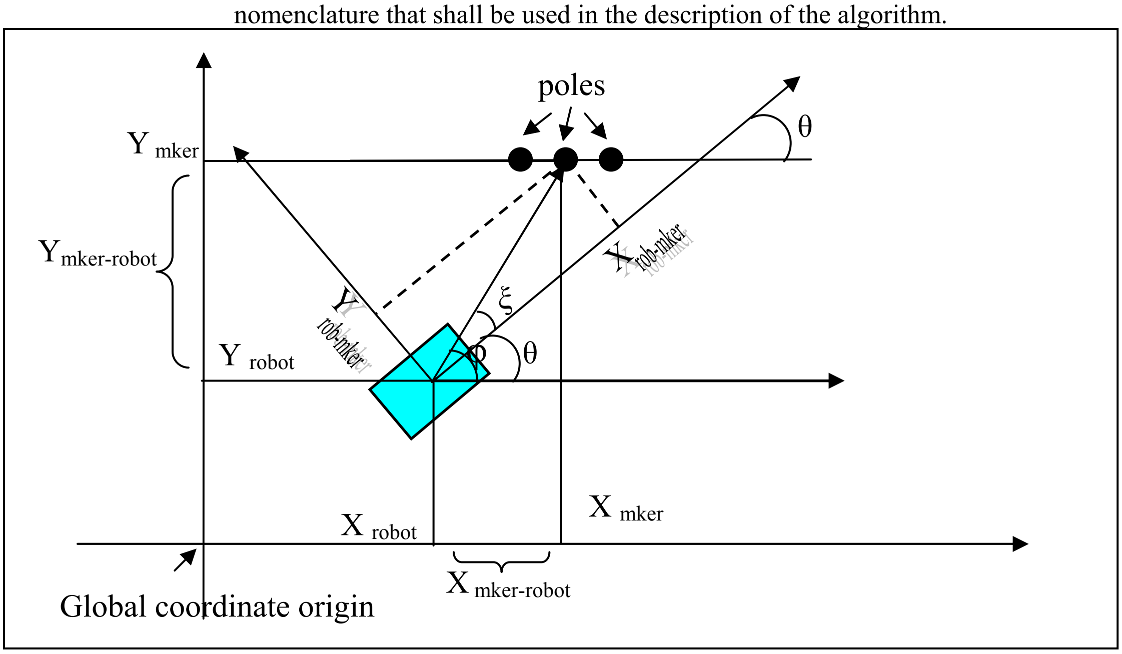


Figure 3-5. Coordinate system set-up

All the references, except at the time of calculating the robot position (X_{robot}, Y_{robot}), in global coordinates, made to the ranged points, group centroids and beacons are in terms of this robot coordinate system.

Point: A *point* refers to an x and y Cartesian coordinate pair in the robot coordinate system.

Group: A *group* refers to a collection of *points* that are related to each other by the fact of their proximity to one another. *Groups* are formed at the Data preparation and pre-processing stage by an agglomerative clustering procedure. All *points* in the same *group* have the same *group number*. *Groups* are represented in space by the centroid of the points within the group, $[\bar{x},\bar{y}] = \left[\sum_{i=1 \ldots n} x_i \big/ n \; , \sum_{i=1 \ldots n} y_i \big/ n \right]$, where n represents the number of points in the *group*.

Clusters: In the beacon extraction stage, A second agglomerative clustering is carried out , using the nearest neighbour method on the *group* centroids It is sought to include all the beacons into the same *cluster* and thus ease the computation, especially the fitting of a straight line to the centroids potentially representing the beacons.

Point and Group Removal: The algorithm is fed with the (x, y) coordinate pairs obtained from the LRF scan. At the outset, any of the 180 points is treated in the same fashion and has the same probability of affecting the identification of the landmark. Through the successive algorithm stages, *points* are eliminated from the body of points that will provide the position of the beacons. The term ***removed*** has been utilized to refer to this elimination. The points that are *removed* might, however, still be utilized in the landmark confirmation stage.

In the following sections, the algorithm details are explained. The illustrations utilized are taken with the robot position at a particular position in the hall with a view as seen in Figure 3-6. The scan in Figure 3-3 was obtained from this position.

Figure 3-6. A panoramic view of the environment from the robot position

3.5. Data preparation and pre-processing

After obtaining the string containing the 180 measurements from the Robuter processor, the data are treated to aid the actual search procedure that follows.

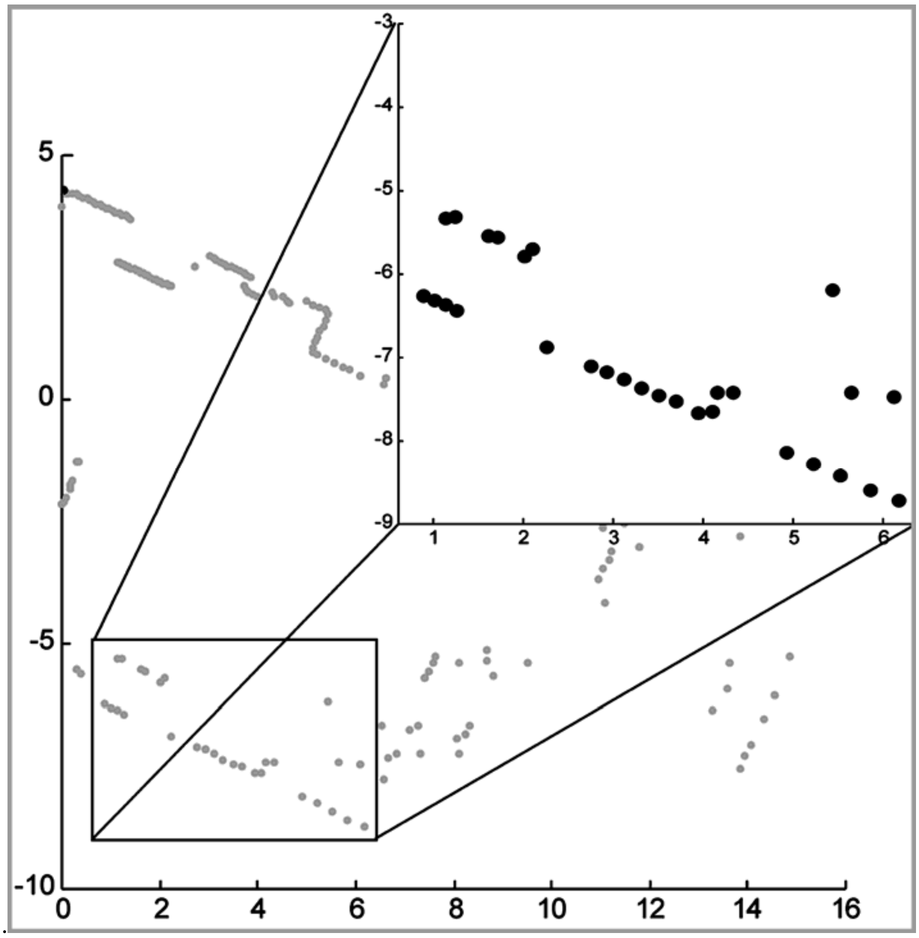


Figure 3-7. Raw laser scan with box showing sub-region referred to in text below

The search algorithm for the poles was found to work much faster when the points are subject to clustering. This comes from recognising that points lying close to one another might belong to the same feature, or belong to features that are related in some way. This fact, besides providing information as to the origin of the ranged point also means that all the points in the same group can be treated in the same way.

In this section are described the parameters used to set up the algorithm. To aid the text, a specimen laser scan (Figure 3-7) is presented, wherein a sub-region is examined in detail. In brief, the steps undertaken in this phase might be summarised thus, following the initial clustering, an elimination criterion was utilized in order to remove from consideration groups that spanned across an area wider than a specified value. This second step allowed a narrowing down of the number of groups within which the beacons might possibly lie, resulting in an increase in algorithm efficacy and a reduction in computational time

3.5.1. Grouping of points

Great advantage was obtained in the aggregation of the data into groups and their subsequent submission, in grouped form, to subsequent processing stages. Through the nearest neighbour (also called the single-linkage technique) hierarchical clustering technique points separated by a distance inferior to a defined value are included in the same group. A threshold value that attempts to groups points on the same feature together (but keeps apart the points belonging to separate landmarks) is used. The value of this parameter, the *Landmark contiguous spacing,* depends on a number of factors as shall be explained next.

Landmark contiguous spacing (LCS)

The classic way in which the nearest neighbour technique is executed is by the calculation of the inter-point distance matrix by using a distance measure (Euclidean or otherwise) for every pair of points [Everitt, 1993]. The smallest value in this matrix is then utilised to agglomerate the two groups which contain the pair of points involved. Here the matrix is recalculated (one row and one column), taking into consideration the constituents of the new group. Agglomeration stops when only one group remains, or, when some criterion related to the number of groups or the number of points within the groups or the least value in the distances matrix is applied.

$$D = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ \vdots \\ 180 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & \ldots & 180 \end{array} \\ \begin{bmatrix} 0 & & & & & \\ a & 0 & & & & \\ b & c & 0 & & & \\ d & e & f & 0 & & \\ g & h & i & j & \cdots & \\ k & l & m & n & \cdots & 0 \end{bmatrix} \end{array}$$

where, for example the element D(3,2), b, is calculated as the Euclidean distance $\sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}$ between (x_3, y_3) and (x_2, y_2). This matrix is symmetric.

Since the end criterion for the clustering is known to be the distance between the nearest neighbour, a modification of the above procedure is employed. The *LCS* is the threshold distance to limit the growth of an individual cluster.

The laser scan is swept one point at a time from beginning to the end. Initially, each individual point is taken to represent a distinct *group*. An iterative process that joins groups is then run. Beginning with the first *group*, the distance of all the points constituting the *group* from all the points belonging to other groups is calculated. Each of these distances is then tested against the LCS, and agglomeration takes place wherever the distance is inferior to the LCS. The procedure is repeated for the same *group* till no new agglomeration results. Attention is then shifted to the next *group*. Each *group* is represented by the area centroid, i.e. the mean value of the x and y coordinates, of the points that comprise the *group*. The result of the procedure is demonstrated graphically in Figure 3-8.

In Figure 3-8 the circles, drawn around the points, have a radius that represents **half** the *LCS*. An intersection of two circles thus indicates that the two points lie within a distance inferior, or at most equal, to the *LCS*.

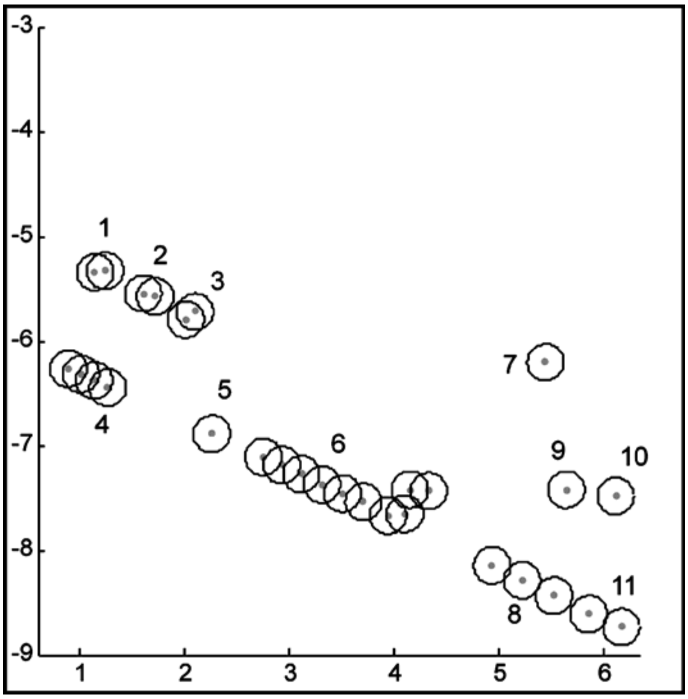


Figure 3-8. Illustration of the group creation procedure using LCS.

The procedure in the form of pseudo-code is provided below.

```
for i = 1 to no_of_points_to_be_grouped
        no_of_grps = +1
        do
                calculate distance of points in grp to points not in group
                if(distance < LCS), then agglomerate
        while (agglomeration successful)
end
```

This parameter must be decided keeping two things in mind:

1. Width of the poles: Intuitively the width of the poles should correspond to the minimum value of the *LCS*. This allows for an aggregation of all the range points arising out of a single pole into a single *group*. In practice, a value larger than the pole width is utilized in order to account for angular and linear uncertainties of the LRF and other effects that result in some dispersion of points.
2. Distance of the closest object to a landmark-pole (including another landmark-pole): An attempt is made to include the points obtained on an individual pole in a single *group*, separate from the points obtained from the incidence of the laser beam on other surfaces. If the *LCS* is greater than the distance that separates a point on a pole from any other surface (be this another pole, some nearby object or even a spurious data point), then the points on the pole will be included in the group representing the other surface or vice-versa. Either way, the ability to isolate the points representing the pole and the feature(s) shall be lost.

The *LCS* can be varied from a very small value that would result in no aggregation of individual points into groups to a large value that would agglomeration all the points into a single, very large, group. The number of groups obtained as a result of a particular value of the *LCS* usually depends on the distance of the opaque surfaces from the robot.
The *LCS* should be as large as possible as a larger value gives rise to a smaller number of *groups*, each containing, on average, a greater number of points. This results in faster processing at the later stages.

3.5.2. Elimination of large groups

After grouping, an initial 'segmentation' of the data is performed in order to extract and remove *groups* that might belong to large surfaces such as walls, large boxes, cupboards etc. The points that are eliminated are not examined further for consideration as part of a set of possible poles but will still be used in the Landmark confirmation stage.

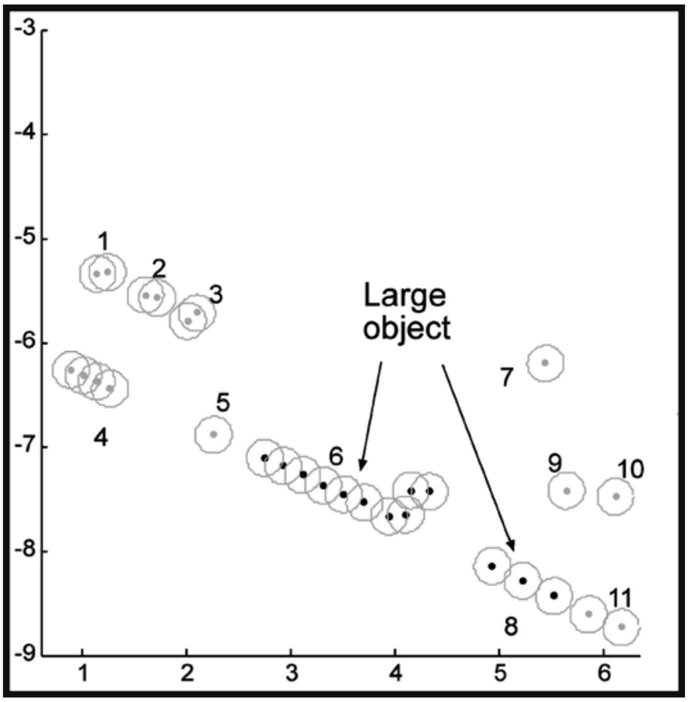


Figure 3-9. Elimination of large groups.

Group spread (GS)

The previous section described the grouping of points that serve, essentially, to identify and isolate, into distinct *groups*, those points lying on the same surface. A number of different features are to be found all around the laboratory. Walls are a especially troublesome type of feature since points ranged by the LRF on a wall satisfy the linearity criteria required for points lying on the poles and may sometimes satisfy the separation and dispersion (*groups* separated by regions not containing *groups*) criteria specified for the landmark. It is desirable to eliminate walls and other similar, often large features.

Large, linear features can, however, be identified and then eliminated by means of a simple test. For this, the extent of the feature is verified and then all the features (groups) that are longer (larger in a particular direction) than a particular length are eliminated. In this process, other features such as clutches of cable, points ranged on curved surfaces, etc might also be eliminated.
The distance separating points in the same group furthest away from each other is calculated for each group and compared with the *GS*. This scheme is acceptable though it was originally meant for the elimination of linear features. As mentioned earlier, a number of large (spread over space) non-linear groups are also eliminated in the process. It can be seen in Figure 3-9 that groups six and eight are large enough to be removed. The size of the groups, besides depending upon the features scanned, also depend upon the value of the *LCS* utilized.
The *GS* parameter affects the algorithm by separation and elimination of walls and other larger linear features and by reducing the number of points in the scan that are passed on to subsequent tests. If the value of the *GS* is too high then all or most of the points will be considered and passed on to subsequent test increasing the computing load greatly. This value must, therefore, be tuned in order to keep the proportion of points passed on to the subsequent tests to a minimum while maintaining in consideration the groups representing the poles.

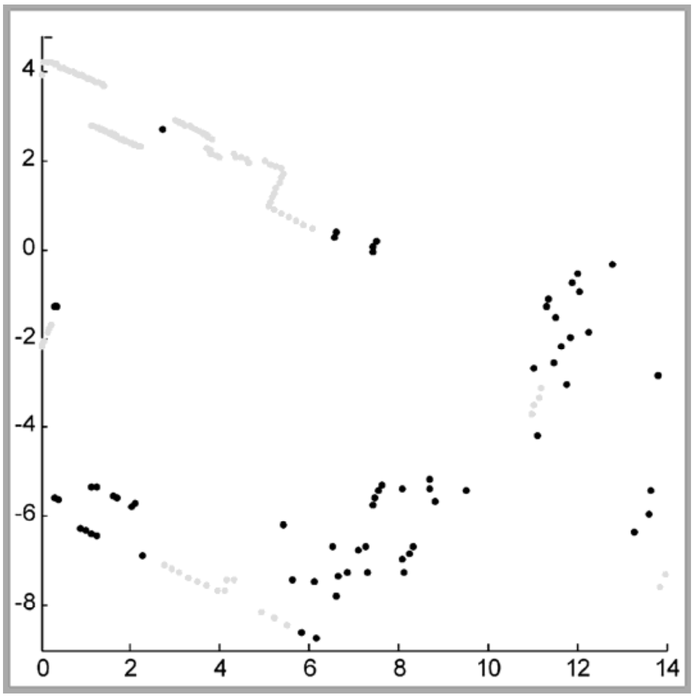


Figure 3-10. Elimination of large features (darker points remain in consideration)

The procedure utilised to obtain the extreme points of a *group* is reproduced as follows in the following pseudo-code

```
for i = 1 to number_of_groups
    Px = find (min_value, index of min_value) of x coordinates of points in group i
```

```
        Py = find (min_value, index of min_value) of y coordinates of points in group i

        Qx = find (max_value, index of max_value) of x coordinates of points in group i
        Qy = find (max_value, index of max_value) of y coordinates of points in group i

        if (max_x – min_x > max_y – min_y), then
                span_of_group = Euclidean dist bet Px and Qx
        else
                span_of_group = Euclidean dist bet Py and Qy
        end

        reject groups if span_of_group > GS
    end
```

The lower limit for the *GS* is the size of the pole. If the *GS* is set to less than the width of the pole, the groups representing the individual poles might get eliminated. By eliminating features that are clearly larger than the poles, a large number of points can be removed as seen in Figure 3-10.

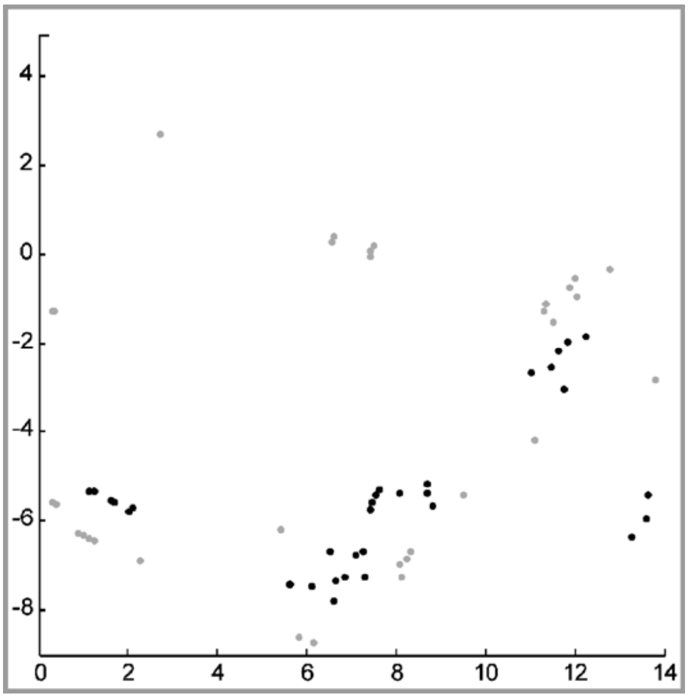


Figure 3-11. Isolated groups eliminated (dark Points remain in consideration)

Besides the above two procedures that are carried out in pre-processing, an additional procedure has also been included in order to increase computational efficiency. Here groups in relative isolation are removed. These include the odd cable, furniture legs, persons moving around the hall and other isolated instances. The occasional odd *group* obtained by specular reflection was also eliminated in this step. These *groups* are removed by verifying that each of them has no neighbours closer that the distance given by the sum of the pole separation and the pole width. This criterion is provided to prevent the elimination of *groups* that represent the poles. In pseudo-code:

```
for i = 1 …no_of_centroids
     for j = 1 …no_of_centroids, j ≠i
               if Euclidean distance(Ci, Cj) < pole_separation + pole_width,
then score(i) = +1
     end
     if score(i) <= 1 eliminate centroid i
end
```

The result, at the end of this procedure is seen in Figure 3-11.
At the end of this stage, from the original data there only remain *groups* that represent features whose dimensions are approximately the size of the markers.

3.6. Beacon extraction

At this stage, a reduced number of *groups* of points exist, some of which might contain some of the three markers. As mentioned earlier, these *groups* are represented by their centroids.

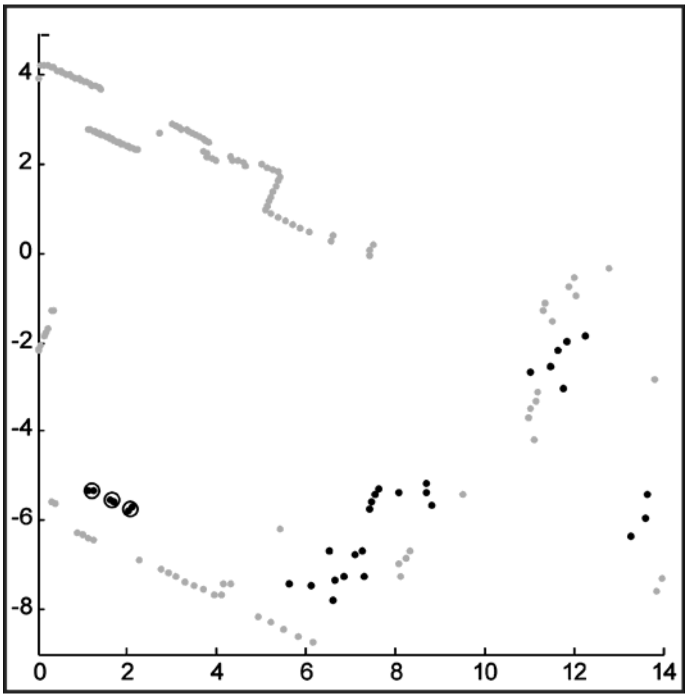


Figure 3-12. Beacon extraction (polygons indicate estimated centres of the poles)

The earlier tests were designed to minimize the chances of elimination of groups that might possibly hold the markers, but there is still a chance they might have been inadvertently discarded. Such a situation is not remediable. In addition, if the scan were taken in a random fashion, there would be a high probability that the complete landmark is not observed at all, and, hence, does not even appear in the scan. If, however, the markers do lie in one of the groups, this step is entrusted with the task of extracting them.

A new clustering is carried to bunch together the existing groups. A procedure similar to the one used in the Data preparation and pre-processing stage is carried out, the coordinates centroids of the *groups* being used to calculate the Euclidean distance between the *groups*.

For each new cluster formed, combinations of centroids taken three (the number of beacons) at a time are then analysed. A linear regression analysis is run to test the co-linearity of the centroids, resulting in a total of $\sum_{i=1\ldots m} {}^{n_i}C_3$ (for m clusters and n_i centroids in cluster i) different linear regression calculations. In order to reduce the number of calculations, clusters containing 5 or more centroids were rejected. The regress(Y, X) function in Matlab and the corresponding OLS(Y,X) function in Octave were utilised. Combinations that are sufficiently collinear, defined through a minimum value of the coefficient of regression (a value of 0.8 was utilized in the experiments) are then tested for the linear separation between the poles.

The distance of separation between the three poles is one of the landmark properties provided in the problem statement. The distances that separate each of the two extreme centroids from the middle one are compared to the inter-pole separation. After experimenting with different values a tolerance of 10% was introduced in order to account for errors in range measurements and imprecise set-up of the poles. If the two instances of linear separation are satisfied, a set of three *groups* that might possible represent the three poles is obtained. After having searched the entire scan for such combinations of points, these are then passed on to the confirmation stage (more than one successful combination can be passed on for verification).

3.7. Landmark confirmation

The Beacon extraction stage sometimes passes more than one combination of points, either because of a false identification or because more than one valid landmark is visible.

In this stage therefore, other properties, concerning the layout of the landmarks that were not included during the search for the beacons, are introduced. More specifically, information regarding some possible wall or other natural, more permanent, feature with a known location relative to the landmark is utilized. Information regarding some open space (clear of features) somewhere in the vicinity of the landmark might also be added. These additional properties are defined in terms of the existence of and absence of, respectively, scanned points in demarcated rectangular regions around the landmarks. The potential trio(s) of centroids is (are) tested against information about the features.

If the rectangle within which ranged points are expected is defined by the 2 pairs of parallel lines $a_1x + b_1y + c = 0$, $a_1x + b_1y + d = 0$ and $a_2x + b_2y + e = 0$, $a_2x + b_2y + f = 0$. A voting scheme as explained in the pseudo-code below is set up:

```
for i = 1 ….l80 (all the ranged points)
```

if($\min(c,d) \le a_1 x_i + b_1 y_i \le \max(c,d)$ **and** $\min(e,f) \le a_1 x_i + b_1 y_i \le \max(e,f)$ **), then**

vote = +1

end

if vote > minimum_vote, then confirm landmark

In the case of open spaces the **vote** should be less than a threshold defined for such a situation.

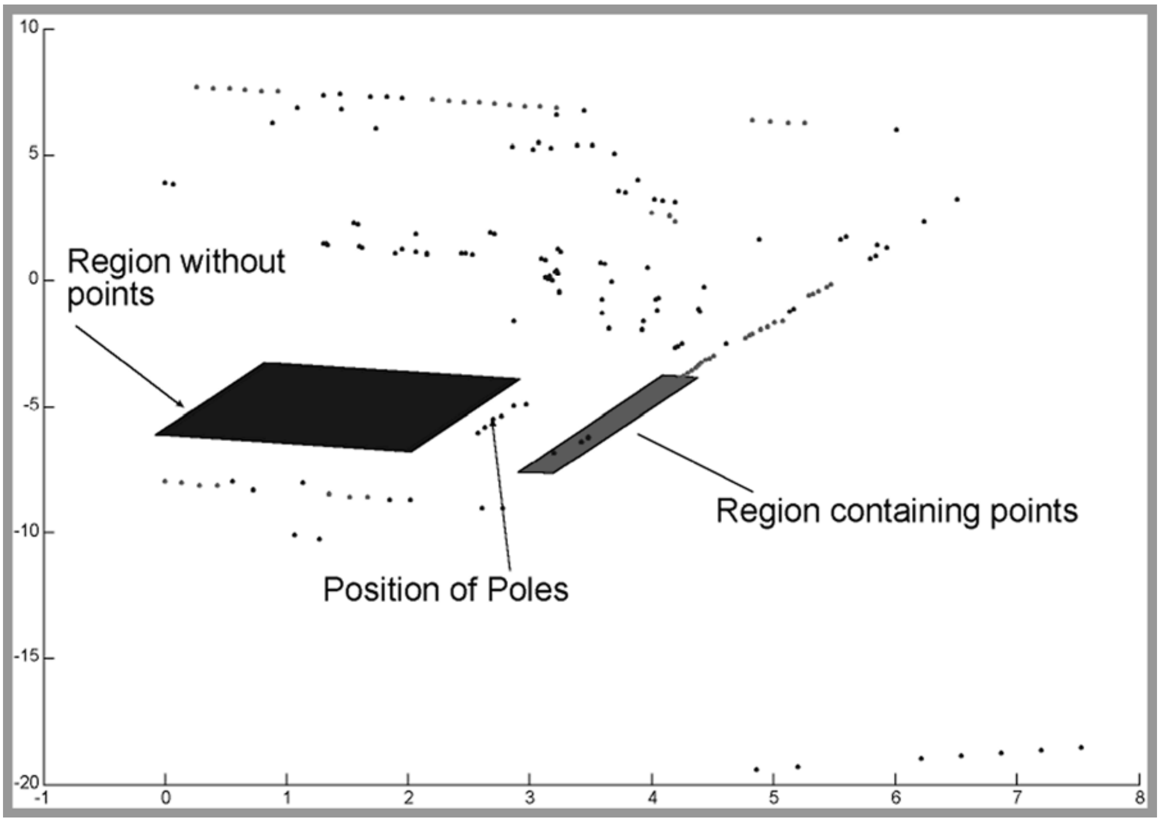


Figure 3-13. Bounded regions with and without points

For example, in the scenario presented in Figure 3-2, if the landmark is related to the wall that lies behind poles, and the open area in front of the poles. The original scan data is analysed for the presence of points lying in a thin strip that represents the glazed surface behind the markers and a large rectangular area in-front of the poles. While the presence of the solid feature is necessary, the inclusion of the space devoid of features is optional.

This confirmatory step is carried out for all the sets of points passed on by the earlier step, The last set of points, that pass the confirmatory step are utilised to calculate the robot position, i.e., at present, there is no way to handle more than one combination of points.

3.8. Calculating robot position

Upon the successful identification of the landmark, the program retrieves the absolute position of this particular landmark and works out the rotation and translation transformations required to match the landmark actual position to the position occupied by it in the environment. This allows us to calculate the robot pose (X, Y, θ) in global coordinates.

Referring to Figure 3-5, we can come up with the transformations (rotation and translation) required to convert the scan (and the corresponding origin which describes the robot position) into global coordinates.

Suppose the poles are arranged parallel to the X axis of the Global coordinate system and in the first quadrant, as illustrated in

If $X_{rob-m\,\mathrm{ker}}$ and $Y_{rob-m\,\mathrm{ker}}$ represent the coordinates of the landmark in the robot's frame of reference, and $X_{m\,\mathrm{ker}-robot}$ and $Y_{m\,\mathrm{ker}-robot}$ are the projections of the distance between the landmark and the robot in the global frame of reference (specified in the program), thus:

$$\mathrm{X}_{\mathrm{mker-robot}} = \left(\sqrt{\mathrm{X}_{\mathrm{rob-mker}}{}^{2} + \mathrm{Y}_{\mathrm{rob-mker}}{}^{2}}\right) \times \cos(\theta + \xi)$$

$$\mathrm{X}_{\mathrm{mker-robot}} = \left(\sqrt{\mathrm{X}_{\mathrm{rob-mker}}{}^{2} + \mathrm{Y}_{\mathrm{rob-mker}}{}^{2}}\right) \times \sin(\theta + \xi)$$

where, $\theta = \mathrm{atan}\left(Y_{rob-m\,\mathrm{ker}} \big/ X_{rob-m\,\mathrm{ker}} \right)$ and ξ = the angle that the x-axis of the robot makes with the line containing the poles[1]. The sum of θ and ξ gives φ.

Now,

$$\mathrm{X}_{\mathrm{mker}} = \mathrm{X}_{\mathrm{robot}} + \mathrm{X}_{\mathrm{mker-robot}}$$

$$\mathrm{Y}_{\mathrm{mker}} = \mathrm{Y}_{\mathrm{robot}} + \mathrm{Y}_{\mathrm{mker-robot}}$$

Giving us:

$$X_{robot} = X_{m\,\mathrm{ker}} - X_{m\,\mathrm{ker}-robot}$$

$Y_{robot} = Y_{m\,\mathrm{ker}} - Y_{m\,\mathrm{ker}-robot}$, $\mathrm{X}_{\mathrm{mker}}$ and $\mathrm{Y}_{\mathrm{mker}}$ being given parameters.

To obtain the position of the scan points in global coordinates, they shall have to be rotated using the matrices for rotation:

[1] If the line makes an angle other that zero with the X axis of the global coordinate system, this angle must be added in order to obtain ξ. This angle is provided in the landmark parameters and is also utilised in order to demarcate the regions within which to check for confirmatory landmarks

$$Rotation_{scan} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and translation:

$$Translation_{scan} = \begin{bmatrix} 0 & 0 & X_{m\ker} - X_{m\ker-robot} \\ 0 & 0 & Y_{m\ker} - Y_{m\ker-robot} \\ 0 & 0 & 1 \end{bmatrix}$$

Combining the two, the general transformation matrix is obtained

$$Transformation_{scan} = \begin{bmatrix} \cos\varphi & -\sin\varphi & X_{m\ker} - X_{m\ker-robot} \\ \sin\varphi & \cos\varphi & Y_{m\ker} - Y_{m\ker-robot} \\ 0 & 0 & 1 \end{bmatrix}$$

If the scan is represented by the matrix Original_Scan[] $_{3xn}$ the final position of the points is given by the matrix Transformed_Scan[] $_{3xn}$ where:

$$Transformed\_Scan[\ ]_{3\times n} = Transformation_{scan} \times Original\_Scan[\ ]_{3\times n}$$

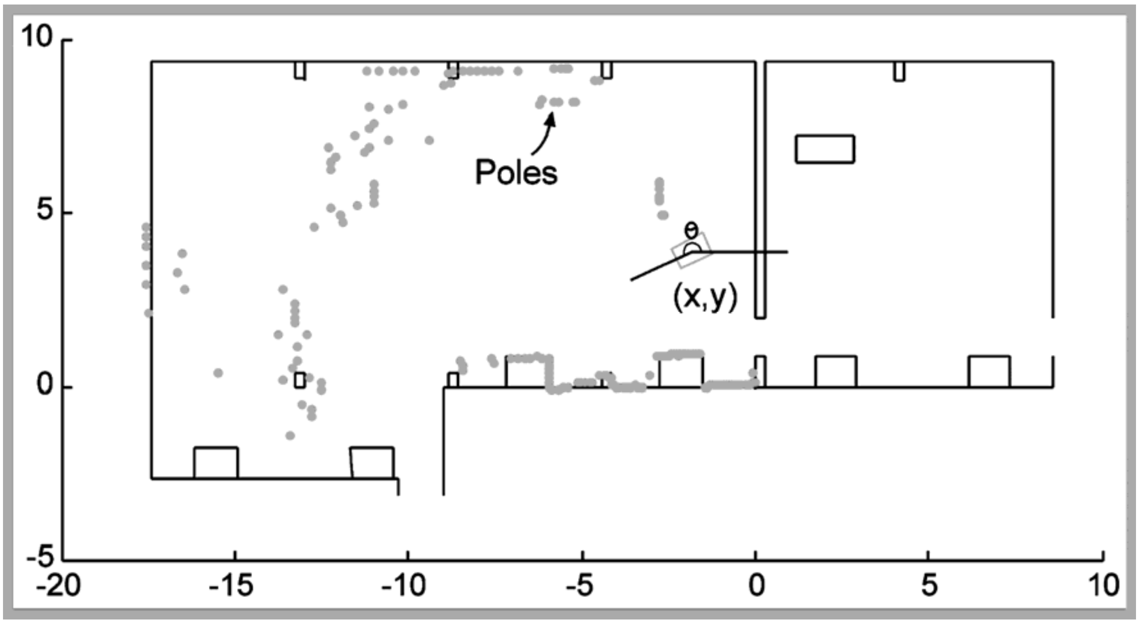


Figure 3-14. Superimposition of rotated and translated scan upon diagram of laboratory.

For visualization purpose, the laser scan has been superimposed on an outline drawing of the extents of the environment (bereft of furniture) in Figure 3-14. The Robot pose as calculated by using trilateration and linear transformations has been presented. As can be noted, the scan after transformation does not exactly superimpose the line drawing of the laboratory, there being a translation and rotation error.

At the end of the data pre-processing, the beacon search and the landmark confirmation stages two outcomes are possible, depending on the results obtained from the Landmark confirmation stage, either the application succeeded in isolating the landmarks or failed. Failure might arise either because the LRF was not able to scan the complete landmark (poles and confirming surface) or because the laser scan did not successfully represent the landmark. Occlusion, far-away beacons, incorrect parameters and spurious points are the usual reasons for failure of localisation. Chapter 4 presents the results of a system of four landmarks that was implemented so that localisation could be carried out through a wider range of directions and over a larger area. Such a system might be implemented in part or as a whole so that localisation might be more effective and occur over a shorter time.
The flowchart of the algorithm described in this chapter has been described in Figure 3-15. The description of the landmarks and their properties are provided by the user.

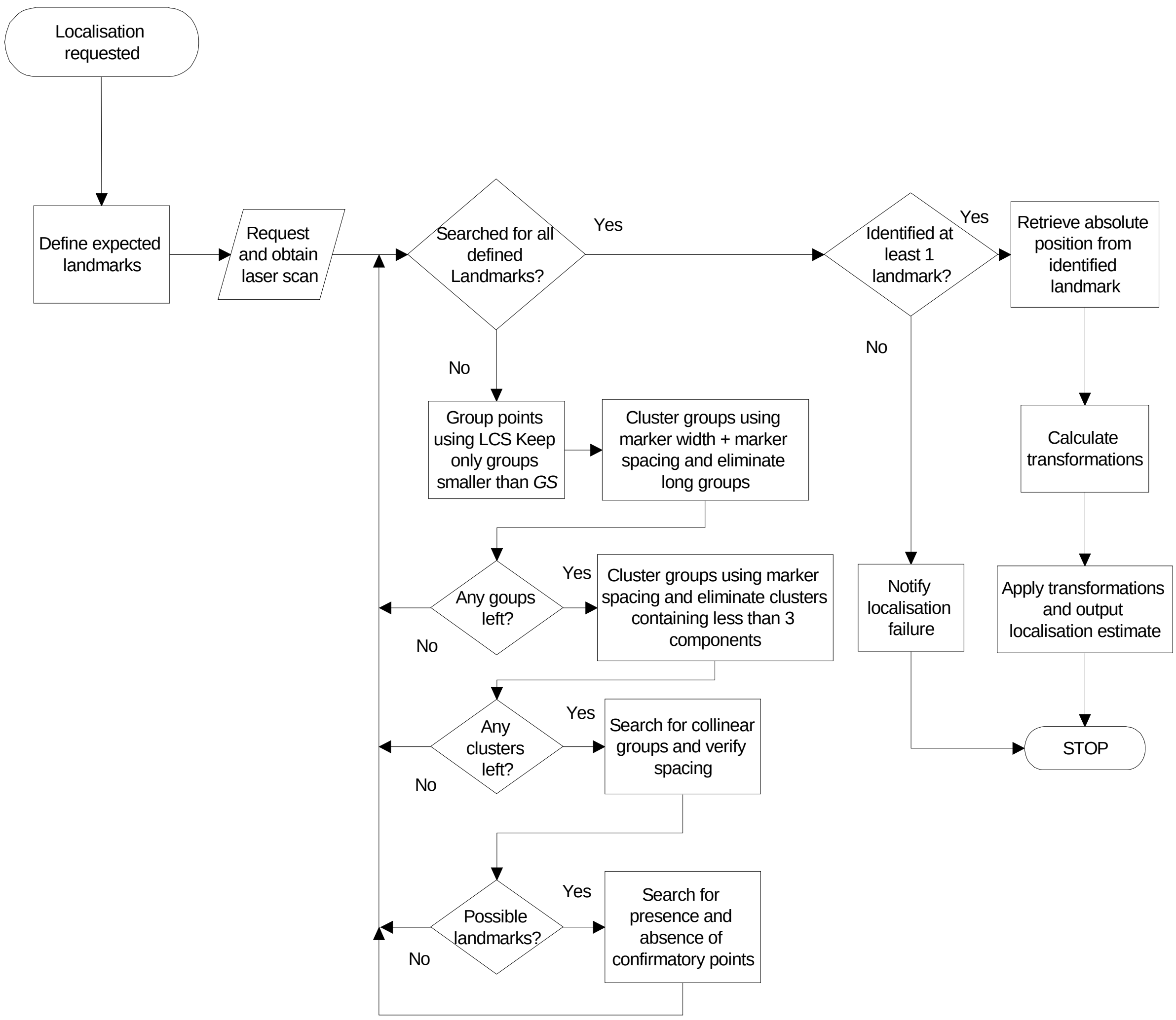


Figure 3-15. Algorithm flowchart

4. Experimental Results and Performance Testing

The efficacy of the algorithm in obtaining a location estimate is a measure of the algorithm's performance. Performance should be evaluated especially at the time of mission start-up and at the time of final positioning at the gantry robot.

Upon the completion of a mission, the robot is usually parked somewhere within the LAR (referred to as the common parking region in Figure 1-6). This position might be manually altered upon robot shutdown. Consequently, the robot position within the parking space in the hall is not known a priori. As mentioned in the introductory chapter, the principal reason behind the development of this work was to enable the robot to acquire an initial position estimate at the start-up of the robot's mission. Thus, the ability of the algorithm to pick out the landmark from laser range scans taken from several distinct places shall be evaluated. The ability of the algorithm to obtain a localisation estimate in the presence of multiple landmarks shall be tested. The repeatability of the algorithm is also a property of interest, i.e. whether the algorithm is able to consistently detect the landmark when different scans, taken from the same spot.

The ability of the robot to obtain a good localisation estimate (an estimate with a low error) at the site of the gantry robot (Figure 1-4) must also be tested. Therefore, the error obtained while calculating the position estimate at close range shall be evaluated. An estimate of the error that might be obtained in various situations around the gantry robot shall be obtained.

The following were the tests carried out:

1. Attempts at localisation at various points in the common parking region given the presence of a single landmark.
2. A test to obtain some measure of the positioning error of the algorithm at a given position in the start-up region (comparing the algorithm estimate with another obtained using a measuring tape).
3. Attempts to obtain a localization estimate at various points in the start-up region with four different landmarks set up. The ability of the algorithm to discern between different landmarks and a demonstration of the landmark layouts that can be used is tested here.
4. Tests to obtain a measure of the positioning error at a few different positions near the gantry robot.

4.1. Test 1- Localisation through the extent of the start-up region

The robot was placed at various points in the hall and scans were made after a cursory check to verify that the poles did indeed lie within the 180º sweep of the laser scan and that they were not occluded by other features. A set of three 12.6 cm external diameter poles each separated by a distance of 50cm from the other was erected. In a total of 161 readings the algorithm detected 132 instances correctly, corresponding to a success rate of 81%. The remainder 19% of non-detections were attributed to various causes. The most frequent reason was thought to be due to spurious points that appear between the poles, or points that influenced the orientation of the estimated straight line. These ‘spurious’ points could not be eliminated by the screening tests and originally appeared with greater frequency due to the averaging of ranged data and problems related to the angular resolution of the LRF. A modification of the laser set-up parameters allowed for a reduction in the number of unsuccessful detections. Other non-detections occurred due to partial or total occlusion of the landmark. Importantly, the algorithm made no erroneous detection in all 161 scans.

Trials	Correct Landmark Detections	False landmark detections
161	132 (82%)	0

Table 4-1. Test 1 - Summary of results.

The actual execution time varied, depending upon number of potential landmark sets in the tests. Also, the total time of execution was substantially longer because of the time required to obtain the data string from the LRF.

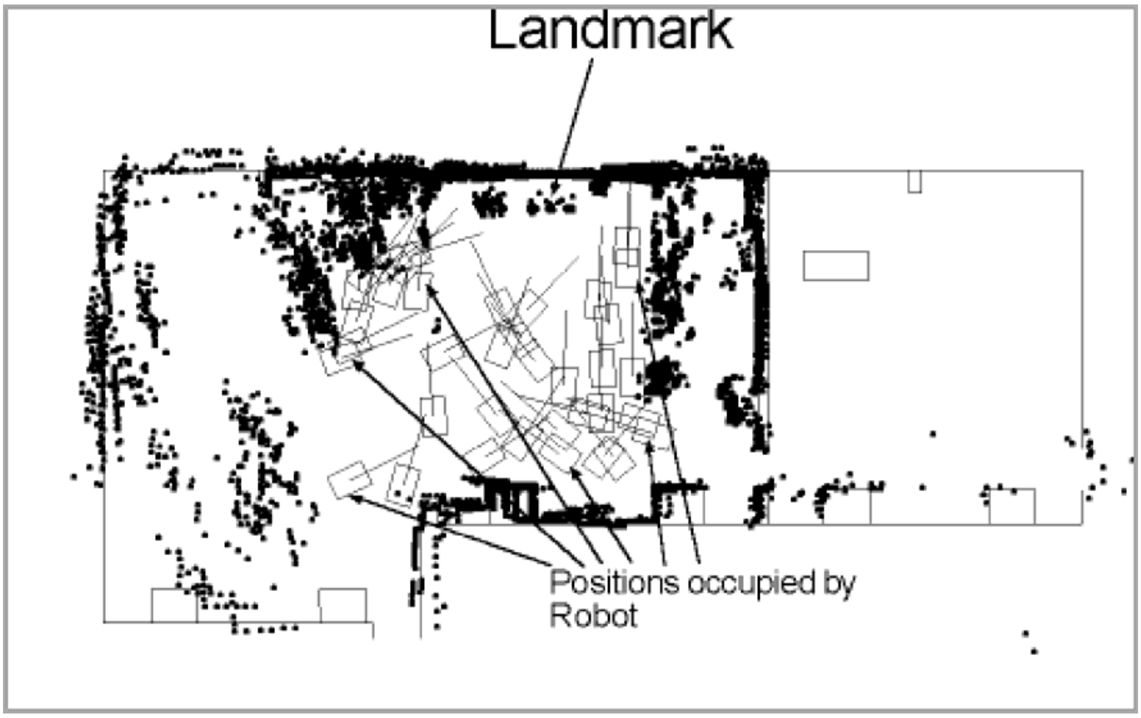


Figure 4-1. Test 1 - Superimposed robot positions and scans.

4.2. Test 2-Repeatability of the estimate.

Tests were conducted for two different pole-spacings, each using two different pole sets. This allows for a comparison of the effect that the pole spacing has upon the error of the position estimate. The effect that total pole array length has upon the error of the estimate may be evaluated.

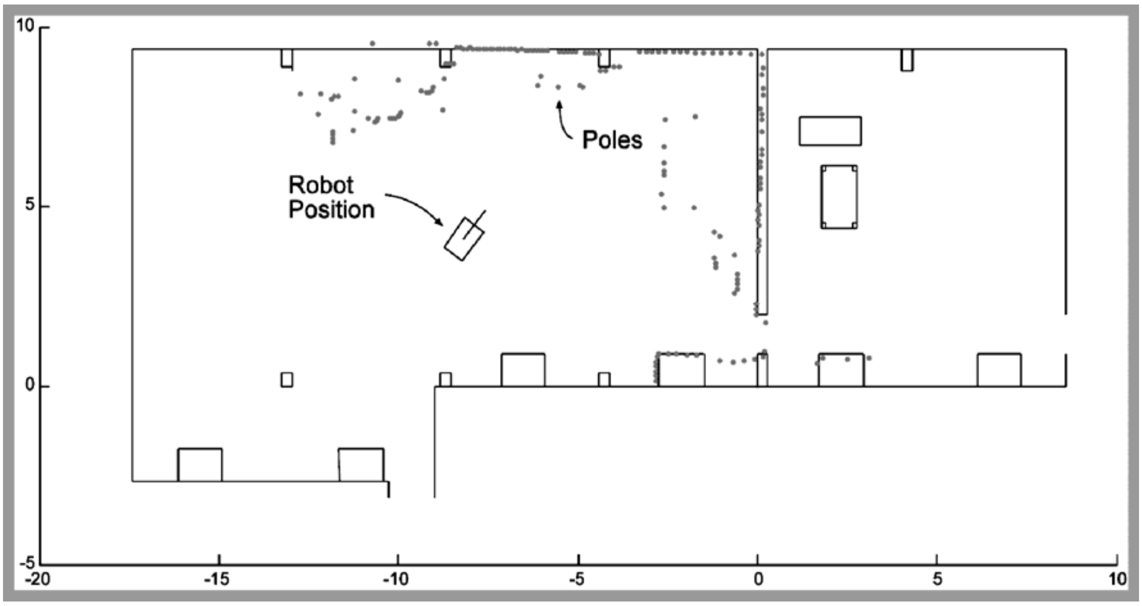


Figure 4-2. Test 2 - Localisation in the LAR (single scan).

With the help of a measuring tape, the robot right rear wheel was positioned at (–7.95, 3.57) and the left wheel at (–8.49, 3.90). The LRF mirror is situated approximately at the position (–8.11, 3.90) and the X axis of the robot makes an angle of around 58º with the absolute X axis of the environment. The results have been tabulated in Table 4-2. As can be seen, the fourth case shows an abnormally high standard deviation, due to a shift in the robot position during the execution of the test.

Configuration	No. of trials	Mean X Meters	Mean Y Meters	Mean θ Degrees	Std Dev X Meters	Std Dev Y Meters	Std Dev θ Degrees
7.5cm poles 25 cm apart	50	-8.34	4.12	52	0.092	0.027	1.24
7.5cm poles 60 cm apart	50	-8.15	4.05	54	0.022	0.013	0.29
12.6 cm poles 50 cm apart	50	-8.13	4.02	55	0.028	0.017	0.36
12.6 cm poles 60 cm apart[1]	50	-8.09	4.06	54	0.176	0.139	2.07

Table 4-2. Test 2 - Summary of results of position estimates

4.3. Test 3-Localisation using multiple landmarks.

Four different, independent landmarks were set up using the same procedure for setting up landmarks defined in chapter 3. Software parameters were accordingly added in the program. While the program searches for the existence of each of the landmarks, only the last identification is utilised to obtain an estimate of the position of the robot.

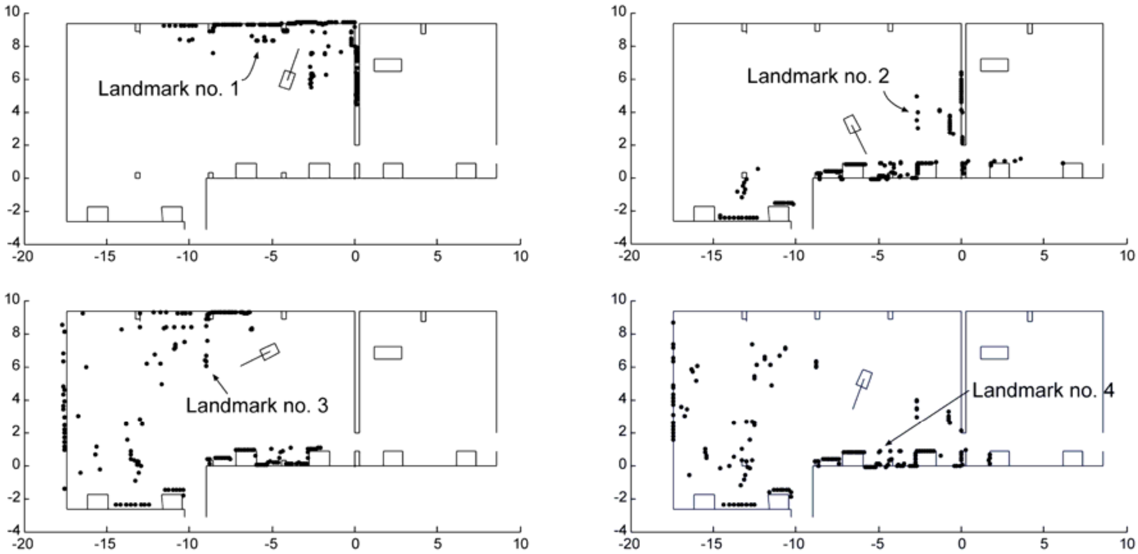


Figure 4-3. Test 3 - Position estimation using four different landmarks

Using these four landmarks, the parking space of the robot is almost completely covered, allowing a position estimate to be obtained anywhere within this region and in almost all directions.

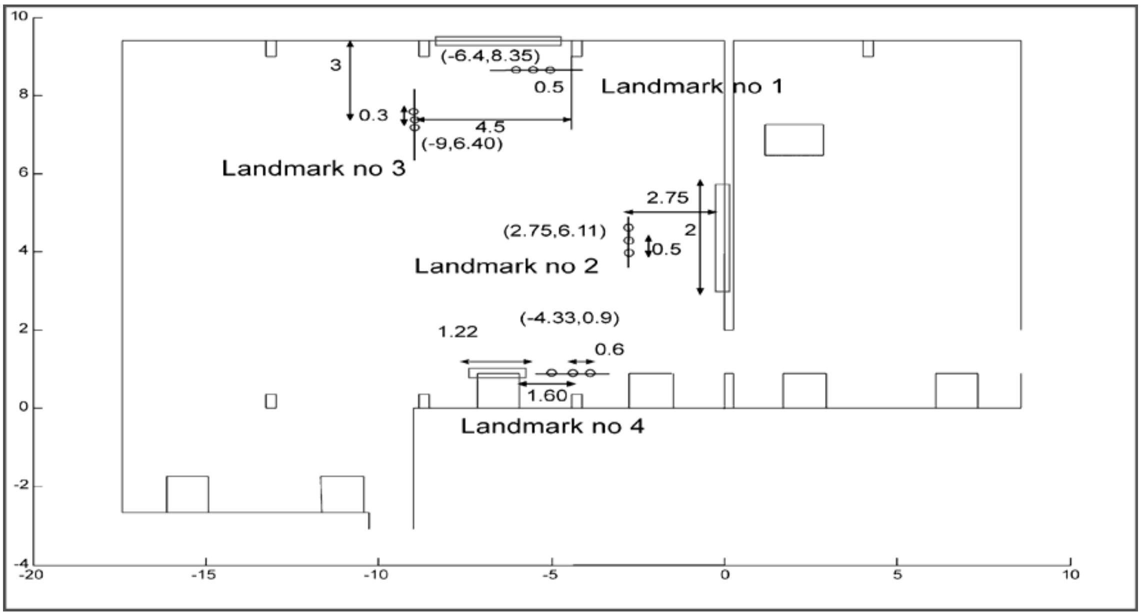


Figure 4-4. Test 3 - Schematic of four landmarks set up in the LAR

The results obtained using the 4 landmarks are seen in Figure 4-5. The localisation was performed using one identification, though in some cases, as can be seen, more than one landmark was completely visible in the same scan.

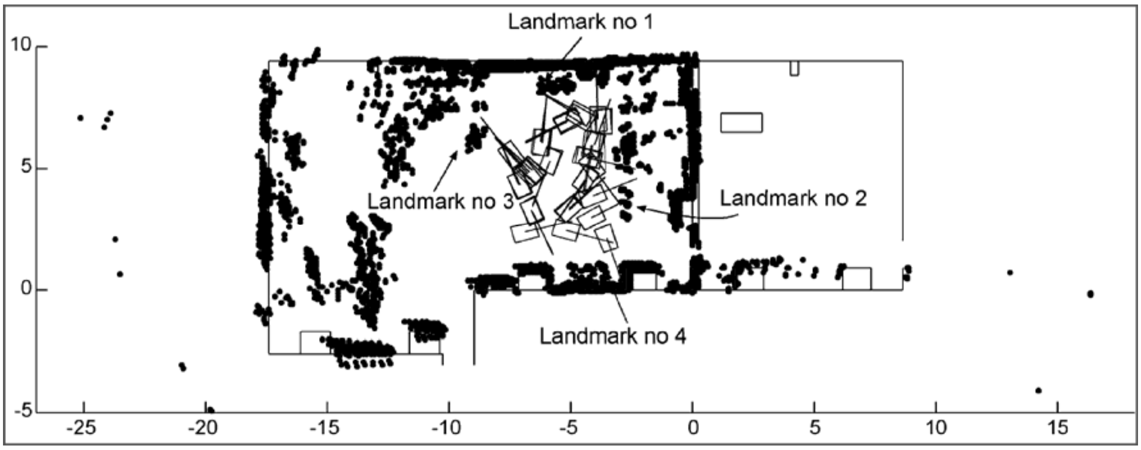


Figure 4-5. Test 3 - Localisation utilizing one of four different landmarks.

As can be seen, most of the region where the robot is usually kept to recharge batteries after the end of a mission is covered successfully. This, in terms of X and Y coordinates together with orientation.

Of the twenty-five scans taken, at least one localisation estimate was obtained in twenty cases

4.4. Test 4- Measuring localisation accuracy for near landmark

The LRF was utilised to obtain range data from the region around the gantry robot where a set of three beacons, of external diameter of 0.075 cm were erected, separated by equal distance of 35 cm, with the central pole at (3.03, 4.85) and at a known distance from the conveyor belt, one side of which acts as the confirmatory surface (Figure 1-7). A quick check was undertaken to include the strips in the field of view of the LRF. The set-up is shown in Figure 4-6.

Localisation was carried out from three different positions. In one such position the algorithm failed to obtain a location estimate. In the case of the other two instances the results are as shown in the Table 4-3.

Distance of LRF from Central Pole	No. of trials	Mean X Meters	Mean Y Meters	Mean θ Degrees	Std Dev X Meters	Std Dev Y Meters	Std Dev θ Degrees
1.45 m	50	1.59	4.98	-5	0.004	0.011	0.45
1 m	50	2.08	5.12	16	0.005	0.006	0.38

Table 4-3. Test 4 - Summary of results

Once more, no false detections were obtained.

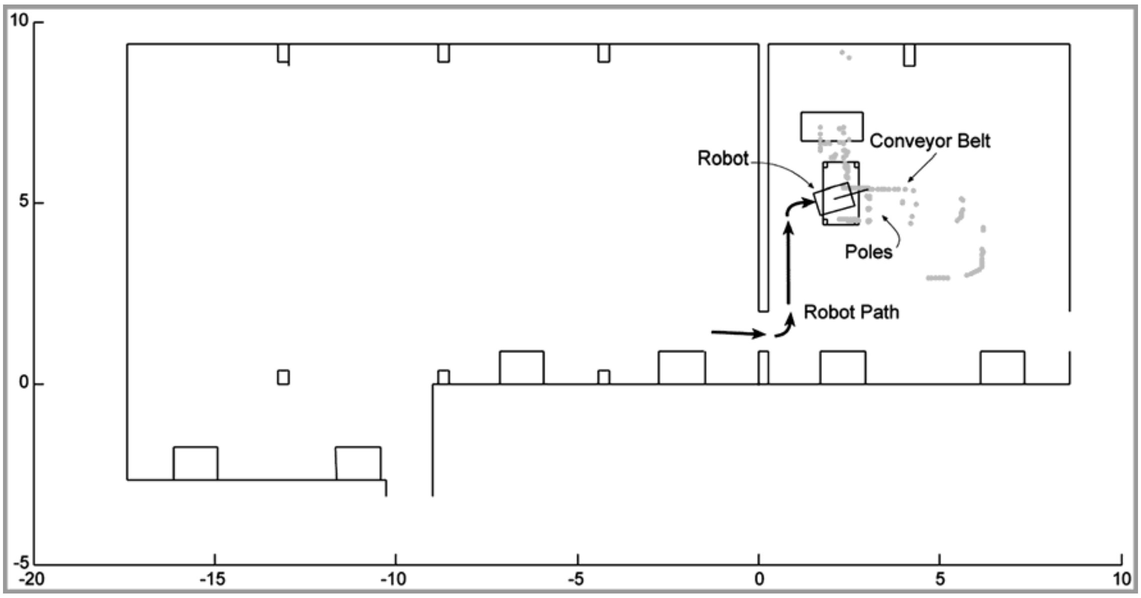


Figure 4-6. Test 4 - Set-up for localisation at the CIM

4.5. Observations

The following are some findings that came to light during development and testing. The effect that the inter-pole spacing has on the position estimate, the occurrence of incorrect identifications, the effect of presenting multiple landmarks and the extent of the region over which the localisation algorithm works have been discussed here.

4.5.1. Effect of varying distance between the markers

The algorithm was tested for a number of situations in which the distance between the poles was varied from a lower limit of (pole width x 2) to 0.75 metres (for a set of 12.6 cm diameter poles). The algorithm worked successfully in all cases after the necessary modification of the parameters. The success rate obtained in some layouts was inferior to others and though this might simply be an effect of sampling, it might have to do with geometrical errors in the layout of the landmark.

Another effect that was noted was the variation of the angular error with the variation in the distance between the poles. The standard deviation of the angular estimate is inversely proportional to the distance between the extreme poles. This appears to be because the errors introduced in the scanning process seem to be non-correlated to the actual distance being ranged. This means that while the spread of the ranged values for each pole (the source of the angular and position errors) was the same, a smaller angle was subtended as the distance between the poles was increased.

4.5.2. False detections

The program proved to be adequately resilient to false detections. The combination of the checks for pole width, the open spaces between and on either end of the poles combined with the data preparation steps of grouping and elimination of large features allow for the practical non-existence of false detections. This has an implication in that *the choice of landmark and pole configuration need not depend on the environment that is being ranged*. Throughout the testing phase one false detection was encountered. This came about due to a combination of circumstances in which the distance of a surface from the robot and the layout of surrounding surfaces construed to create a pattern similar to that created by a bonafide landmark.

In this regard, it must be mentioned that the algorithm is designed for a relatively imprecise set-up of the poles. Quite a large tolerance is provided in the set-up of the distance between the poles, in their inclination (due to imperfect horizontal surface or pole base) and in their orientation. The rate of detection is however affected. For example, in one case, with poles of 7.6 cm diameter the middle pole had sufficient inclination to result in the co linearity of the poles being affected resulting in non-detection of the landmark.

4.5.3. Presence of spurious points

There is a continued presence of spurious points in the laser scan obtained from the LRF. These are found to be especially troublesome when they occur around the poles where they upset the aggregation of points into groups representing the poles. These are consistently found behind the poles and in the blank spaces between the poles thus forming a "bridge" between the poles themselves or between the poles and other nearby surfaces. They also upset the separation criteria specified.

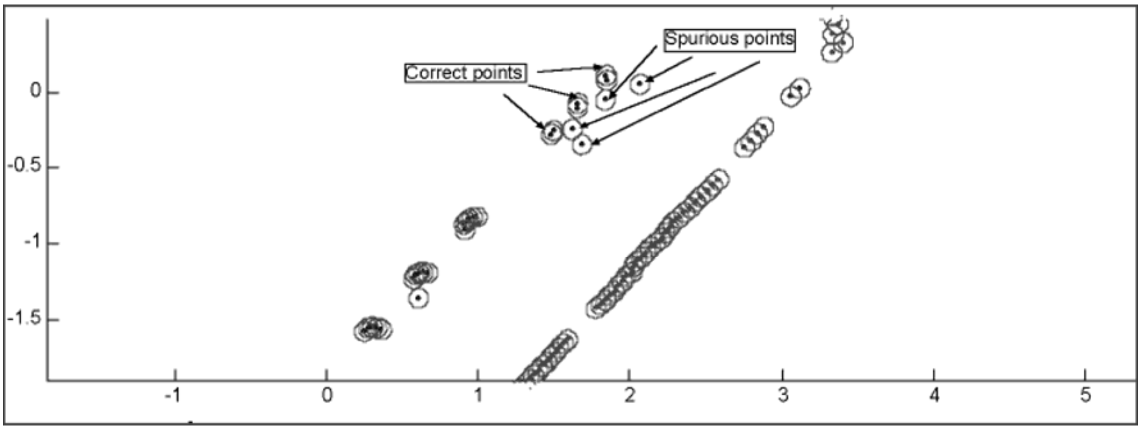


Figure 4-7. The problem of spurious range points.

Their presence (or absence) does not seem to be correlated to the distance of the robot from the poles. This was verified by taking multiple readings while the robot stayed at the same place. The presence of spurious points meant that the robot failed to extract (group, correctly match against specified configuration) the poles and failed to localise.

These spurious points seem to be related to certain conditions of multiple reflections that exist on some surfaces. At the beginning of the present work the problem was more serious as the LRF had been set up for multiple evaluation (the LRF is set to two evaluations-perscan by default), which resulted in large occurrences of ‘spurious points’. The LRF settings were later modified to remove multiple evaluations.

4.5.4. Presentation of multiple landmarks to the same algorithm

The same program was utilised to search for more than one landmark. Here the same data was present repetitively to the program, which applied tests with different parameters to the same data set. In Test 3, the program attempts to identify one of four landmarks, which, through a choice of parameters are sufficiently different from each other (see chapter 5 for setting up the landmark)

While the introduction of provisions for the recognition of more landmarks in the program is feasible (as long as no two landmarks are too similar), the way the algorithm is written at present means that the computation time increase is of O(n), where n is the number of landmarks the program can identify.

4.5.5. Effect of the distance between the robot and the landmark

Also, when more than one point is obtained on each pole an averaging effect is obtained resulting in an estimate with a smaller error. Since the number of points ranged on a pole depends on the distance between the LRF and the pole, position estimates become less accurate as the LRF is moved away from the poles. However, since beyond a certain distance the LRF is not able to range all the poles there is an upper limit on the error.

In a similar way, the confirmation stage requires that there be a sufficient density of points ranged on the surface chosen for the confirmation of the landmark. At certain positions, because the distance of the robot to the surface is large and/or because the point of view of the robot is such that too few points are obtained on the surface chosen for confirmation, localisation fails to occur.

5. Setting the Algorithm Parameters

As described in Chapter 3, the algorithm can be thought to be made up of 3 steps; the first a pre-processing stage followed by a beacon search procedure and a landmark confirmation stage. This Chapter explains the various parameters that must be set up in the program and their significance. It first deals with two parameters that must be set up for the pre-processing of the scan and then with the remaining parameters that are necessary for the beacon search, landmark confirmation and robot position calculation procedures. All the distance parameters are expressed in meters.

5.1. Pre-processing of range points

For the pre-processing stage, the software requires that the following parameters be set:

1. The Landmark Contiguous Spacing (LCS)
2. The Group Spread (GS)

5.1.1. Landmark contiguous spacing

Tests were performed on landmark configurations in which the same poles were arranged in different configurations with varying inter pole distance. Good localisation results were obtained for a range of *LCS* values. The value of the *LCS* is linked to the pole width. It could be varied from a value slightly less than the width of the pole to values somewhat greater than the width of the pole. For the poles with a diameter of 0.126 m, values ranging of *LCS* from 0.1 m to 0.25 m were found to be effective.

Note: the value of the *LCS* that gave the best results varied, but this seems to be partially explained by the variance in the sample range scans and due to the presence of spurious points.

5.1.2. Group spread

The effective value of the *GS* was linked to the value of the *LCS* and the nature of the scan (whether the robot is exceptionally close to walls and other features). The value of the *GS* should be as small as possible as this means that fewer points qualify for subsequent processing and computational time is saved. The values for which good results were obtained depend on the value of *LCS*, the actual spacing between landmarks and the distance of the robot from some large feature or laboratory extreme. For a pole diameter of 0.126 m and LCS values in the range 0.1 to 0.25 m, good performance was obtained for *GS* values between 0.15 m to 0.4 m

It should be noted the *LCS* and the *GS* must be considered together. While values of 0.15 m and 0.2 m respectively worked for poles of width 12.6cm, better results were obtained

with parameters of 0.25 m and 0.4 m (in this case the arrangement of markers allowed for these values).
The reason because the choice of the two must be made together is because a larger *LCS* means that the total number of different groups gets reduced and that the average size of the groups increases, leading to larger groups and point eliminations for the same *GS*, altogether resulting in a fewer number of points for processing.
This relation between parameter values and efficacy is modified/altered by specific pole arrangements and by the presence of spurious points that serve to bridge the gap between the poles (and other features) and increase the observed pole width.
As a rule of thumb, the *LCS* is fixed at a value slightly greater than the pole width while the *GS* takes values 30 to 60 percent higher.

5.2. Beacon search

The actual search procedure checks for clusters that are sufficiently collinear and, then verifies if the groups are separated by the necessary spacing.
The parameters that must be set are

1. The pole width
2. The inter-pole separation

5.2.1. Pole width

This parameter reflects the physical width of the poles that comprise the landmark. Its value should be equal to or slightly greater than the diameter of the poles used.

5.2.2. Inter-Pole separation

This parameter is a reflection of the physical distance separating the poles. By this it is meant the distance between the centres of the cross-section of two adjacent poles. The program utilises the value of this parameter, together with the pole width specified earlier to account for the fact that range points might be obtained anywhere on the surface of the poles.
It is this parameter together with the earlier one that serves to distinguish one landmark from another. While the parameters in the pre-processing stage help to reduce the probability of false detection (beside their primary effect of improving computation efficiency), if the Pole width and Inter-Pole separation are not sufficiently different, erroneous detections will result.

5.3. Landmark confirmation

The verification phase in which other information that might be known about the range points in the neighbourhood of the landmark are checked against the results of beacon search. Namely, the verification of the presence and the absence of ranged points in some regions described with respect to the poles is carried out.

The parameters that must me set in the program are

1. Resolution of robot position ambiguity relative to landmark
2. Rectangular area containing range points
3. Rectangular area not containing points

5.3.1. Resolution of ambiguity in the position of the robot relative to landmark

Since the feature that is utilised is a linear one there always exist two symmetrical points between which the algorithm will be unable to distinguish, in the absence of additional information. This has been resolved by constraining the construction of the landmarks. The layout allowed for the landmark specifies that the robot can occupy positions only on one side of the line that defines the row of poles. This region that can be occupied is defined in terms of the global coordinate system. Four configurations are possible in all depending on the orientation of the line of beacons and region occupied by the robot. They are illustrated in the Figure 5-1. The value of the parameter to be used for each of the cases is as follows a) 0, b) –90, c) 180 and d) 90.

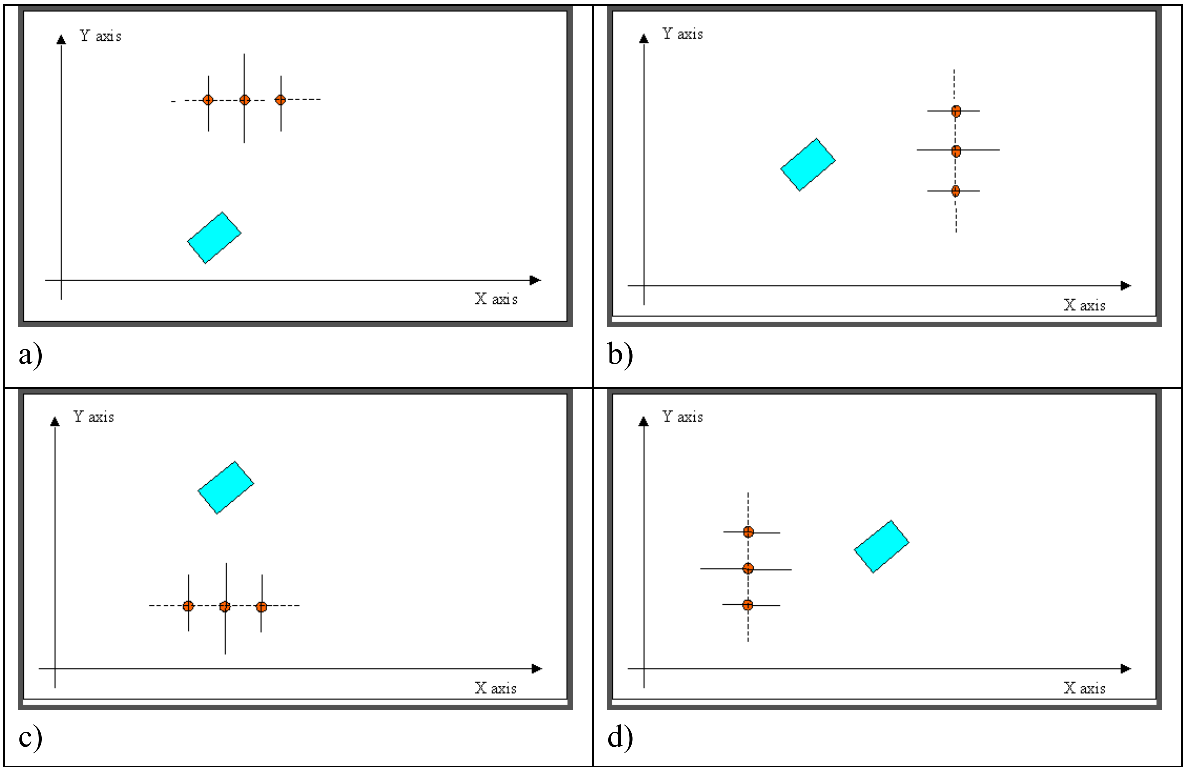


Figure 5-1. Resolution of robot position ambiguity relative to landmark.

5.3.2. Rectangular area containing range points

As mentioned earlier, the search process includes two stages; that of separating out potential beacons followed by the verification stage. The following two parameters deal respectively with the specification of regions in the hall that contain points and those regions that are not supposed to contain points.
This parameter (consisting of 4 specifications) specifies the completely bounded region that is supposed to contain ranged points that represent a certain permanent feature. Since rectangular regions were chosen to bind these regions, 2 pairs of parallel lines that are perpendicular to each other were chosen. The specification is made with respect to the position of the central marker in the global coordinate system.
Each pair is specified in terms of the distance from the central pole together with a 'tolerance' that specifies the distance that separate the pair of lines. The position of the second line is given by the value of this tolerance (sign of the tolerance value x sign of the distance of the first line from the central pole), added to the position of the first line. One pair is parallel to the row of poles while the other is perpendicular to the same row.

5.3.3. Rectangular area not containing points

The specification of the boundaries of this region is performed in the same way as in the case of the *"Rectangular area containing range points"* since, again, the boundaries of this region are rectangular.

5.4. Calculating robot position

For the program to calculate the transformations matrices to perform the necessary rotation and translation the absolute position of the beacons in terms of the global coordinate system must be known. The algorithm requires that the coordinates of the central pole be provided.

5.4.1. Absolute coordinates of the middle pole

In the above experiments, an edge of the Automation and Mobile Robotics laboratory (LAR) was taken to represent the origin of the 2-dimensional global coordinate system. The axes were taken to lie parallel to the walls of the same laboratory. The parameter specified here is the position of the centre of the central pole in terms of the global coordinate system.

5.5. Description of the parameters in graphic form

The following is a brief summary of the parameters that must be set.

1. Spacing between the poles
2. Width of the poles
3. X-coordinate of the central pole.

4. Y-coordinate of the central pole.
5. Parameter used to group belonging to the same object in the scan (LCS).
6. Parameter used to eliminate objects larger than the poles (GS).
7. Defines one limit of the region that must contain points (line parallel to line of poles).
8. Defines other limit of the region that must contain points (line parallel to line of poles).
9. Defines one limit of the region that must contain points (line perpendicular to line of poles).
10. Defines other limit of the region that must contain points (line perpendicular to line of poles).
11. Defines one limit of the region that must not contain points (line parallel to line of poles).
12. Defines other limit of the region that must not contain points (line parallel to line of poles).

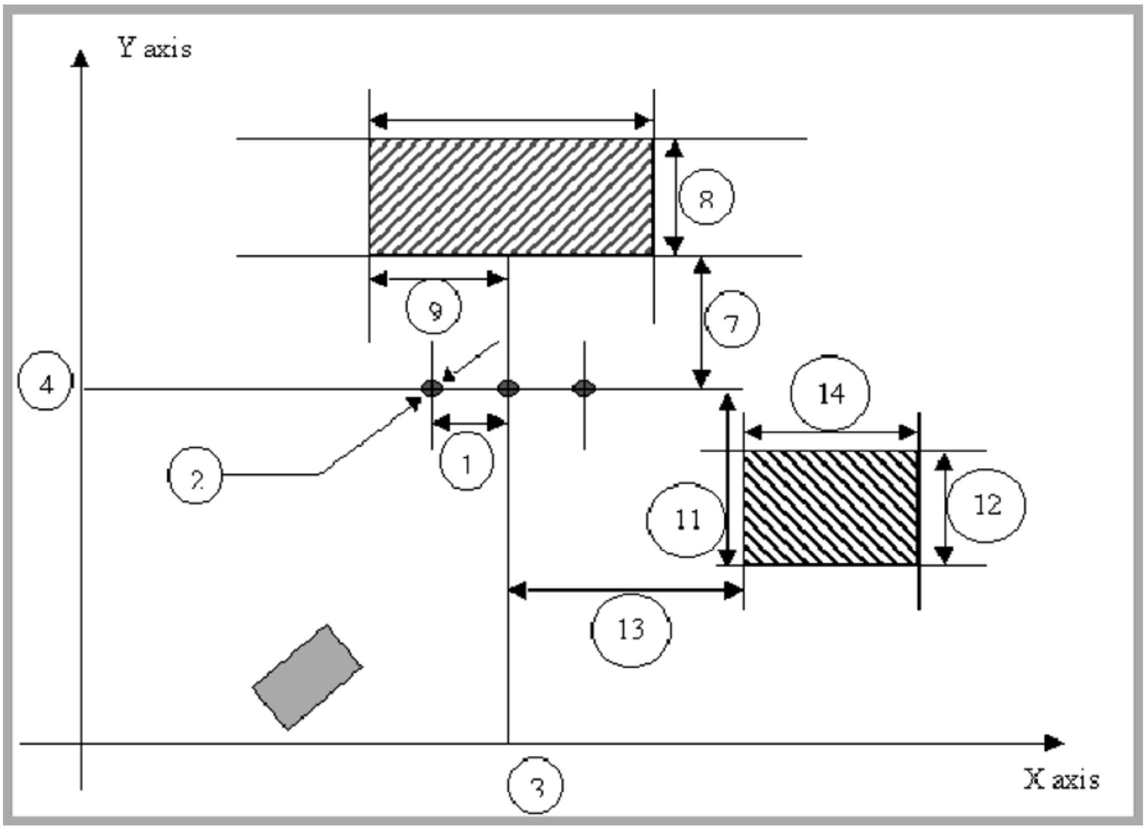


Figure 5-2. Graphical description of a landmark with the required software parameters

13. Defines one limit of the region that must not contain points (line perpendicular to line of poles).
14. Defines other limit of the region that must not contain points (line perpendicular to line of poles).
15. Orientation of the poles with respect to the coordinate axis

5.6. Example of a localisation set-up

The following is a description of a set-up for localisation that demonstrates, in brief, the nature of the tasks required to erect and use a landmark.

Figure 5-3. Setting for robot mission.

The robot is located in a wide hall with (Figure 5-3) with ample space for the robot to move around. A mission for the robot is defined, in LAMP, wherein the robot must:

1. Approach a certain wall,
2. Set itself such that its left side is parallel and closest to that wall, travel straight ahead till
3. It comes upon a narrow passage that it must negotiate,
4. Travel some distance before turning around and negotiating the same path in reverse order.

In the absence of an initial localisation procedure, some more instructions would have to be added at the start of the mission in order to bring the robot to some initial orientation and/or position. With the addition of the localization algorithm the insertion of two instructions allows the robot to achieve the same effect. The instructions are to attempt localisation and, in case of failure, to rotate about itself by some angle and try again

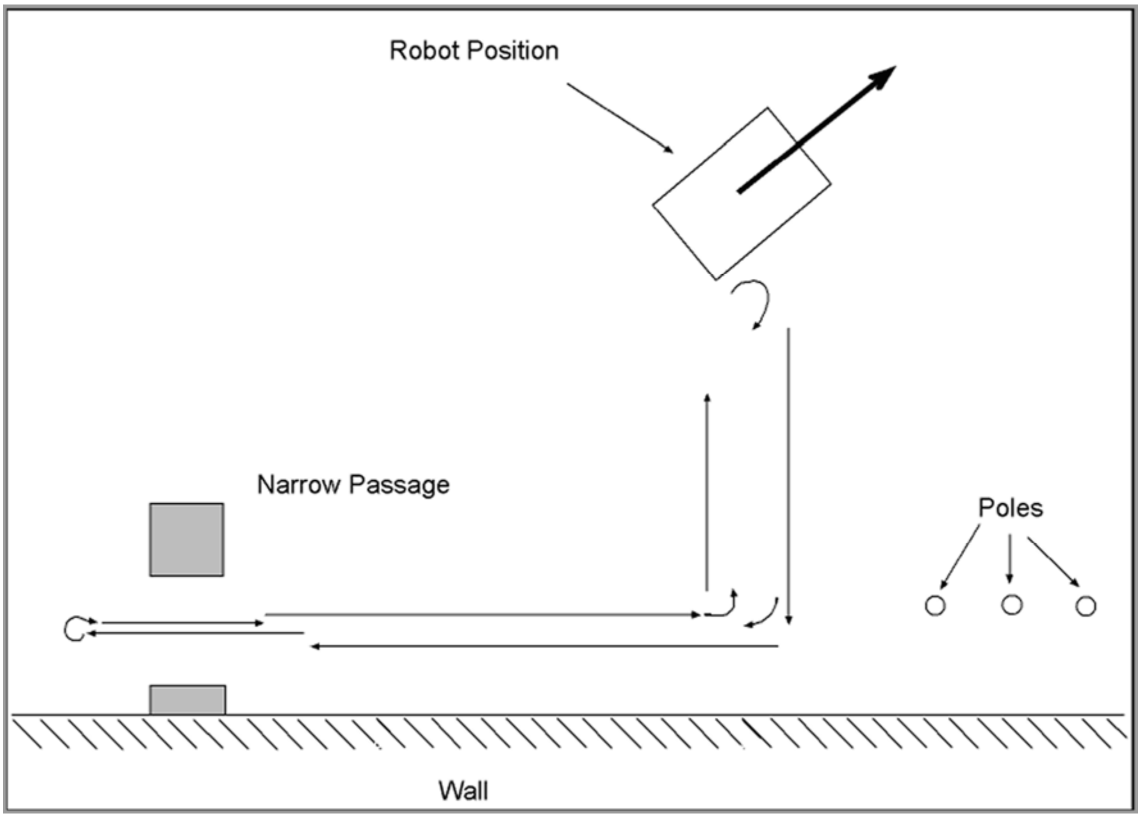


Figure 5-4. Sample robot mission illustration

Three PVC pipes of diameter circa 13 cm are arranged in a line at a distance of about 70 cm from the wall, separated by about 50 cm. In this particular application, only the orientation of the robot can be used, ignoring the actual position of the robot. The wall itself (white wall below staircase in Figure 5-3) is utilised as the confirming surface for the confirmation phase of the algorithm. Also, the region that should be devoid of points for purposes of confirmation is not utilised (only the region containing points, in this case the wall, is specified).

The parameters for the algorithm are listed below followed by a listing of the mission in LAMP.

1. Spacing between the poles = 0.5
2. Width of the poles = 0.13
3. X-coordinate of the central pole = 0.
4. Y-coordinate of the central pole = 0.
5. Landmark Contiguous Spacing (LCS) = 0.25
6. Group Spread (GS) = 0.4
7. Limit of region that must contain points (line parallel to line of poles) = -0.6.
8. Tolerance of region defined by above line (line parallel to line of poles) = 0.3.
9. Limit of region that must contain points (line perp. to line of poles) = -1
10. Tolerance of region defined by above line (line perp. to line of poles) = 1
11. Limit of region that must not contain points (line parallel to line of poles) = 0.

12. Tolerance of region defined by above line (line parallel to line of poles) = 0.
13. Limit of region that must not contain points (line perp. to line of poles) = 0
14. Tolerance of region defined by above line (line perp. to line of poles) = 0
15. Orientation of the poles with respect to the coordinate axis = 0

MOVE LV 25 AV 0 USG 500 SEN 1	Move with a linear velocity of 25 cm/s till the sensor one measures 500mm.
MOVE LV 0 AV -10 ANL 80	Turn 80º, in a clockwise direction
MOVP MV 25 PS 0 DIL 500	Travel parallel to the wall, at least some 500mm, at mean velocity of 25 cm/s.
CROS SD 1200	Cross narrow passage and travel 1200mm
REVR	Reverse direction (turn about 180º).
CROS SD 900	Cross narrow passage and travel 900mm
MOVP MV 25 PS 0 DIL 1500	Move parallel to the wall on the closest side some 1500m at a mean velocity of 25 cm/s
MOVE LV 0 AV 10 ANL 90	Turn through 90º, anticlockwise
MOVE LV 25 AV 0 DIL 2500	Travel 2500 mm at a velocity of 25 cm/s

A C shell script file for Linux defines the entire mission including the initial localization algorithm. This script begins with a loop that includes a command to obtain the laser data string and run the localization algorithm. As long as the algorithm fails to recognize the landmarks from the laser data, the robot is rotated some 60º about itself and the loop is run again. A successful localization exits the loop and the Robuter CPU begin executing the lamp commands sequentially.

6. Conclusions

This work involved the use of an LRF to obtain a position estimate of the robot, relative to a specified landmark. The estimate would have to be obtained without the aid of any prior position information. The same method works in two distinct instances; localisation of the robot at the start of a mission, and localisation at the gantry robot of the FMS system.
The conditions defined for the success of the algorithm were:

1. Successful localisation within a defined area,
2. In the language of hypothesis testing, generation of a hypothesis with a low type II error at the expense of a higher type I error [2], i.e. the probability of obtaining a false estimate was to be kept to an absolute minimum, despite a resulting reduction in the rate of successful localisation.
3. Knowledge of the typical value of the error estimate in case of final localisation at the gantry robot (localisation at close range).

An important aspect of the method to be implemented is that it should reflect the principal idea behind the development of the LAMP programming language, namely the absence of a model of the environment. The method that was implemented involved the extraction and identification of special beacons inserted in the environment for the purpose of localisation. The beacons make up a composite landmark (consisting of more than one, individual, separable components), set up with respect to some other, more permanent, feature of the environment. After passing through a two-step procedure that seeks to extract and then identify the beacons, the existence of landmark is confirmed using the relation that is known with respect to the permanent feature of the environment (typically a wall). The output of this confirmation stage is then utilised to obtain the position of the robot. In this regard the position of the landmark in the desired 2-D coordinate system maybe communicated to the algorithm.
From previous discussions the following is summarised:

1. The method focuses on the robust extraction and identification of a single, defined landmark. No allusion is made to the use of a map model either for matching or for the identification of natural features existing in the environment. In fact, besides the information necessary for the placement of the landmark no information regarding the environment is utilised.
2. While an idea of the mean error of the estimate (at a particular distance, and for a particular landmark configuration) can be experimentally obtained, the error has not been mathematically quantified.

---

[2] A Type I error refers to the rejection of the hypothesis given that it is true, while the type II error refers to the acceptance of a hypothesis when the same is actually false

6.1. Inferences

The method implemented seems to be effective for the conditions specified for the initial localisation in the laboratory. The typical parking space of the robot can be covered in terms of position and orientation using either four separate landmarks or through the use of a single landmark (provided the robot can change its orientation), ensuring a short and purposeful mission start. The presence of a greater number of landmarks increases the chances of obtaining a correct location estimate. Setting the beacons such that a particular landmark configuration is linked to a particular position in the coordinate system of the laboratory, allows a quick set up for initial localisation.

The results of implementation of the algorithm at a public exhibition showed that the requisites of the algorithm, i.e. an unobstructed view of the beacons and of the confirming surface coupled with unoccupied spaces, is not without problems in the presence of many dynamic objects having access to the region of operation of the robot. The onlookers tended to place themselves in such a way that they either distorted the pattern presented by the beacons, so as to make it unusable, or obstructed the view of the beacons and the confirming planar surface. It is a situation in which Real-World conditions differed from laboratory settings. The presence of multiple landmarks alleviates this problem.

The lack of free space around the gantry robot allowed for the positioning of the poles in a particular layout that made them visible only when the robot was close to its final parked position. This brings up another algorithm requirement, namely, the need for some free open space (visible from the robot) around the area of operation in order to erect the poles.

For the purpose of final position estimation at the gantry robot, the localisation algorithm provides position estimates at close range with standard deviations of around one cm and angular deviations of less than three degrees. The orientation error and, to some extent, the position error might be reduced by using the known orientation of some major surface. This means that in some cases it might be acceptable to make some assumptions as to the nature of the walls in the building, more concretely, assuming the walls are orthogonal to each other.

It is important to note that most of the works reviewed in chapter 2 were methods in pose tracking. While some authors also include special schemes for localisation without a previous estimate (to be used at start-up and when the robot gets lost), these are usually modifications of the method implemented for pose estimation. Pose tracking requires that the robot posses some model of the environment. The various methods either provide such a model at the outset or allow for an initial map-building phase. The methods vary in the type of environment representation, the treatment of sensor data, the handling of environmental features and the form in which sensor data is integrated with map information. This work, however, does not deal with pose tracking. This means that it has no use for an environment model. Instead, somewhat detailed information regarding one or more features of the environment is explicitly specified and an identification of this/these feature/s is attempted. It does not depend on the nature of the method for navigation and pose tracking implemented on the mobile robot. While it has its obvious limitations, it is extremely easy to implement and has been shown to work in different indoor environments.

6.2. Future work

In its present state, the work is a simple implementation of a method for obtaining an initial location estimate. It fits well with the idea behind the LAMP mission language with good results in terms of precision and success in a variety of settings. While it is sufficiently robust in that the chances of a false localization are very low, it is not efficient enough to be utilised as a stand-alone method for localisation in a Real-World type of environment containing people and other objects having similar dynamic behaviour. In addition, it requires some modification, although minimal, of the environment. In the future, attention might be focussed upon the following possibilities:

- While the algorithm was implemented with success, first using Matlab and later Octave for Linux, the scripts written in the above two languages are slow in execution. The speed of execution should improve with the implementation of the algorithm in the form of a compiled executable file using a programming language such as C++.
- Replacement of the artificial landmarks with natural landmarks such as pillars, which possess similar characteristics. Some modified, more robust procedure might be found to achieve the goal of localisation in an unmodified environment. Such a method could then be applied as a procedure for localisation without a prior estimate, as in the current case, or even for pose tracking.
- Though it has not been explored in this work, the method might be utilised as a rough-and-ready means for obtaining the position of the robot in the laboratory either for the purpose of obtaining a second estimate, independent of a primary one or in aiding of a task such as environment mapping. In case of the former, an alternative to manual measurement using a scale might offer certain advantages. In the latter case it must be mentioned that the environment as ranged by the LRF appears very different from an architectural representation or even from a representation of the environment as produced from the US sensors. The different planes in which they work and the presence of unmapped or changing features, means that obtaining a map of the environment in the plane of the LRF is not a very straightforward procedure. The localisation algorithm presented in this work, together with inputs from a CAD model of the environment, might be utilised in methods aiming to build maps for use with the LRF.

6.3. Conclusions in a nutshell

The main features and the contribution of this work might, in point form be summarised as under:

- Absence of a map of the environment
- No a priori estimate of the robot position and orientation
- Requires landmarks that utilise multiple vertical poles together with information regarding some neighbouring permanent natural feature such as a wall.
- Extremely robust to false detections
- Easy to set up
- Good integration in existing navigation solution

The algorithm development was done primarily in Matlab for Windows® and subsequently ported to Octave, a Matlab clone for Linux.

7. References

[Arras and Tomatis, 1999] Arras K.O., Tomatis N., "Improving Robustness and Precision in Mobile Robot Localization by Using Laser Range Finding and Monocular Vision", Proc. of the 3rd European Workshop on Advanced Mobile Robots (Eurobot 99), Zurich, Switzerland, 1999.

[Arsénio, 1997] Arsénio A. M. A., "Active Laser Range Sensing for Natural Landmark Based Localisation of Mobile Robots", Msc. Thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.December 1997

[Borenstein et al., 1996] Borenstein, J., Everett, H.R., Feng, L., " 'Where am I?' Sensors and Methods for Mobile Robot Positioning." Technical Report, The University of Michigan., 1996

[Burgard, 2001] Burgard W. "Probabilistic Techniques for Mobile Robots", Tutorial – European Summer School on Mobile Robotics, Lausanne, 2001

[Cox, 1991] Cox Ingemar J, "Blanche – An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle" IEEE Transactions on Robotics and Automation. Vol. 7, No. 2, April 1991

[Drumheller, 1987] Drumheller Michael, "Mobile Robot Localisation using Sonar", IEEE Transactions on pattern analysis and machine intelligence, VOL. PAMI-9, NO. 2, March 1987

[Dudek and MacKenzie, 1993] Gregory Dudek and Paul MacKenzie, "Modelbased map construction for robot localization", In Proceedings of Vision Interface 1993, North York, Ontario, May 1993, http://citeseer.nj.nec.com/dudek93modelbased.html

[Everitt, 1993] Everitt B. S., "Cluster Analysis(3 ed)", Arnold, 338 Euston Rd, London NW1 3BH, 1993

[Gray, 2000] Keith W. Gray, "Obstacle Detection And Avoidance For An Autonomous Farm Tractor", Master of Science in Electrical Engineering thesis, Utah State University, Logan, Utah, 2000

[Jensfelt and Christensen, 1999] Patric Jensfelt and Henrik Christensen, "Laser Based Pose Tracking", In Proc. of IEEE International Conference on Robotics and Automation, Detroit, Michigan, USA, May, 1999

[Madsen and Andersen, 1998] C.B. Madsen, C.S. Andersen, "Optimal Landmark Selection for Triangulation of Robot Position", J. Robotics and Autonomous Systems, Vol 13:4, 1998, pp277-292, http://citeseer.nj.nec.com/madsen98optimal.html

[Oliveira et al., 1999] Oliveira E., Dias P., Santos V., Almeida L., 1999 "Sistema de Atravessamento de Passagens Estreitas para um Robot Móvel", Revista do Departamento de Electrónica e Telecomunicações da Universidade de Aveiro, Vol. 2, n.° 4, Jan. 1999, pp. 547-551.

[Oliveira and Santos, 2000] Oliveira E., Santos V., 2000 "Giroscópios ópticos: Princípios de Funcionamento e Especificações", *Robótica,* Num. 40/41, pp. 43-50.

[Oliveira and Santos, 2000b] Oliveira, E., Santos V., 2000b "Fiber Optics Gyroscopes Evaluation and Calibration with a Mobile Robot", *Proc. of the 8th International Symposium on Intelligent Robotic Systems, SIRS2000, July 2000,* University of Reading, United Kingdom, pp. 281-286.

[Pradalier and Sekhavat, 2002] Cedric Pradalier and Sepanta Sekhavat, "Concurrent matching, localization and map building using invariant Features", Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland • October 2002

[Ribeiro and Gonçalves, 1996] Ribeiro M. I., Gonçalves J. G. M., "Natural Landmark Based Localisation of Mobile Robots Using Laser Range Data", Proceedings of EUROBOT' 96, IEEE, 1996

[Santos and Oliveira, 2001] Santos, V., Oliveira, E. 2001 "Missões de Navegação para um robot Móvel baseadas em Tarefas e Relações Topológicas do Ambiente", *Robótica,* Num. 45, pp. 14-22.

[Sequeira and Gonçalves., 1993] Sequeira V.A, Gonçalves J.G.M., JRC Technical Note No. I.93.132, October 1993, "Position Estimation of a Mobile Robot Using Laser Range Data"

[SantosJ et al., 2000] José P. O. Santos, J. J. Pinto Ferreira, José Mendonça, "Manufacturing Device Integration Builds on Shop Floor Functional Operations", In 9^{th} – Dedicated Conference on ASIM 2000, Berlin, 8-9 March, 2000

[Sim and Dudek, 1999] Sim Robert and Dudek Gregory, "Learning visual Landmarks for Pose Estimation", In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Detroit, MI, May 1999. IEEE Press. http://citeseer.nj.nec.com/sim99learning.html

[Singhal, 1997] Singhal A., "Issues in Autonomous Mobile Robot Navigation", Computer Science Department, University of Rochester, May 1997, http://citeseer.nj.nec.com/singhal97issue.html.

[Thompson and Zelinsky, 2000] Thompson S., Matsui T., Zelinsky A., "Localisation using Automatically Selected Landmarks from Panoramic Images", Proceedings of Australian Conference on Robotics and Automation (ACRA 2000) Melbourne, Australia, August 2000

[Thrun, 1997] Thrun Sebastian, "Learning Maps for Indoor Mobile Robot Navigation", Computer Science Department and Robotics Institute, Carnegie Mellon University, September 1997.

[Thrun, 2002] Thrun S, "Robotic Mapping: A Survey", -CMU-CS-02-111-, School of Computer Science, Carnegie Mellon University, Pittsburgh, ", 2002, http://robotics.stanford.edu/~latombe/ cs326/2002/thrun-mapping.pdf

[Vale et al., 2001] Alberto Vale, João Simões, José Machado, Pedro Lima "Navigation in a Maze", Revista Robótica 2001, No 45 – 4º Trimestre de 2001, pg 66-70

[Wulf, 2001] Oliver Wulf, "Recharging Station for a Team of Mobile Robots", University of Hannover, Institute of Automatic Control (IRT) and Royal Institute of Technology, Stockholm, Centre for Autonomous Systems (CAS), September 2001