Islands of Reliability for Hybrid Topological-Metric Mapping

Shlomo Saul Simhon

TR-CIM-99-01 January 1999

Mobile Robotics Laboratory
Centre for Intelligent Machines
McGill University
Montréal, Québec, Canada

Postal Address: 3480 University Street, Montréal, Québec, Canada H3A 2A7
Telephone: (514) 398-6319 Telex: 05 268510 FAX: (514) 398-7348
Email: cim@cim.mcgill.ca

ABSTRACT

This thesis describes a method for mapping unknown large scale static environments using a hybrid of topological and metric representations. A global map is formed from a set of local maps for sub-regions of the environments. Each local map contains quantitative environment information used to define a local reference frame. These maps are referred to as *islands of reliability* because they are associated with the sub-regions whose local structure is best matched to the sensors we are using. The connectivity of these islands is represented topologically. The key mapping problem we consider is *where* to place the islands of reliability and to what extent they should cover the environment. This is accomplished by defining the placement criteria in terms of the task to be satisfied and the uncertainties of the mapping agent. Islands are distributed about the environment at areas suitable for extracting metric information relevant to a localization task.

RÉSUMÉ

Cette thèse décrit une méthode pour cartographier une inconnu large échelle statique environnement en utilisant une hybrid de topologique et métrique représentation. Une carte globale est crées à partir des cartes local pour des sous-région de l'environnement. Chacune des ces cartes locales détiennent de l'information quantitative, utilisée pour définir un cadrage de référence local. Elles sont référées à des *iles de fiabilité* puisqu'elles fournissent de l'information métrique très précise de l'environnement. Ces iles sont disposés par une structure topologique qui inclus les descriptions de leurs conjonctions. Nous considérons que notre problème clef dans la cartographie est: où placer les iles de fiabilité et à quel étendue elles devraient couvrir l'environnement. Ceci est accomplie en définissant le critère de placement selon les termes des taches des iles de fiabilité qui sont destinées à satisfère et les incertitudes des cartographes. Les iles sont distribuées vers l'environnement dans des régions approprié pour extraire de l'information pertinente pour une tache de localisation.

ACKNOWLEDGEMENTS

My thanks are to Gregory Dudek for providing outstanding supervision over this work along with the brilliant insights and ideas that I couldn't have done without. It is a pleasure to be apart of his team where I may expand my knowledge and understanding in a dynamic and open environment. I further thank Greg Dudek for his financial support over the past terms. I would also like to thank Paul Mackenzie and Chi Zhang for their help and explanation on their thesis work, allowing me to incorporate their software into mine.

DEDICATION

This dissertation is dedicated to my wife Stacy Simhon and our families. I would like to thank Stacy for the greatest support that I could ever have ask for. Making every difficult challenge that seemed impassable a small hill to climb. I would not have made it this far had it not been for my Stacy. I especially want to thank my parents for being the greatest parents I can ever have. I find myself very lucky being brought up by such noble parents. When it comes down to it, all my learning and studies are based on the foundation I was given by my parents. I would also like to thank my brothers for all those great times that we shared and am exited to see what comes next!

Contents

List of Figures

CHAPTER 1

Introduction

This thesis presents an approach to mapping initially-unknown large scale static environments in the context of autonomous mobile robotics. We examine the problem of deriving a suitable representation for such environments that accounts for the uncertainties of a real mobile robot. Due to the nature of the problem, the model used for the representation is a dual topological-metric model. Our approach allows a mobile robot to autonomously construct a map while taking into account its sensory and odometric limitations.

1.1. Motivation

Most autonomous mobile robots are required to have some ability to remember and represent their surroundings. The ability to do so extends the capabilities of these agents, specifically when interacting with the surroundings in accordance to certain tasks. Some preprogrammed agents are limited to a predefined environment representation that is permanently embedded in their programming. More complex autonomous agents can dynamically construct representations by sensory data acquisition. Humans, for example, can explore unknown environments and represent sensory information in both symbolic or topological terms as well as quantitatively [**2**]. On the other hand, a standard assembly line robot has a limited representation of the environment along with a set of instructions that focus on a single task. In any event, maps are an important component when interactions with the environment are required. They form guidelines as to what and how commands are executed.

The actual execution of commands poses an important issue when dealing with real mobile robots. The basic approach to perform an action is to control the mechanical components of the robot while maintaining internal pose estimates. However, for real robots, their internal estimate of their position in space is prone to large errors. Generally, an estimate is generated by knowledge of the robot's own internal actions, tracking what motions have been executed. Position estimates based on this type of internalized modelling is knows as *dead reckoning*. Measurements of successive actions are compounded and any errors within the measurements are accumulated rapidly. For example, when a wheeled robot is used, its wheels tend to slip over the terrain while driving. This is increasingly amplified on a rough terrain (one covered with bumps and cracks). Over large scales, this results in a significant discrepancy between the internal position estimate and real position.

When a robot's internal position estimate is significantly offset from the true position, interactions with the environment can result in undesirable side-effects. To avoid this, the position estimate must be re-calibrated before the robot can continue the ongoing task. For an autonomous agent, this can be accomplished by extracting environment information and comparing it to an internal map. As such, the robot can determine its position in the reference map (localize) and update the dead reckoning estimate.

This leads to the conclusion that, for many tasks, a mobile robot system requires some ability to acquire data of the surroundings, even when it is operated in a known environment where a map is available [**2**]. While in certain cases it may suffice to provide predefined instructions that are blindly executed (such as the assembly line robot). It may not be possible to attain an acceptable level of accuracy upon the execution of instructions, especially when dealing with large scale regions where accumulated robot errors are amplified. Sensory feedback may be necessary to recalibrate and update the robot's position within some internal map.

This map based position update raises two key issues that need to be dealt with. Firstly, the position estimate is subject to the uncertainty of the sensing system. Imperfect sensors pose limitations on what environment features can be sensed accurately. Secondly, even with perfect sensors, attempting to match sensory and map data at areas where the map provides insufficient constraint may drastically offset the robot's pose, even more so than the dead reckoning estimate. (Although there are techniques using Kalman filters, discussed later, that use both dead reckoning and data matching estimates to output a new estimate rather than only relying on one of them.) Due to regions containing structural ambiguities, the utility of the internal map is not necessarily uniform.

Another important consideration in this regard is to determine how the *a priori* map was generated in the first place. Do we consider the map as unreliable data or is it an accurate architect's drawing that was fed into the robot? It may seem that an architect's drawing might be an ideal map, however, a representation such as this does not necessarily match the types of measurements acquired by robotic sensors [**2**]. For consistency, the representation of the environment should be constructed using the same sensory system and modelling method that is used during localization. The map should be built as the robot sees the world. Furthermore, an architect's drawing requires *a priori* knowledge of the environment and is not available in unknown areas. Therefore, it is the autonomous robot itself that must explore and map the environment.

This illustrates some of the issues in properly acquiring maps. In particular, it is preferable that the robot itself maps the environment as it is sensed, rather than to read in a manually generated representation. Since the robot is not perfect, we must develop a method such that the robot can build and use an accurate and concise environment representation despite its uncertainties. Furthermore, the importance of maps is outlined in terms of their utility for extracting a position estimate. The two issues that we will deal with in using maps for localization are the uncertainties in sensing information and the identification of regions that lead to ambiguities.

Using maps for localization is just a particular instance of their application. In general, maps are required in order to accomplish many other actions. However, it is difficult to produce general purpose maps since different problems require different environment representation so that they can be solved efficiently. Environment information is usually modelled in context to a specific task. Thus, we put an emphasis on differentiating between *generic* maps and *task specific* maps. This separation is necessary since in this work we must locally characterize the utility of a map. This allows us to build our topological-metric model as a composition of several *task based* maps. Thus, when we send the robot to explore and map the environment, we must define the configuration of the map to satisfy some goal rather than executing a generic data collection algorithm.

In this thesis, we consider building a map with a navigation task in mind. The representation and configuration of the map is modelled in consideration of this task, taking into account the physical limitations of the agent utilising and constructing the map. We note that two main limitations of a real robot are:

- odometric errors
- imperfect sensors

Not only do we build a map such that the robot can deal with these limitations later, we must also deal with these limitations while constructing the map itself.

A common misconception is that a one can eliminate these limitations by perfecting both the mechanical and the sensing systems of the robot. This tends to be an infeasible solution since there will always be different environment structures that can exploit some unaccounted limitation. Furthermore, this view of perfect mechanical and sensing systems is not justified by human or animal behaviour. Instead we consider adapting to such limitations by intelligently interacting with the environment, examining where these limitations are amplified and where they are minimized.

1.2. **Background**

1.2.1. **Topological Maps.**

Topological mapping is a method of qualitatively describing the environment. Standard representation for this is in the form of a graph, providing an efficient way of symbolically connecting features in space i.e., neighbourhood relations between landmarks. Representing environments in this fashion simplifies certain computations by switching to the topological framework [**3**] where a good foundation exists. Difficult problems may be dealt with by well established search algorithms and graph traversal algorithms along with heuristics. Furthermore, a high level environment description forms a convenient way of storing information, consuming less storage space than a full metric description.

Current topological models of the environment include using visibility graphs to efficiently describe free space [**4, 5**]. Rao, Iyengar and deSaussure [**5**] propose algorithms to navigate an unknown polygonal environment by dynamically constructing the visibility graph using sensors capable of detecting edges and corners of obstacles. The global navigation algorithm consists of incrementally constructing the visibility graph while following a local navigation algorithm. Other work in topological maps deal with the problems of detection and identification of nodes to form a graph [**6, 7, 8**]. In that work, maps are built without metric information; only a topological hierarchy of nodes specifying neighbourhood relations amongst features. Each node in the map is detected by a set of non-unique signatures. Using a collection of these nodes, an extended signature is constructed to uniquely identify the nodes and avoid ambiguities. Mataric [**9**] identifies nodes by an environment property that is predefined by the designer along with a compass heading for additional reference. Graph exploration strategies using markers [**10**] are described by traversing a graph and placing markers at nodes. The markers identify previously visited nodes and detect topological cycles. Even shape recognition problems [**11**] are also formulated in a topological mapping framework. In most of these cases, once the topological representation is defined, complex computational tasks can be formulated using graph-based methods

Although current work in topological maps is theoretically sound, most limit the ability to metrically represent important environment structure. A purely topological representation does not include metric information, all measurements are symbolic. Thus, a robot will not be able to extract metric information when needed (for instance where position re-calibration is required). It is not clear that robots can only rely on symbolic information. Furthermore, topological maps alone are difficult to learn and maintain, particularly when their configuration is ambiguous [**3**]. The problem of defining an unambiguous graph without using a metric representation is not a trivial one.

1.2.2. **Metric Maps.** In order to perform accurate positioning, Dudek and Mackenzie [**1**] construct sonar-based maps where explicit models are built out of sonar reading distribution in space. The maps are used to determine robot pose by fitting new sensor data to the model. This is accomplished using a weighted voting of *correction vectors*; computing vector differences between observed data points and target model in the map. Holenstein *et. al.* [**12**] also build model based maps using ultrasonic data coupled with a localization procedure. During localization, new sensory readings are compared to the model map using clustering. Curran and Kyriakopoulos [**13**] construct metric maps from range data, complementing their localization algorithm that uses an Extended Kalman Filter to combine dead reckoning, ultrasonic and infrared sensor data. Dudek and Zhang [**14**] use a vision system to model the environment and extract positioning information. The model consists of extracting appropriate features from images and correlating them to pose. Position calibration is attained by training a neural network, which allows accurate interpolation through the feature-pose space. Krotkov [**15**] also uses a vision system to determine pose by establishing correspondence between observed landmarks and map landmarks. The approach uses objects commonly found in indoor environments as landmarks (in particular, vertically oriented parts of fixed objects such as doors, desks and wall junctions).

While these traditional metric methods can determine a pose estimate without the need of having artificial beacons placed about the environment, they use *a priori* maps that are usually built using a single coordinate frame. Although such maps provide accurate local correspondence, accumulated errors tend to warp the representation over larger scale areas. Data measurements taken far apart are susceptible to large errors. Using such erroneous relations may mislead position estimation computations. These methods lack the ability to model large scale environments without distortion, especially in unstructured regions and rough terrains.

To compensate for the limitation of accumulated error, there are mapping techniques where semi-continuous localization is used. Leonard and Durrant-Whyte [**16**] and Lu and Milios [**17**] employ such methods. The work provides a self consistent description of the environment by using a Kalman Filter and registration techniques to compare predicted and perceived data while updating a map.

These methods, like many others, can lead to two key problems if used indiscriminately. Firstly, time and energy may be wasted in attempting to accurately map regions irrelevant to the tasks of interest. Secondly, they may attempt to detect landmarks and establish a reference coordinate frame in regions where the local structure is ambiguous or unreliable. Consider a long white hallway, where information gathered is irrelevant for a localization task. If pose estimation is to be extracted in such an area, it may result in an erroneous and misleading solution. Furthermore, metric relations between two rooms at each end of the hallway cannot be accurately determined, nor are they important. Thus, attempting to construct a single absolute reference coordinate system can be problematic [**18**].

1.2.3. **Topological-Metric Maps.** It is not always beneficial to keep metric relations over large scales (since the relation are usually erroneous). A better alternative might be to provide topological or qualitative relations over such extents, while storing metric relations over local areas. Prior work in cognitive science suggests humans use a set of *local* reference frames topologically connected to model large scale environment. Yeap [**19**] shows that a module of the Cognitive Mapping Process can be represented with a Relative Absolute (R-A) model. The model consists of a global representation (referred to as Relative Space Representation, or RSR) which is qualitative composition of a sequence of local representations {S(1), S(2),...} called Absolute Space Representations or ASRs. That is, the global map can be considered as a set of clear and accurate patches of local information linked topologically by fuzzy, semi-unknown areas (Figure 1).

This is easily exemplified by a person travelling down a street. While walking on the uninteresting sidewalk, the person's attention is often diverted from the environment (reallocated to other thoughts); the description of the environment is vague. When reaching a point of interest or distinction, such as an intersection, the person redirects his attention to the environment in order to accurately localize to the sidewalk edge, check the street names, signals etc. At this point, the environment's precise structure is re-acquired. Thus, the Human cognitive map can be conceptualized in both qualitative and quantitative components, the quantitative components are the local patches of interest (ASRs), used for computational purposes, while the global hierarchy organization of the patches is the qualitative term (RSR). Furthermore, the ASR configuration is dependent on the humans' experiences and goals. For example, if the person walking on the sidewalk was a painter with a goal to paint the sidewalk, what is normally an uninteresting sidewalk becomes a part of the ASR.

Work by Nehmzow [**20**] suggests that foundations of robust robot navigation are based on using landmarks, canonical paths and topological models. Examples of animal behaviour justify this view. It was shown that animals use as set of landmarks such as prominent trees, rocks etc., as well as reference landmarks such as sun, stars

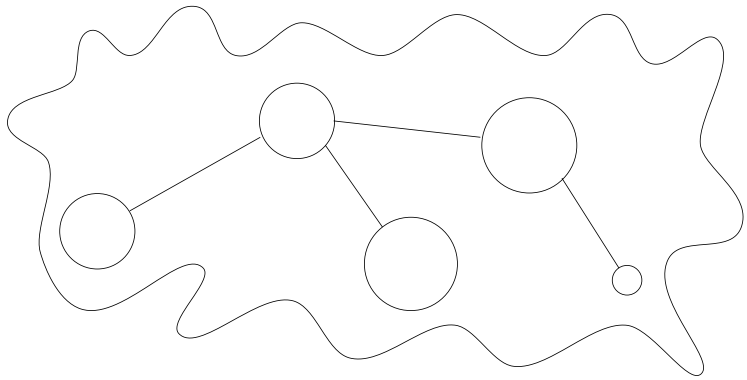


FIGURE 1. Global map composed of a set of local maps. Circles represent metrically accurate local maps. Edges represent qualitative information.

and magnetic senses to navigate. Furthermore, constant, well defined paths were preferred even if it means longer travel distances.

Kuipers and Byun [**21**] develop a robot mapping and exploration strategy composed of both qualitative and quantitative components. Their method considers distinctiveness measures in terms of certain pre-defined sensory criteria that can be maximized. The map is composed of a set of edges (distinct paths) defined by 2-D distinctiveness measure criteria, and a set of nodes (distinct places) defined by 1-D distinctiveness measure criteria. In their work, a robot explores an environment by following control strategies that maximize the 2-D criteria. This forms well defined paths about the environment and constitute the edges of the topological map. During the exploration, the robot gathers sensory information for the metric map while also looking for distinctive regions to include as nodes in the topological map. Once in the neighbourhood of such a region, a hill climbing routine is executed to maximize the 1-D distinctiveness measure criteria and reach a distinct point.

A *rehearsal procedure* uses geometric information gathered along the nodes to distinguish new places from old ones. This is done by first assuming the new node is an existing node. Then searching the graph up to a certain depth to determine if the assumption is valid. The paper [**21**] , however, only indicates tentative guidelines to construct the sensory criteria. Inappropriate sensory criteria can result in non-unique solutions, the placement of edges and nodes may not be clearly defined. Furthermore, metric information is gathered in global correspondence, therefore accumulated dead reckoning error may distort true metric relations resulting in an erroneous map.

Thrun produces hybrid topological-metric maps, similar to those of Kuipers and Byun using a subset of the Voronoi diagram computed from the metric map to define the topological components [**3**]. First, the metric grid-based model of the environment is built by interpreting sensory data with an artificial neural network. This maps the data into the probabilities of the occupancy. Bayes' rule is used for integrating multiple interpretations over time. Then, *critical* points that minimize local clearance of the occupancy grid are used to obtain critical lines that partition free space into disjoint regions. The partitioning is mapped to a graph. By combining both topological and metric representations in this fashion, accurate/consistent and efficient maps are constructed for navigation. Although, the metric component itself is subject to large scale odometry errors and the topological points are distributed arbitrarily (they may be very numerous).

1.3. **Outline**

The work described in this thesis is motivated by the way humans depict the environment as described in [**19**]. The model we use consists of a hybrid topological-metric representation configured for a given task. We present this thesis as follows: chapter 2 discusses the approach we take to build a map and the necessary modules. In chapter 3, we discuss the topological-metric model and its configuration. Chapter 4 describes the models used to represent the nodes in the topological map and chapter 5 defines the spatial configuration of nodes. Chapter 6 gives some experimental results and a discussion. We then expand on future work in and conclude in chapter 7.

CHAPTER 2

Approach

Providing a concise description of large scale environments proves to be a difficult problem in the mobile robotics field. Generally, a robot explores an environment gathering data (at each sampling position) using a single sensory system. During the exploration, the collected data is matched with internal odometric position measurements, typically using a single coordinate frame. This describes the raw map. This raw map is then encoded in accordance to predefined modelling specifications. Standard approaches of world modelling include fitting primitives or applying operators to collected data. These approaches are often encountered in vision where super-ellipsoid or other parametric models are used to represent sensory input [**22, 14, 23**]. Other work in data modelling includes forming models for sonar readings taking into account the intrinsic characteristics of the sensor [**24, 1, 25**]. Once the raw map is modelled, the final environment representation is stored to be used when necessary. This mapping architecture has several drawbacks when dealing with real world interactions.

Using a single sensory modality is often not adequate to provide an acceptable description of the environment. Many situations arise where certain sensory systems are preferred over others, where one sensory system reveals no information while another can extract useful features. Previous work in data fusion examines how to match up noisy data from several sensing system to form a consistent map [**26, 27, 13, 28, 29**]. Hackett and Shah [**26**] provide a survey of papers related to sensor fusion and categorize them. They looked at a number of methods of sensor fusion that use simple set intersections, heuristic rules, non-linear least squares fits and maximum likely-hood estimates. Other work [**29**] builds an architecture for fusing sensory systems (range finder, camera and sonar) to autonomously drive a vehicle. The work exploits sensor modality differences to produce complementary rather than competing perceptual processes. The ability to fuse data or to automatically select the best sensory system at a given location is a useful asset. Incorporating this within the mapping architecture improves the utility of the map.

A global reference frame or large scale data correlation poses several problems when mapping a real environment. While a single metric coordinate system is a natural way to map space and is effective over small areas, over large extents of space it becomes problematic. In particular, over large regions of space incremental position errors can accrue to cause large errors in the global coordinate system. Correlating sensory data with erroneous dead reckoning measurements skews the representation. This can occur even when beacons or landmarks are used to reduce odometry error. Further, this type of error can cause inconsistencies in a map when updating is performed, since updated information may be put in the wrong place. In many cases, it is sufficient to create local coordinate frames only in selected regions, where odometry error must be minimized. Kuipers, for example, considers this problem in the context of building topological maps and proposes using *rehearsal procedures* to identify nodes, eliminating the effect of odometric errors [**21**].

In this work, we are interested in mapping using a collection of local coordinate frames organized topologically [**30**]. Data collected in separate coordinate frames can be treated as independent local maps, each corresponding to a specific sensory system and modelling method. This allows to select the best sensing system at a given location. Furthermore, all metric relations are consistent with the real world since they are gathered in local areas, avoiding large scale errors. Each frame is considered as a separate local metric map. The global hierarchal organization of these local maps comprises the large scale map (figure 1).

Where and how should we attempt to create a local metric map so that it will be accurate and effective? To answer this, we must first determine what each node represents, what computational procedure will be executed at each local map. We consider how to evaluate the local environment with respect to an arbitrary localization procedure. Candidate locations can be found where local metric maps are generated specifically for localization. Hence, the mapping criteria are based upon the best locations suitable for localization. However, this mapping framework remains generic in that it can be used to construct maps where other tasks are involved. Localization based nodes were chosen since they form an example of an ubiquitous and challenging task of using maps. Furthermore, we will see later that they are essential components for building a navigation map.

In this work, we employ two localization methods illustrating the use of two drastically different classes of approach. We show that for each method, we can develop techniques that predict how appropriate a given region is for localization (and hence for metric mapping). These techniques are incorporated within a mapping architecture to produce a hybrid of topological and metric representations.

We propose a mapping architecture as follows: A robot explores the environment using some exploration strategy. While exploring, region selection criteria are evaluated for each sensor modality to determine whether the robot should build a local map at the current location. Once the criteria are satisfied, the robot begins to generate the local map of the region using the best sensory modality. During this phase, the robot also continues to evaluate the region to confirm its original decision. If the criteria are continually satisfied and the local map is complete, the robot stores and connects it within the global map. The robot then continues the next iteration.

The following lists the problems that are encountered in our mapping architecture:

- Environment exploration
- Region Selection
- Local map generation
- Topological connections

Although some of these problems are interrelated, to an extent, each one can be treated independently. We can build separate modules to find solutions to these problems where the complexity of the algorithms within each module can be treated independently. However, a compatible data model between certain modules is required. For example, the robot may be manually driven in the exploration step or it may use an existing exploration and control strategy to autonomously drive (such as [**31, 32, 33, 34, 35**]). However, there must be a compatible record of the exploration procedures executed such that the nodes may be connected together appropriately during the topological connection phase. Similarly, if the robot attempts to build a local map using a certain sensing system, the region selection algorithm must understand the information model used such that it can evaluate the environment properly. The specific modelling methods used to represent the local maps are assumed to exist in advance.

The main focus of this thesis is on region selection. The robot must autonomously determine whether an area is appropriate for metric mapping. It will be seen that this decision, the configuration of the map and what information flow is required within the above listed modules is based on the task the map is intended for along with the limitations of the agent constructing/traversing the map. The flow chart in figure 1 shows the architecture of our system.

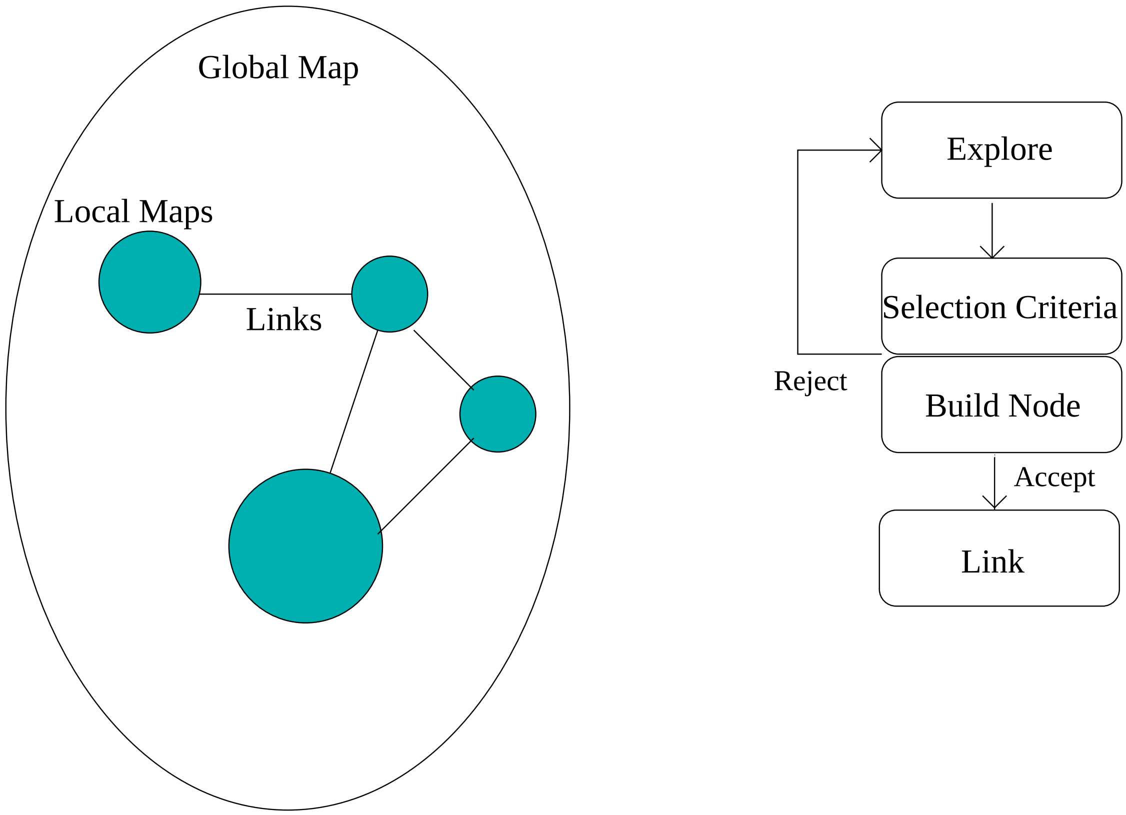


FIGURE 1. A global map composed of connected local maps (left) and a flow chart of the map construction algorithm (right). The robot explores until the selection criteria are activated. At that point, the robot builds a local map while intermittently confirming that the criteria evaluation is still positive. Once the local map is complete, it is linked to the hierarchy of local maps previously constructed.

CHAPTER 3

Islands of Reliability and Topology

In this work, the environment is represented by a set of accurate local maps. Each local map is built using its own local coordinate frame and modelling method. In theory, any sensory system and modelling method can be used to create a local map. Hence, it is possible to have a set of local maps where each map consists of information gathered with a different sensory system and modelled with a different modelling method. The local maps are assumed to be accurate i.e., they contain data or modelled data that is in accurate correspondence to the real world. These local maps are referred to as *islands of reliability*. When the robot lies in the vicinity of an island, it has the potential to perform local computational tasks involving real world data and stored data with accurate results. The environment is represented by a set of these islands, where each island forms an independent computational component of the global map.

The islands of reliability are organized in a topological structure. They constitute nodes of a topological model of the world, i.e., a graph [**21, 36, 37**]. Nodes are connected together by edges of the graph that include descriptions of their connectivity. An important note in describing the connectivity is to do so without use of *a priori* metric data of areas outside the nodes, otherwise it would not be consistent with our model. We define the global map, formed by the islands of reliability, as follows:

$$\begin{array}{l} M = \{V, E\} \\ V = \{v_0, v_1, ..., v_i\} \\ v_i = \{L,\ T\} \\ E = \{e_1, e_2, ..., e_i\} \\ e_i = \{v_i v_j, S, w\} \end{array} \tag{3.1}$$

M is the global map, it consists of a set of nodes V and a set of edges E. A node v_i corresponds to a local metric map L and the sensing/modelling type T used to build the local map. An edge e_i corresponds to two connecting nodes $v_i v_j$, a set of instructions S describing the connection and a weight w. Each edge carries a weight that may be related to the corresponding instruction set S or to some properties of the target node. This describes the environment by a set of clear and accurate patches of information that are organized topologically.

Consider a robot that lies at a node. As such, the environment structure is acquired in the form of a local map L of type T. When the robot veers away from the node, following an edge e_i, the environment is unknown until it reaches the next node where environment structure is re-acquired. The robot must exit a node following and edge. If it does not, it will no longer remain within the domain of the map and may not find it's way to the next node. This mechanism is similar to that of static attention focus where the robot only diverts attention to certain environment features while ignoring others. The features are unchangeable landmarks that can be used to perform metric operations.

By using separate local reference frames, we avoid the need to perform large-scale error integration. Metric data is gathered only within restricted regions. At each of these areas, data is mapped in an arbitrary way, forming independent local maps (islands of reliability). The global map consists of a set of these local maps distributed about the environment, selecting *interesting* regions for metric mapping and ignoring others. (The robot does not wast time mapping uninteresting areas.) Ignoring the uninteresting areas causes voids in the global map. However, providing procedures that describe the connectivity between adjacent nodes (within the voids) eliminates the need of an *a priori* metric map. These links may preclude any metric relations between the nodes. Nevertheless, it is not necessary to maintain these relations since over large scales they are erroneous due to accumulated error.

3.1. **Task Based Distinctiveness Measure**

The key issues in constructing the topological-metric representation described by equation 3.1 are selecting where to place the nodes and how to connect them. That is, we would like to determine where good candidate locations for local reference frames lie and record the topological connection of these locations. Once the regions and their topology are determined, the map is formed by both the metric data gathered at the regions and the descriptions that relate these regions. This topological-metric model, in the form of equation 3.1, decomposes the environment into qualitative and quantitative components. In order to determine these components, the purpose of the map must first be defined.

The model's configuration is dependent on the task it facilitates. Without knowledge of what goal the map is intended for, an appropriate configuration cannot be determined. Once the task is defined, the task itself can be examined and decomposed into qualitative and quantitative components. The quantitative components describe computations involving *a priori* metric data and real world data, while the qualitative components are high level instructions not requiring *a priori* metric data of environment features. These components define the configuration of the map. The mapping procedure can then be accomplished in 2 steps: 1) Areas relevant to the computational components are metrically mapped, generating the islands of reliability (V in equation 3.1), 2) The islands are connected together by including high level instructions that are defined by the qualitative components of the task (E in equation 3.1). The connecting instructions are subject to the model's *a priori* knowledge constraint; they are expected to be executed without the need of *a priori* metric information outside of nodes.

By defining the topological-metric configuration with relevance to a task, we form a general framework to construct the model. This approach is necessary since when looking for *interesting* landmarks to insert as nodes, we must first define what interesting means [**38**]. We define interesting areas as areas that provide information that satisfy the computational components of the given task. As such, the task can be executed by traversing the graph, applying the metric computations at the nodes and following the instructions at the edges.

This task based approach has analogies to models of human cognition [**19**]. Just as the ASR configuration is dependent on a person's goals (section 1.2), the node and edge configuration of our model is dependent on the robot's goals. For example, if the robot is to paint hallways, then the map is configured to facilitate that task. Nodes are placed at the hallways, such that the robot can accomplish the computational painting tasks. The hallways become areas of interest, or distinction in the map. Edges between the nodes may describe the painting order, the nearest neighbour hallway, the complexity of the painting area, what color to paint the next node, etc.. This map is a painters map as perceived by the robot. It stores and relates information according to a hallway painting task.

Different topological-metric configurations exist for different tasks, each with the corresponding node placement and node connectivity. Ongoing work in our lab [**39**] similarly produces a task-specific environment representation. In that work, the goal at hand is to present a suitable configuration for a human observer. Therefore, the nodal distribution is based on psychophysics, what catches a human's attention in the environment.

This illustrates that the topological-metric model described is a general one and can be used for many different mapping goals. So long as the map is tagged with a task, we can decompose the environment at areas computationally relevant to the task and at uninteresting areas. Were those areas lie can be determined by a distinctiveness measure. We define a general form of the distinctiveness measure as follows:

Definition: *The distinctiveness measure R for a computational task T at an area A is measure of how well the structure at area A allow the accomplishment of the computational task T.*

Given a computational task, we can evaluate areas of the environment to predict if the computation can be accomplished properly. Areas that show success are labelled with a high distinctiveness measure and areas that fail are labelled with a low distinctiveness measure. R can be described as a laciency measure of the environment that transforms explicit features to quantitative measures based on the usefulness of data.

3.2. **Navigation Map**

The constraints for the spatial distribution of the islands of reliability (nodes) are determined by the computational task they facilitate. In this case, the map we intend to build is a navigation map. In order to properly and safely navigate, we should have the ability to localize [**12**]. That is, when an entity traversing the environment infers it is lost, it should have the ability to compare sensory information with stored map data in order to localize; to determine where it lies in the *a priori* map. Updating the position estimate relative to environment features provides a safer way to navigate than by dead reckoning. This fundamental computational task defines the islands of reliability as localization based maps. Consequently, the computational component for the navigation task, involving *a priori* information and real world information, is localization. Hence, islands are placed at areas that provide good localization information. Calculating the distinctiveness measure for the localization task provides the constraints necessary for a beneficial distribution of the islands.

It is possible to form the islands of reliability using different sensory systems and modelling methods. Furthermore, there exists many different localization techniques, each with their own strengths and weaknesses. Therefore, the formulation of the distinctiveness measure R depends on the actual localization scheme and island models. However, general guidelines can be provided in order to evaluate the environment and select distinctive regions. The general form of the distinctiveness measure R for localization can be described as the following:

$$R_{Loc} \propto \frac{f(I, \Delta I)(1 + \sum_j \lambda_j Q_j)}{(1 + \sum_j \lambda_j)} \tag{3.2}$$

where I represents the reliability and strength of the response of a sensing technique and ΔI represents the amount of spatial change of that response (which may be expressed as spatial constraint). $f()$ is some function monotonically proportional to both I and ΔI. Q_j is a quality measure specific to the properties of the localization technique and λ_j is a corresponding weight. That is, to successfully perform localization, there must be sufficient reliable information I subject to spatial variation ΔI along all degrees of freedom. The addition of ad-hoc quality measures Q_j, specific to the technique, can improve region selection. However, most of the emphasis is on searching for areas with enough reliable information subject to spatial change (low structural ambiguity). A good choice for $f()$ is one of the form $I * \Delta I$, taken along the dimension that results in the minimum value. As such, R is large if both I and ΔI are large in all degrees of freedom.

Note that it is not suffice to evaluate R using a single one of these two elements, I or ΔI, both are necessary for localization. For example, if the evaluation is based only on I, areas with reliable features in the form of a pattern, say a fence or wallpaper, would show a high measure. However, localization at those areas is ambiguous (ΔI is zero at the corresponding scale) and a node at those positions would not satisfy the localization task. Similarly, if the evaluation was based only on ΔI, then features with good spatial variance may show large ΔI, but the features themselves may not be reliable sources of information. Therefore, we combine the strength of features along with their spatial variance. Islands placed at areas with a high distinctiveness measure render local maps with enough relevant features for accurate positioning (avoids placing them in ambiguous areas such as a long white hallway). An important note about these two elements is that they reflect the two discussed limitations of a real robot; uncertainty in sensing and positioning.

To complete the topological-metric model, it is not suffice to present the environment only by local maps. As mentioned earlier, the local maps form nodes of the topological model that are linked together by edges. The edges are an important component, providing instruction to follow between adjacent nodes. The types of instructions required to accomplish our task are navigation ones. The map should provide the ability to navigate from one island to another. Hence, the edges must include descriptions that allow reliable navigation between connecting nodes, without use of an *a priori* metric map. In this regard, the edge of a source node to a target node entail a set of control strategies that navigate the robot from a given position in the source frame to an estimated position in the target frame [**21**]. The idea of using such canonical paths is justified by animal behaviour [**20**]. These instructions may navigate a robot from one node to another, but exactly where in the node the robot may end up is undetermined.

Due to accumulated error, accurate robot positioning is not available when executing an edge's navigation instructions. We cannot assume that the robot would follow the instruction exactly, without errors. Furthermore, in accordance with our model, no environment data is gathered between nodes and no reference frame is given. It is not possible for computational position updates using environment features. All position estimates are based on internal measurements, such as dead reckoning. Thus, any attempt to extract a position estimate is subject to a degradation of accuracy as the robot diverts further away from the node. However, an error bound can be estimated among neighbouring nodes such that the robot's position can be bounded within an error radius. The robot can blindly reach *somewhere* inside a node (within a bound). Once the robot reaches the node, its position can be calculated more accurately using the corresponding computational metric map dedicated for such a task.

This further relates the node and link distribution to the dead reckoning error. To confidently navigate from node to node, the links should not be too long. The error radius should remain small, coinciding with the connecting island of reliability. The robot must avoid getting too lost in a link where it cannot find its way to the next island. When required, position calibration can always be done by reaching the connecting islands. Hence, the size of the islands are proportional to the accumulated error. Furthermore, we enforce a lower bound on the error radius. The lower bound prevents nodes from being unnecessarily close to each other when the qualitative control strategies suffice to navigate. Attempts to build nodes are only invoked once the robot infers its has crossed the lower error bound.

Assuming consistency of dead reckoning error for an exploring robot, we can define the node placement criteria as follows:

$$\begin{array}{l} \epsilon > E_{min} \\ and \\ max(\frac{R_i}{T_i}) > \frac{E_{min}}{\epsilon} \end{array} \tag{3.3}$$

where ϵ is the estimated accumulated error, E_{min} is the predefined lower error bound, R_i is the distinctiveness measure for model type i and T_i is the predefined acceptable distinctiveness measure threshold for sensing type i. When the exploring robot infers that it is lost (once the estimated accumulated error has crossed the lower bound error) the robot attempts to build an island of reliability using the available sensors. If the best distinctiveness measure R_i normalized by the corresponding acceptance threshold T_i is large enough, the island is inserted to the map. Otherwise, the robot continues exploring, searching for a good area. Note that the more lost the robot is the more willing it is to accept a potential island.

The size of the region mapped is related to the error the robot accumulated:

$$v_{radius} > \epsilon \tag{3.4}$$

However, the islands should not grow too large such that they remain locally consistent.

In summary, the navigation task is decomposed into qualitative and quantitative (computational) components. The qualitative components consist of navigation instructions that inform the robot where to go, what path to follow. The quantitative components are the islands of reliability that allow the robot to reposition itself. The environment is mapped according to these components to form the topological-metric model. The configuration is constrained by the distinctiveness measure for localization and the accumulated error.

CHAPTER 4

Local Map Perception

This section discusses the methods used for representing the islands of reliability. The methods rely on dead reckoning information for pose-data correspondence, therefore they are only locally consistent and hence are referred to as local maps. The modelling techniques are developed in order to satisfy the corresponding localization techniques which will also be discussed here. For a more complete description refer to [**1**] and [**14**]. We will briefly examine some of the details of these techniques since they describe the local map perception and the distinctiveness measure correspond to their encoding.

4.1. **Sonar Based Environmental Model**

In order to perform localization, a model is constructed of how sensory data varies as a function of the robot's position. The model resembles the layout of the environment but is not a veridical map describing the layout of real objects. That is, it describes the range sensor inputs as opposed to actual 2-D or 3-D occupancy. The model is built by fitting primitives to sensory data. Line segment primitives are considered efficient in modelling a collection of observations of the environment. Their utility is appropriate given the characteristics of simple threshold-based sonar sensing where even a small structure will produce collinear measurements. (Although, RCD modelling may form a better representation but are computationally intense when fitting to data.)

The line fitting method is done in several steps. First, a spatial clustering algorithm is employed to determine groups of neighbouring points that correspond to a potential line segment. Then, by using a line fitting procedure, a fitted line segment is used to model each cluster. Finally, a split and merge routine is applied to further segment and merge the lines at each cluster for a more proper fit. The final result is a map composed out of line segments that may be used for localization. A sample map is shown in figure 1.

4.2. **Sonar Based Pose estimation**

The pose estimation problem is formulated as an optimization problem in terms of the extent the map explains observed measurements. There are two phases involved in position calibration: 1) Classification of Data Points and 2) Weighted Voting of Correction Vectors. In the first phase, each measurement is associated to a line segment in the model using a clustering algorithm. This allows to determine the Correction Vector relative to the line segment in the second phase. An important note is that only the perpendicular error of points are used to determine their Correction

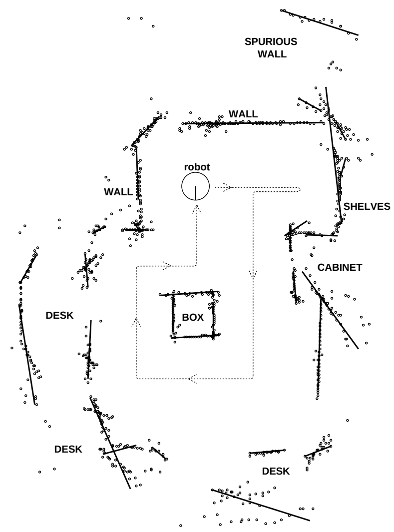


FIGURE 1. An environment modelled by line segment fits [**1**]. A robot explored the environment, following the dotted path, to build the line segment model.

Vector. It is a one-dimensional position constraint provided by each measurement along the normal of the associated line segment. This lack of constraint manifests itself as what is called the long hallway effect, where observation of the position of a line segment gives only information along its normal. Multiple measurements from non-parallel line segments are essential to constrain the problem in both dimensions. The second phase is that of a non-uniform weighting of Correction Vectors. Each point is given a weight in relation to the distance it lies from the associated line segment. The weighting factor is defined as a sigmoid function:

$$w(d) = 1 - \frac{d^m}{d^m + c^m} \tag{4.5}$$

Where d is the distance from the line segment, m and c are constants. Points near their line segment are weighted more than those that are far since far ones are probably outliers. The overall Correction Vector V is calculated as:

$$V = \frac{\sum_i w(\|v_i\|)v_i}{\sum_i w(\|v_i\|)} \tag{4.6}$$

where v_i is the perpendicular error vector for point i. The position estimate is resolved after several iterations of translating about the Correction Vector. Ideally, the measurements would be distributed equally in association to lines of both dimensions. This would allow a position estimate in both dimensions with similar confidence. Figure 2 illustrates the localization scheme.

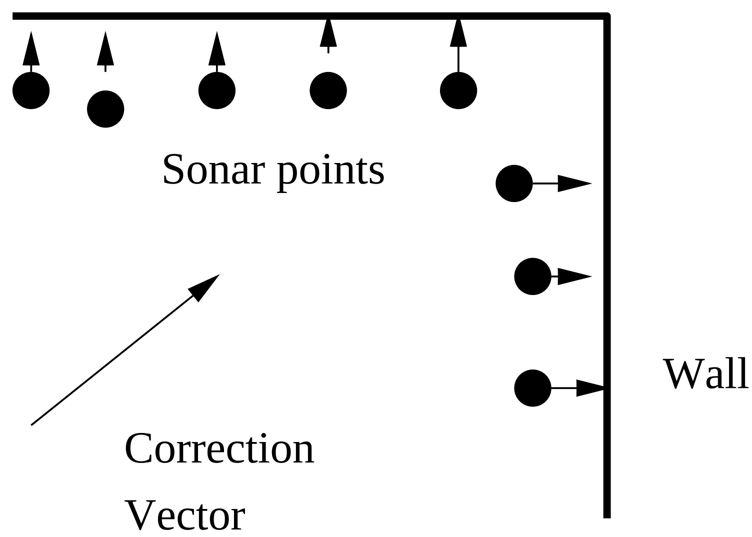


FIGURE 2. Localization by iterating about the Correction Vector consisting of a weighted sum of the error vectors.

4.3. **Image Based Environmental Modelling**

Building a description of the environment's structures using vision is known to be a difficult problem. Generally, parameter or CAD type modelling of the environment from a set of images entails solving the inverse problem defined by the surface geometry and reflectance. The method considered here differs in that is uses images in creating its own implicit model optimized for localization. It avoids the difficult problems of computing a 3D model and forms a perception of structures by statistically encoding image features as a function of pose.

For a camera mounted on a mobile robot, the dependency of the image and the pose $q = (x, y)$ is related by some function:

$$i = \Phi(q) \tag{4.7}$$

where i is an N-dimensional vector of pixels. In order to solve the problem of computing the camera position we must invert the function:

$$q = \Phi^{-1}(i) \tag{4.8}$$

However, computing the inverse directly on images is computationally impractical. Φ in itself is not necessarily one-to-one and an inverse may not exist. To produce a computationally tractable solution the images are modelled by a set of M features:

$$G(i) = \{g_1(i), g_2(i), ..., g_M(i)\} \tag{4.9}$$

This produces a lower dimensional space that relates the features and the pose with a mapping:

$$f(q) = G(i) \tag{4.10}$$

At each sampling position, the robot takes a snapshot of the environment. Features are then extracted and correlated with the odometric readings, encoding the environment representation in terms of a feature space. Given a new image, it is now computationally tractable to determine its features $G(i_{new})$ and interpolate through the a priori feature space to extract a pose estimate. Thus, lowering the dimensionality of the problem allows us to solve it more efficiently.

Measurement features were derived from statistical properties of edge images (using the Canny-Deriche operator) to minimize the effects of illumination variation. The perceptual structure associated with a position in space consists of the following class of measurements:

- First and second moments of the edge distributions
- Mean edge orientation
- Densities of parallel lines at four orientations

These features compromise the first central moments of the edge distribution in space and are the natural choices for efficient encoding.

4.4. **Vision Based Pose Estimation**

Since it is inefficient to sample the environment at every possible location and sensory data is often noisy, it is important to have the ability of robustly interpolating within the feature space. For local areas this can be done by a linear interpolator:

$$q = \frac{\mid G(i) - G(i_1) \mid (q_2 - q_1)}{\mid G(i_2) - G(i_1) \mid} + q_1 \tag{4.11}$$

Empirical results showed that the linear interpolator is only applicable to very restricted regions. In large regions or more complex areas, the linear interpolator fails. In practice, a three layer back propagation neural network is used. The network takes training examples and assigns appropriate weights to each network node by minimizing training set errors. When a new feature set is used as input to the network, the pose can be revealed by taking a linear combination of the output units.

Generally, if the feature space is smooth, the interpolator would output good results. On the other hand, if the feature space consisted of many gaps and discontinuities, interpolation between these gaps may produce inaccurate results. We have a trade off between practical sampling resolution versus accuracy of the interpolator. The sampling demand is determined by the complexity of the environment, choosing a sampling resolution for the most complex environment (worst case scenario) can be impractical. Although, environment structure may also amplify discontinuities no matter the sampling resolution.

CHAPTER 5

Local Map Distinctiveness Measure Criteria

When human observers are exploring new territories, their notion of the environment can be described by a set of distinct landmarks. Once they veer off from the last known landmark, they set out to find the next distinct feature to be recorded in the cognitive map. What is distinct to human observers is associated with their goals and perception. Similarly, we seek out the best (most distinct) parts of the environment corresponding to the robot's perception (which also corresponds to a task). The task at hand is (x,y) position estimation and the perception models considered are the sonar and vision systems (described in the previous section). The distinctiveness measure is derived accordingly. We derive methods to predict where the robot can extract pose information for the (x,y) coordinates based on the general guidelines of equation 3.2. The robot orientation Θ is not considered since only in rare cases (the presence of circular symmetry) will the robot be able to localize its (x,y) coordinates but not the rotation Θ. That is, in most cases, the distinctiveness measure for localization over (x,y) is the same as that for localization over (x,y,Θ).

5.1. Measure Criteria for the Sonar System

A good distinctiveness measure for the sonar based localization and modelling technique is one that assigns high values at areas well constrained (ΔI) by near line segments of significant length (I). That is, a local map with enough constraining information reflects a region where the robot can localize, while near line segments are more reliable than distant ones, providing more dense sensory feedback. Furthermore, it is desirable that the line model shows similar orthogonal constraints along both degrees of freedom. This results in equal localization confidence along the dimensions, keeping the error bound round. In the extreme case, a map constituting of parallel lines would result in ambiguities along one dimension and will not provide enough information to adapt the full potential of the localization method.

For the line model method, the distinctiveness measure R at a point $p = (x, y)$ over a square neighbourhood $(2\epsilon)^2$ can be calculated as:

$$R(p, \epsilon) = N(p, \epsilon) \int_{y-\epsilon}^{y+\epsilon} \int_{x-\epsilon}^{x+\epsilon} \frac{f(p)(1 + Q(p))}{2} \delta x \delta y \tag{5.12}$$

where,

$$f(p) = Min[f_{\perp}(p), f_{\parallel}(p)] \tag{5.13}$$

$$N(p,\epsilon) = \frac{1}{\int_{y-\epsilon}^{y+\epsilon}\int_{x-\epsilon}^{x+\epsilon}\delta x \delta y} \tag{5.14}$$

Q is a quality measure of the localization scheme. The functions $f_{\|}$ and $f_{\perp}$ depict the amount of reliable information I and spatial change of information ΔI along two orthogonal directions. The smaller of the two represents $f()$ in equation 3.2, choosing the weaker dimension as a worst case scenario. We integrate over a rectangular area defined by ϵ and normalize.

The next and most important step is to define $f_{\|}$, $f_{\perp}$ and Q. For these, we must first derive I and ΔI in terms of the line model. In general cases, the reliability of information I is proportional to the distance of a sensed signal. Distant lines provide sparse and less reliable information due to a weaker, degraded sensory signal. Furthermore, a line segment provides strong constraining information ΔI only along its normal. That is, an orthogonal position change with respect to the line guarantees a sensory measurement change. We define ΔI to be the orthogonal component of a sensed line segment.

For each line segment, we integrate the strength I and the orthogonal constraint ΔI to determine the *vector influence* along the normal to the line. We compute the vector influence for each visible line segment (in the form of $I * \Delta I$) by:

$$\mathbf{V_i}(p) = \hat{N}_i \int_{\Theta} W(\mathbf{pp_i}) * (\hat{N}_i \bullet \mathbf{p\hat{p}_i})\delta\Theta \tag{5.15}$$

Vi is the orthogonal vector influence for line segment i seen by point p and N_i is the unit normal of line segment i. Θ sweeps the visible viewing directions from the point to the line segment. Only angles within a reflectance threshold are taken in to account in order to simulate specular reflection of real range signals. p_i is the intersection point of line segment i and a line emitted from point p along the viewing direction Θ. The constraining relation for $\mathbf{V_i}$ is in essence a projection of the vectors formed from point p to line segment points onto the normal of the line segment (shown in figure 1). $W(..)$ expresses the reduced probability of observing an object as a function of distance. W is described by an exponential decay function:

$$W(\mathbf{v}) = e^{-k\|\mathbf{v}\|} \tag{5.16}$$

k is the decay constant that is determined by the range of sensor confidence.

Once the vector influence is computed for all visible line segments, we choose a *reference vector* and determine the total number of components parallel and perpendicular to it. These components determine the magnitude of constraint and reliability along two orthogonal directions and are calculated as:

$$f_{\|}(p) = \sum_{i\ lines} \mid \hat{\mathbf{V}}_{ref}(p) \bullet \mathbf{V_i}(p) \mid \tag{5.17}$$

$$f_{\perp}(p) = \sum_{i\ lines} \|\hat{\mathbf{V}}_{ref}(p) \times \mathbf{V_i}(p)\| \tag{5.18}$$

A good choice for the reference vector is that of largest magnitude, since it determines the dominating constraint. A poor choice may lead to inaccurate results.

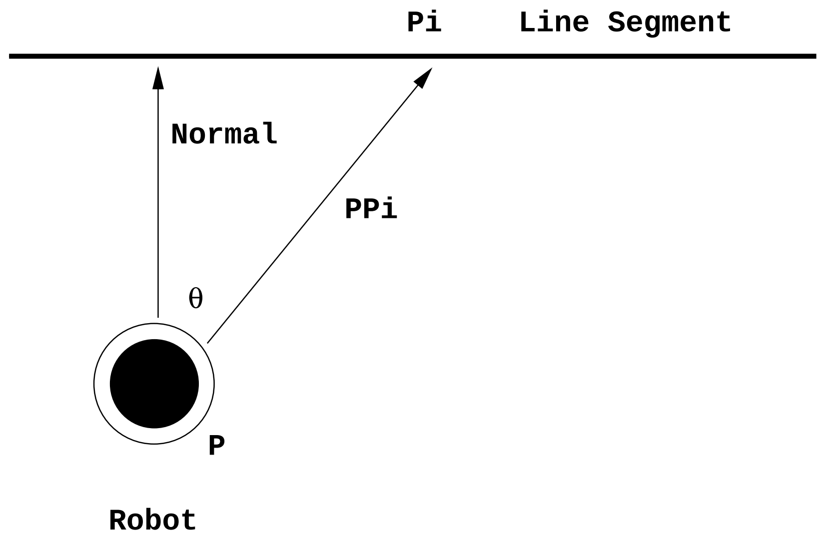


FIGURE 1. Sweep over each line segment to determine the overall constraint and reliability. Scanning resolution can be continuous or match that of the real sensors.

Consider a long line segment of slope 1 relative to a basis composed of a minute line segment. The only constraint parallel to the long line segment is the projection from the minute line segment. However, the computed parallel and perpendicular constraints are almost equal (since the slope is 1) resulting in a high distinctiveness measure. This is undesirable and to avoid it the dominating vector should always be the reference basis.

The functions $f_\perp$ and $f_\|$ describe the total strength and constraint of all the line segments visible from point p. However, the numerical value of these functions does not provide a decision threshold. To distinguish good valued from bad ones, we re-map f_i (i being $\|$ or $\perp$) using a sigmoid filter as:

$$f_i = \frac{f_i^m}{c^m + f_i^m} \tag{5.19}$$

The cutoff threshold c and the decay rate m can be found empirically. This allows us to empirically control the meaning of the numerical results. It provides a measure ranging from 0-1 where values above .5 can be considered acceptable for localization.

The extra quality measure is implemented by evaluating the equality of the above terms along both directions. This determines whether the localization confidence is equal along the two degrees of freedom. We compute a quality measure Q as:

$$Q(p) = \begin{cases} \frac{f_\perp(p)}{f_\|(p)} & \text{if } f_\| > f_\perp \\ \frac{f_\|(p)}{f_\perp(p)} & \text{otherwise} \end{cases} \tag{5.20}$$

Q ranges from 0 to 1 where 1 represents equal orthonormal constraints and 0 represents that only one direction forms a constraint.

5.2. **Measure Criteria for the Vision System**

In terms of the vision based localization technique, good regions are denoted by 3 properties of the feature space: a non-zero, non-constant and smooth feature space. Feature values that are low or close to zero are indistinguishable to noise and don't provide consistent reliability due to lack of environment information. (indicated by a small value of I in equation 3.2). A flat feature space would result in not enough spatial variance of features, leading to positioning ambiguities (indicated by a small value of ΔI in equation 3.2). A highly discontinuous space would reduce the interpolatory accuracy for position estimation due to lack of sampling information. Furthermore, equality of localization confidence along both degrees of freedom is desirable to keep the error bound circular.

For the vision system, the distinctiveness measure R at point $p = (x, y)$ about a viewing window $(2\epsilon)^2$ can be calculated as:

$$R(p,\epsilon) = N(p,\epsilon)\int_{y-\epsilon}^{y+\epsilon}\int_{x-\epsilon}^{x+\epsilon}\frac{f(p)(1+Q_1(p)+Q_2(p))}{3}\delta x\delta y \tag{5.21}$$

where,

$$f(p) = Min(f_x(p), f_y(p)) \tag{5.22}$$

and,

$$N(p,\epsilon) = \frac{1}{\int_{y-\epsilon}^{y+\epsilon}\int_{x-\epsilon}^{x+\epsilon}} \tag{5.23}$$

Q_1 is the quality measure for equal orthogonal constraints, Q_2 is the smoothness of the feature space and $f(p)$ represents $I * \Delta I$ of equation 3.2.

For a feature i in the normalized feature space G, we can determine f at a point p along a direction j (j being x or y) as:

$$f_{j,i}(p) = \mid G_i(p) * \frac{\delta G_i(p)}{\delta j} \mid \tag{5.24}$$

Figures 2(a) to 2(d) show some sample scenes and their feature values. It can be seen that the simple and noisy scene has many locations with small feature values that do not vary much. The details about the features used can be found in the work by Dudek and Zhang [**14**].

To properly evaluate the value of distinctiveness measure, for all the features we average the values $f_{j,i}$ to form f_j and map the result through a filter. We then need to reduce the weight of areas with low slopes and weak features while accepting larger slopes and stronger features. A sigmoid is used for this mapping:

$$f_j = \frac{f_j^m}{c^m + f_j^m} \tag{5.25}$$

We compute $Q_1(p)$ as:

$$Q_1(p) = \begin{cases} \frac{f_x(p)}{f_y(p)} & \text{if } f_y(p) > f_x(p) \\ \frac{f_y(p)}{f_x(p)} & \text{otherwise} \end{cases} \tag{5.26}$$

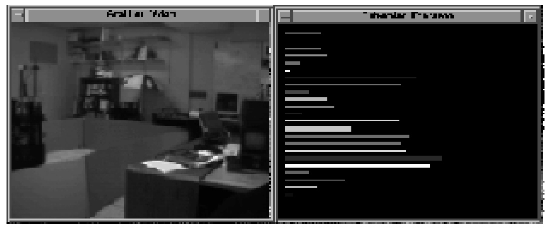
where a result of 0 implies unequal constraints and a result of 1 shows uniform constraint. We then derive the feature space smoothness measure as:

$$Q_2(p) = \mid \nabla^2 G(p) \mid \tag{5.27}$$

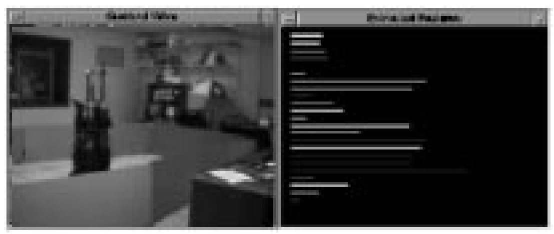
Further, a function is used to map Q_2. This would increase weight to areas with low values of $\nabla^2 G$ (smooth ones) and reduce areas with large value of $\nabla^2 G$ (discontinuous ones). Again the sigmoid in another form can be used:

$$Q_2 = 1 - \frac{Q_2^m}{c^m + Q_2^m} \tag{5.28}$$

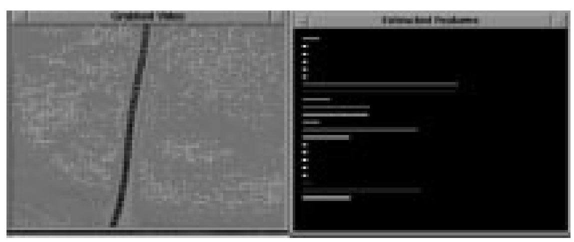
All parameters are determined empirically (see appendix).



(a)



(b)



(c)

(d)

FIGURE 2. 2(a) shows the features of the sample image and 2(b) shows the features of an image taken at a slight offset using the same scene. 2(c) shows the features of another sample image (less visually interesting than the first) and 2(d) is the same scene taken at a small offset. It can be seen how the feature values are low and do not vary for the less interesting scene while the more interesting scene shows good features with significant variation.

CHAPTER 6

Results and Discussion

In this chapter, we evaluate the the distinctiveness criteria by comparing them to the localization error. A good distinctiveness measure should assign large values at areas with low error and low values at high error locations. Comparing the position estimate confidence with the distinctiveness measure allows us to confirm or reject our ability to predict where important data lies and where data should not be trusted. We further proceed in discussing the results of building the topological-metric map using the mapping criteria (equation 3.3).

6.1. Sonar System

For the sonar method, a line segment model was manually constructed providing the simulated environment shown in figure 2. The distinctiveness measure for this sample environment is plotted in figure 3(a). These results were obtained as follows: First, at each sampling point in the map, complex simulated sonar readings were extracted using a robot controller/simulator developed in our labs. Then, the distinctiveness measure operator (equation 5.12) was applied on the data to output the plot. We can see how the long hallways show low measures while regions that are well constrained along both dimensions (such as intersections and bounding areas) show high measures. Furthermore, areas distant from line segments are of lower measure due to the exponential decay (the decay constant k was set to $1/200$ cm to simulate a large scale region). The sigmoid constants were empirically set using a minimum acceptable measure threshold as follows: an area with at least one line segment with minimum length of 50cm seen no further than $1/2k$cm along it's mid-line is considered acceptable. If there are no other visible lines and the seen line segment is smaller and further away, the measure will lie below the sigmoid cutoff. The neighbouring area ϵ of the operator was set to zero (such that measures consist of only a single point rather than an accumulation of a neighbourhood) and the reflectance threshold set to $30°$.

The next step is to determine whether the robot can localize itself at the locations associated with a large value of the distinctiveness measure. Figure 3(b) shows a plot of the localization confidence for the sample map. The data for this plot was generated using the same robot controller/simulator and the sonar localization/modelling software. At each sampling position, simulated sonar data was collected, thereafter employing a position offset by a random δ ranging 10-15 cm along four directions. The localization technique was then executed to output a position estimate. The error was calculated as the difference between the initial position to the estimated one, taking the average of the errors from the four directions. Confidence is simply $c - error$ where c is some constant. Similar to the value the exponential decay constant was set to in the above, the sonar confidence range was set 200 cm for consistency of scale (k=200). The reflectance threshold for the simulated sonar data was also similar to the previously set one.

When comparing the two plots (3(a) and 3(b)), we can seen how the confidence plot is consistent with the distinctiveness measure. Regions where the distinctiveness measure was high tends to be those with high localization confidence. Likewise, Valleys where there was insufficient constraints and the localization confidence is low are also those where the distinctiveness measure returns low values. When the robot was far from the walls, simulated sonar data was of lower accuracy causing the localization confidence to drop, similar to the distinctiveness measure. This match between the plots confirms our ability to appropriately evaluate the environment in relation to the sonar based localization method.

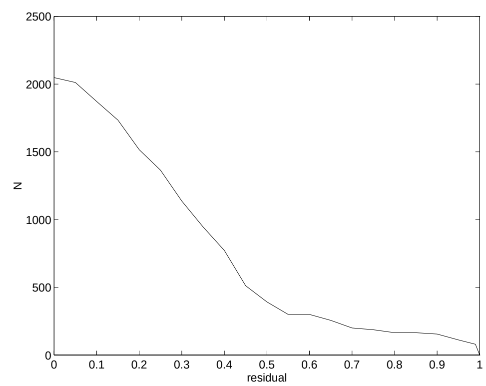


FIGURE 1. Residual plot for figure 3(a) and 3(b), there are N data points with differences greater than *residual*.

Figure 1 shows the residual plot between the distinctiveness measure and localization confidence. The plot displays the accumulated difference between both data sets normalized to one and provides a quantitative comparison between the plots. It shows how many data points have differences greater than the residual axis i.e., the N axis is the number of data points that have a difference of at least the *residual* axis value. It can be seen that there are not many data points with residual greater than 0.5 (about 20 % of the data).

6.2. **Vision System**

In order to test the distinctiveness measure for the vision system, a real sample environment was constructed. It consisted of 3 types of regions: one with substantial constraints (visually interesting), one with limited constraints (little visual interest) and one with no constraints (not visually interesting), show in figure 4(a). Snapshots of the scene were generated using a pan and tilt camera. The experiment was set up to capture an image, extract the features and pan the camera by .5° for the next

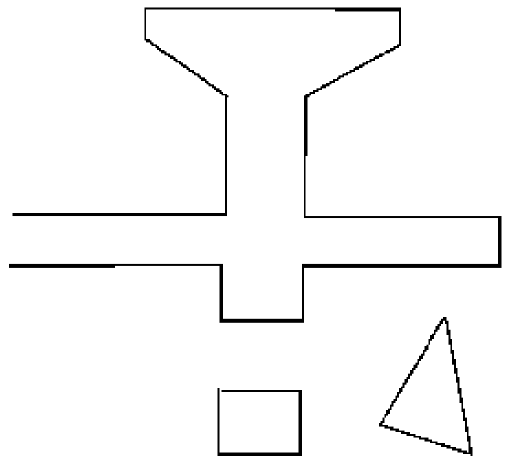
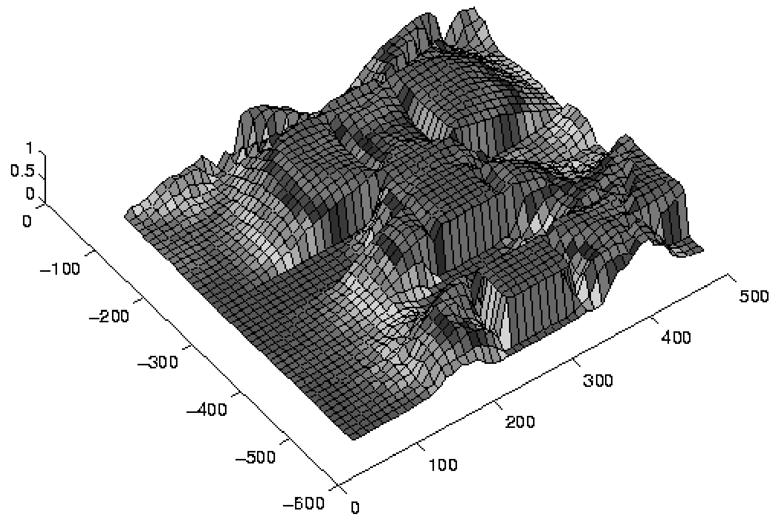


FIGURE 2. A simulated environment with hallways, intersections and bounding regions.

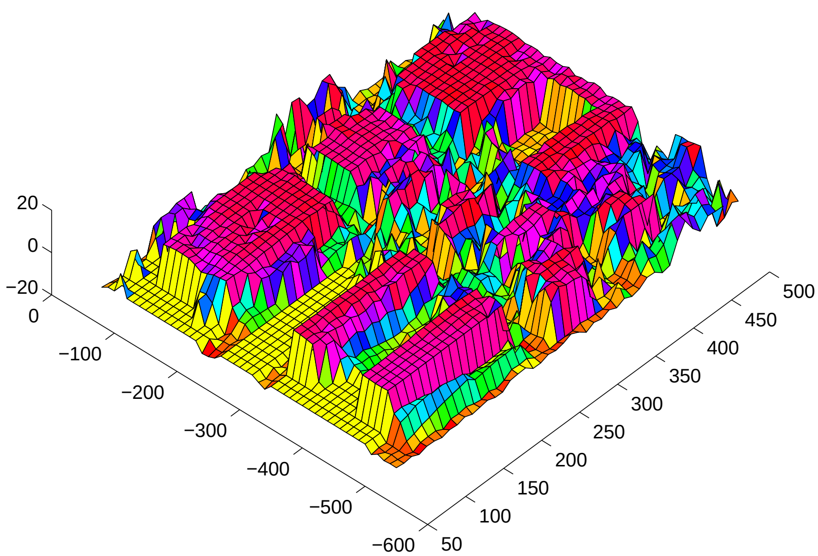
iteration. An 18-D feature space was built using 17 features corresponding to each pan position. The total scene extent had a viewing field of 30° at a distance of $1m$.

The distinctiveness measure (equation 5.21) was employed at each point using a window size ϵ of 10°. Every axis (for a total of 18, 16 features, x and y) of the feature space were normalized to 1 within each window to allow consistent thresholding. The sigmoid function (equation 5.25 parameters were empirically set such that the cutoff c was $1/\sqrt{2}$ and rate m was 3 for a slow transition (cubic sigmoid). That is, features with intensity (I) of 1 and a variation (ΔI) slope greater than $1\sqrt{2}$ are acceptable. For the other sigmoid (equation **??** for the smoothness constraint) the cutoff c was set to .17 (a maximum of 10° slope change) and the cutoff rate m was 3 (cubic sigmoid). The distinctiveness measure for this configuration is shown in figure 4(b). As we move from left to right, it can be seen how the measure is large at the beginning, where the image is visually complex. Then it immediately begins to drop due to the operator capturing the plain white area. The distinctiveness measure remains low within this region and begins to rise when the features at the end are within viewing range.

Although the distinctiveness measure seems to describe what is visually interesting to us as observers, we again must determine if it predicts whether the robot can localize at those areas. Thus, we compute the localization errors by examining the training confidence of the feature space interpolator (the neural network). Where the training confidence is low, localization confidence is also low due to the inability to properly interpolate. If the training confidence is high then so is the localization confidence. For this, we used the previously collected data from each window to separately train the three layer back propagation neural network interpolator. The neural network consisted of 17 input units, for the features, 6 hidden units and 5 output units for the pose. (The pose is decomposed into 5 components for the output



(a)



(b)

FIGURE 3. Sonar based distinctiveness measure 3(a) and localization confidence 3(b) for the simulated environment 2.

units that can later be linearly combined to re-compose the pose.) Training was done using a neural network software package $Xerion$ that was developed at University of Toronto. The training confidence for each window is shown in figure 4(c).

The distinctiveness measure and training confidence show similarities over the sample scene. The distinctiveness measure is high at areas that were interpreted interesting and low at non-interesting areas. The training confidence shows a similar pattern. This validates our ability to predict where areas of the environment provide enough localization information for the vision system. A plot of the residual difference between the localization confidence and the distinctiveness measure is shown in figure 5.

One significant difference for both methods is where the localization confidence exhibits sharp drops while the distinctiveness measure undergoes smooth decays. This is due to the localization techniques that are only accurate within a region of convergence. Once the robot moves beyond that region the solution becomes completely incorrect; there is a narrow mid-ground. The analytic distinctiveness measure, on the other hand, is a smooth continuous function. This attest some of the variational inconsistencies between the plots.

Keeping this in mind, a mapping threshold can be determined by the intersection point of the distinctiveness measure and the confidence cutoff region. Areas where the measure is less than the threshold are not reliable and should not be mapped but areas displaying larger measures are good candidates. Furthermore, the empirical parameters can be determined more accurately by forming an optimization criterion to minimizing the residual plot. Searching the parameter space to minimize the difference between distinctiveness measure and localization confidence can improve results.

6.3. **Mapping with Line Segment Model**

The goal of this experiment is to explore an environment and follow the mapping criteria (equation 3.3) to build a hybrid topological-metric representation (equation 3.1). A large scale simulated environment was built using the robot controller and simulator. A simulated robot explored this environment using a pre-defined set of control strategies extracted from a simple hall following procedure [**35**]. (Although, more complex and autonomous exploration strategies may be used.) The robot was initiated at a starting node and followed the control strategies, while keeping an estimated (x,y) accumulated error. Once the robot inferred it was lost, it attempted to build a line segment model: i.e., an island of reliability using the sonar system. For simplicity, we did not include the vision system for this experiment. Each island was built using a simple node building routine. It consisted of collecting sonar data at four corners of a region within the minimum visibility range (closest object seen) while attempting to constrain the size of the node to cover a square area of $4(\epsilon + \delta)^2$ We only map within the visibility range since the mapping algorithm does not include collision avoidance. (More advanced node building routines can easily be incorporated to the structure.) While collecting data, the robot intermittently calculates the distinctiveness measure to validate the region. If the region is acceptable, the new node is added to the map, including a link to the previous node and the control strategies that were followed.

Figure 6(a) shows the simulated environment and the exploration path and figure 6(b) shows the hybrid topological-metric map. As shown, the robot built nodes at well constrained areas that provide good localization information. Areas such as hallways were not included in the map since they lack in information along one degree of freedom. Note that the robot almost missed an intersection at the top of the map (the last node). This is due to the specular properties of sonar data that were incorporated to the measure and is consistent with localization confidence.

Figure 6(b) shows the line segment models within the nodes of the hybrid topological-metric model. Each node represents a separate local map with its own co-ordinate system, no global reference is used. The spatial arrangement of nodes in the figure are only set for clarity. Links between the nodes in the figure describe the topology and include a compilation of the control strategies used in the exploration step. The exploratory control strategies were compiled in the appropriate reference frame to form the navigation instructions used between neighbouring nodes. They are listed in table 1. Note that if the robot is exploring a cyclical environment, a procedure such as that in [**21**] may be used to identify new nodes from old ones.

Link	Control Strategies
0	Fwrd 660
1	Fwrd 78, Rot 90, Fwrd 360
2	Rot 190, Fwrd 682
3	Rot 190, Fwrd 40, Rot -100, Fwrd 340
4	Rot 90, Fwrd 20, Rot -90, Fwrd 650

TABLE 1. Control strategies for inter-node navigation

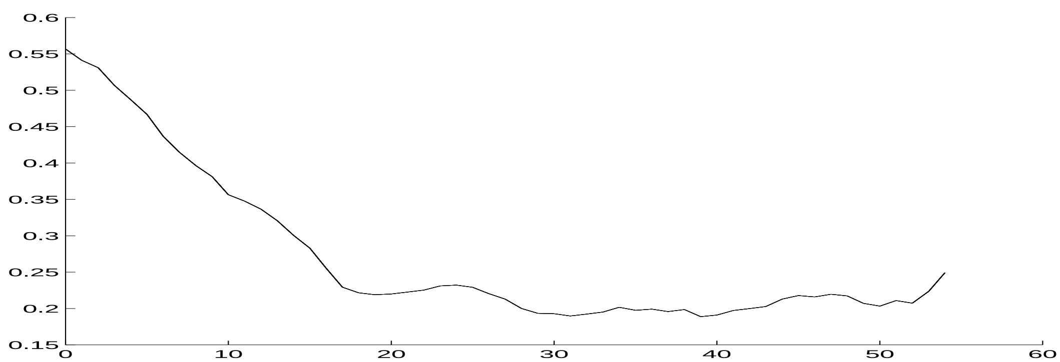
Each control strategy is initiated at coordinates $(x, y, \Theta) = (0, 0, 0)$ in the corresponding reference frame. Therefore, to navigate from node to node, the robot must first localize at $(0, 0, 0)$ then follow the control instructions. These control strategies form a simplistic way to navigate the environment. Although they are not purely qualitative controls, they do form instructions that navigate the robot from one node to the next without the need of *a priori* data gathered at the links. Furthermore, they can be replaced by purely qualitative controls strategies such as *Follow wall to next node* (or such as those listed in [**21**]). Here, sensory criteria can be defined to execute these purely qualitative instructions. For example, the robot can execute a wall following procedure, intermittently comparing sensory data and node data to see if it has reached the next node.

Since rotational error was not included in this simulation, all coordinate frames were parallel. If rotational error is included, then each node would be slightly rotated, the control strategies would vary accordingly and the accumulated error would grow more rapidly. However, the distinctiveness measure used did not include a criterion for rotational degree of freedom. On the rare occasions, it is possible to choose a bad area for angle estimation using the current distinctiveness measure (such as the centre of a round room). Figurer 7(a) and 7(b) shows a topological map of a real environment. The robot was set to extract the first 3 nodes of the hallways near our laboratory. It is easy to see how the robot only recorded local areas that are well constrained and ignored the long hallways. The links are bidirectional since the control strategies can be inverted. Again they originate at $(x, y, \Theta) = (0, 0, 0)$ in each local reference frame. Table 2 shows the links.

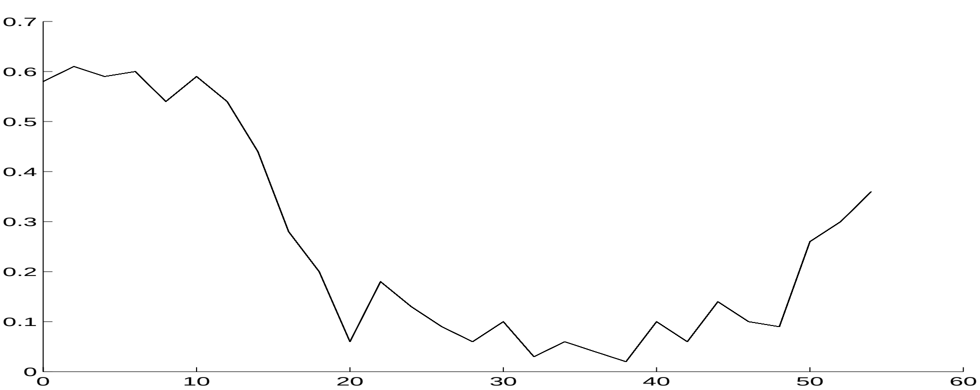
Link	Control Strategies
0	Fwrd 125, Rot 270, Fwrd 652
1	Rot 270, Fwrd 28, Rot 90, Fwrd 832

TABLE 2. Control strategies for inter-node navigation

(a)



(b)



(c)

FIGURE 4. Vision based distinctiveness measure 4(b) and localization confidence 4(c) for a real environment 4(a) with real object, room divider (shading is difficult to observe) and background clutter.

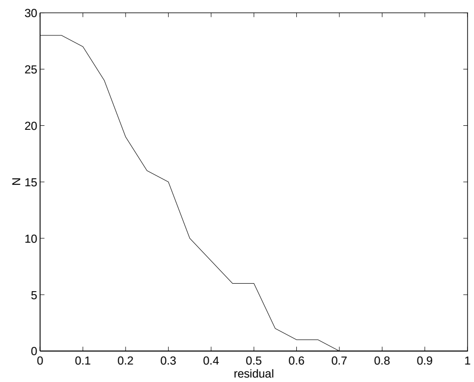
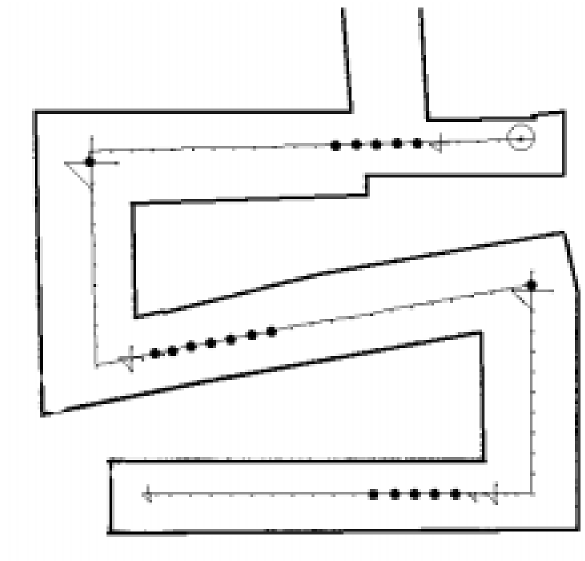
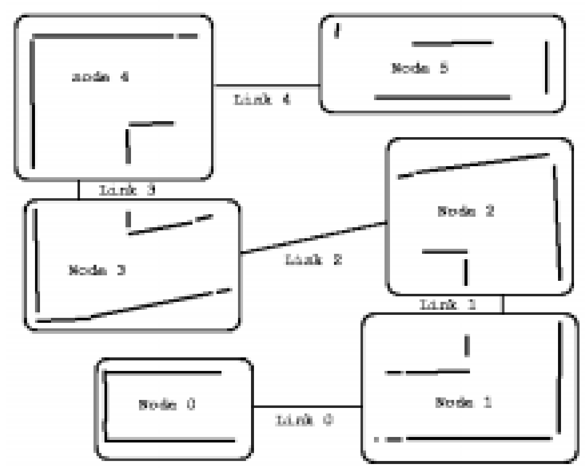


FIGURE 5. Residual plot for figure 4(b) and 4(c), there are N data points with differences greater than $residual$.

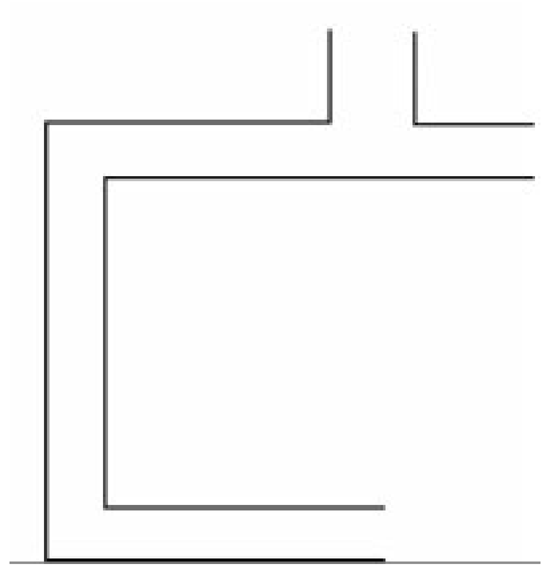


(a)

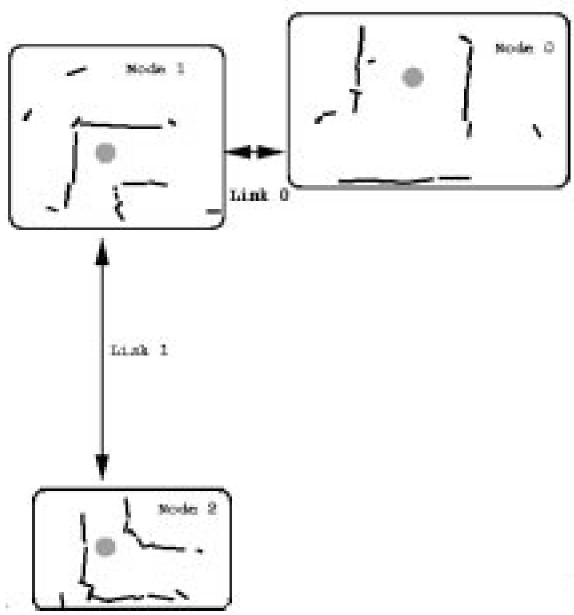


(b)

FIGURE 6. Simulated map and the exploration path 6(a) and the topological-metric map 6(b). Large dots in the exploration path show where the robot attempted to place an island but immediately failed due to low measure. The full star like paths show where the robot had successfully completed to build an island. The partial star like paths show an attempt that partially built a node but was dismissed due to later discovery of low distinctiveness measure.



(a)



(b)

FIGURE 7. A map of a real environment 7(a) and the topological-metric map 7(b). The dots in the topological map show the origin of the local frame.

CHAPTER 7

Conclusion

This thesis describes an important step in the creation of large scale maps that combine both metric and topological knowledge. The main idea was to build a global map by connecting several local metric maps together. We formed a mapping criteria that configured the distribution and connectivity of the local metric maps. Specifically, we describe how the locations of individual localization regions or *islands of reliability* can be selected. Our general concept is illustrated using two specific yet very different types of localization procedure: a sonar system and a vision system. The distinctiveness measure showed consistency with localization confidence, making it a good measure of the utility of environment information for localization. We further examined the mapping criteria using the sonar system with a simulated and a real environment. The results showed that the nodes were connected appropriately and placed at areas where their utility is established. The thesis was put in context to high level mapping goals that require the framework of environment evaluation.

7.1. Future Work

The two sensory and localization systems described in this thesis form an instance of our generic mapping scheme. To improve robustness, other localization methods can be incorporated into the architecture. This would allow a robot to select between many different schemes when building a local island of reliability (a node). Hence, it is possible that an area, where the two (vision/sonar) sensing methods show a low distinctiveness measure, is included in the map due to a high distinctiveness measure of another method. Other localization methods may use information in a better way at that region, even when the same sensory system is used. This opens a broad range of work in developing evaluation criteria and formulating the distinctiveness measure for localization schemes. Each localization scheme can be analysed to formulate the distinctiveness measure as described in this thesis.

Further work can include optimizing the distinctiveness measure by variation of parameters within its formulation to minimize the differences between the predicted measure and the actual localization errors. Evaluating different functional forms of combining both I and ΔI may also improve the measure. The distinctiveness measure operator scale also remains an open problem. An advantage in extending this work is that each localization method can be analysed and optimized separately to be included as an extra component in the architecture.

The mapping architecture itself opens room for much expansion. One main issue is to derive a suitable method for identifying new nodes from old ones. This can be considered as a global localization task within the topological representation. Issues to be considered are problems of detecting visited nodes from new ones, geometrically identical but separate nodes and partially intersecting nodes.

More work can expand on the node building mechanism. In this thesis, we simply collected data from four corners defined by ϵ, unless an obstacle was in the way i.e., map within the minimum visibility. This does not guarantee that the size of the nodes will be large enough so that the robot can reach them. Obstacles may obstruct the node building routine and limit the node size. Thus, a more intelligent routine can be developed to map around obstacles and assure metric mapping of a given size.

The use of more sophisticated exploration strategies can also improve the mapping procedure. An exploration strategy directed by the distinctiveness measure can be developed to explore the environment specifically looking for good regions. A hill climbing technique could be incorporated within the exploration strategy such that the mapping criteria will not be threshold based but will search for neighbourhood optimality of distinctive regions. Future work on navigation can help develop an optimal navigation procedure coupled to our environment representation. With our model, the navigation problem now falls within the topological framework where we can use graph search algorithms.

The topological-metric environment representation described in this thesis provides a framework for developing maps for many purposes. Therefore, it is possible to investigate different mapping goals and develop a distinctiveness measure and the associated algorithm for each. The general approach is to decompose a task into components requiring metric data. Then, to define the node placement criteria by deriving the distinctiveness measure most appropriate for the computational task. The robot can then follow these criteria to construct the nodes and connect them appropriately. It is also possible to investigate a method for combining several task-based maps. For example, combining a navigation map and a painter's map (figure 1) can provide a complete map where the robot can navigate and paint an environment. The problem then becomes how to match and connect the two maps together.

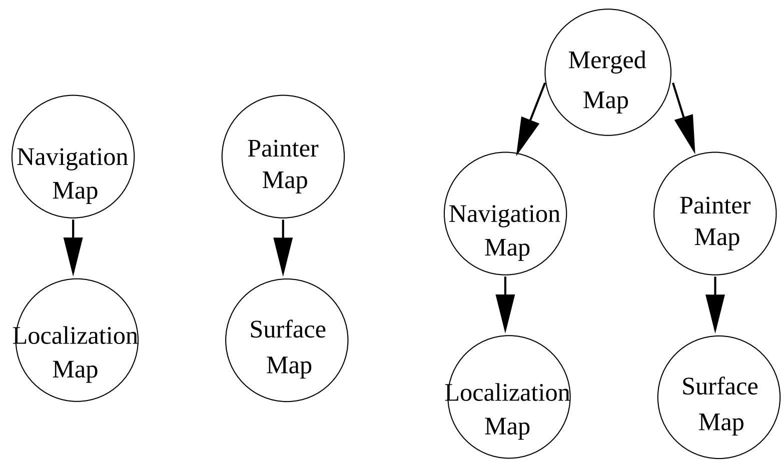


FIGURE 1. An example of 2 task-based maps. A navigation map consisting of connected localization maps and a painter map consisting of connected surface maps (describing the surfaces to be painted). The merged map allows the robot to navigate and paint the environment.

Bibliography

[1] P. MacKenzie and G. Dudek, "Precise positioning using model-based maps", in *IEEE International Conference on Robotics and Automation*, San Diego, California, May 1994, pp. 1615–1621.

[2] G. Dudek, "Environment mapping using multiple abstraction levels", in *Proceedings of the IEEE*, Nov. 1996, vol. 84.

[3] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation", in *AI Journal 99(1)*, 1998, pp. 21–71.

[4] J. Hershberger, "An optimal visibility graph algorithm of yriangulated simple polygons", *Algorithmica*, vol. 4, pp. 141–155, 1989.

[5] N. Rao, S. Iyengar, and G. deSaussure, "The visit problem: visibility graph-based solution", in *1988 Inf. Conf. on Robotics and Automation*, 1988, pp. 1650–1655.

[6] G. Dudek, P. Freedman, and S. Hadjres, "Signatures and extended signatures for topological mapping using graphs", TR-193-231-34, McGill Research Center for Intelligent Machines, October 1992.

[7] G. Dudek, P. Freedman, and S. Hadjres, "Mapping in unknown graph-like worlds", *Journal of Robotic Systems*, vol. 13, no. 8, pp. 539–559, August 1996.

[8] G. Dudek, P. Freedman, and S. Hadjres, "Using local information in a non-local way for mapping graph-like worlds", in *Proceedings of the Thirteenth International Conference on Artificial Intelligence*, Chambery, France, August 1993, pp. 1639–1645, Internation Joint Conf. on Artificial Intelligence Inc.

[9] M. Mataric, "Navigating with a rat brain: A neurobiologically-inspired model for robot spatial representation", in *Proc. 1st International Conf. on Simulation of Adaptive Behaviour*, 1991.

[10] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Using multiple markers in graph exploration", in *Proc. Symposium on Advances in Intelligent Robotics Systems: Conference on Mobile Robotics*, Philadelphia, PA, 1989, International Society for Optical Engineering.

[11] L. Massone G. Sandini and V. Tagliasco, "'Form-Invariant' topological mapping strategy for 2d shape recognition", *CVGIP*, vol. 30, pp. 165–188, 1985.

[12] A. A. Holenstein, M. A. Müller, and E. Badreddin, "Mobile robot localization in a structured environment cluttered with obstacles", in *Proc of the IEEE International Conference on Robotics and Automation*, Nice, France, May 1992, pp. 2576–2581.

[13] A. Curran and K.J. Kyriakopoulos, "Sensor-Based Self-Localization for Wheeled Mobile Robots", in *Proc of the IEEE International Conference on Robotics and Automation*, Atlanta, Georgia, May 1993, pp. 8–13.

[14] G. Dudek and C. Zhang, "Vision based robot localization without explicit object models", in *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, May 1996, pp. 76–82.

[15] Eric Krotkov, "Mobile robot localization using a single image", in *Proceedings 1989 IEEE International Conference on Robotics and Automation*, pp. 978–983. 1989.

[16] J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons", *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, June 1991.

[17] F. Lu and E. Milios, "Optimal global pose estimation for consistent sensor data registration", in *IEEE International Conference on Robotics and Automation*, 1995, pp. 93–100.

[18] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty", *The international journal of Robotics Research*, vol. 5(4), pp. 56–68, Winter 1986.

[19] W. Yeap, "Towards a computational theory of cognitive maps", *Artificial Intelligence*, vol. 32, pp. 297–360, 1988.

[20] U. Nemzow, "Animal and robot navigation", in *Robotics and Autonomous Systems*, July 1995, vol. 15, pp. 71–81.

[21] B. Kuipers and Y-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations", *Robotics and Autonomous Systems*, vol. 8, pp. 47–63, 1991.

[22] P. Whaite and F. P. Ferrie, "From uncertainty to visual exploration", in *Proceedings of the 3rd International Conference on Computer Vision*, Tokyo, Japan, November 1990.

[23] A. Witkin, M. Kass, and D. Terzopoulos, "Physically based modeling for vision and graphics", in *Proceedings of the Image Understanding Workshop*, 1988, pp. 254–278.

[24] S. Lange, L. Korba, and A. Wong, "Characterizing and modelling a sonar ring", in *Proc. SPIE Mobile Robotics IV*, Philadelphia, 1989, pp. 291–304.

[25] D. Wilkes, G. Dudek, M. Jenkin, and E. Milios, "Modelling sonar range sensors", in *Advances in machine vision: strategies and applications*, C. Archibald and E. Petriu, Eds. World Scientific Press, Singapore, 1992.

[26] J. K. Hackett and M. Shah, "Multi-sensor fusion: a perspective", in *Proceedings 1990 IEEE International Conference on Robotics and Automation*, pp. 1324–1330. 1990.

[27] S. Lang and A. Wong, "Building geometric world models with graph synthesis for sensor fusion in mobile robots", *Computational Intelligence*, vol. 6, pp. 91–107, 1990.

[28] H. Moravec, "Sensor fusion in certainty grids for mobile robots", *AI Magazine*, pp. 61–74, Summer 1988.

[29] S. Shafer, A. Stentz, and C. Thorpe, "An architecture for sensor fusion in a mobile robot", in *IEEE Int. Conf. on Robotics and Automation*, 1986, pp. 2002–2011.

[30] G. Dudek, M. Jenkin. E. Milios, and D. Wilkes, "On building and navigating with a globally topological but locally metric map", in *Proc. 3rd ECPD Int. Conf. on Advanced Robotics, Intelligent Automation and Active Systems*, Bremen, Germany, 1997, pp. 132–136.

[31] R. De Almeida and C. Melin, "Exploration of unknown environments by a mobile robot", *Intelligent Autonomous Systems 2*, pp. 715–725, 1989.

[32] N. Roy and G. Dudek, "Learning to rendezvous during multi-agent exploration", in *Proc. 6th European Workshop on Learning Robots*, Brighton, England, Aug. 1997.

[33] I. Rekleitis, G. Dudek, and E. Milios, "Multi-robot exploration of an unknown environment, efficiently reducing the odometry error", in *Int. Joint Conf. on A.I. (see also CIM Tech. Report TR-CIM-97-01)*, Aug. 1997.

[34] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Robotic exploration as graph construction", *Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 859–865, December 1991.

[35] P. Ciaravola, "Maze exploration", Tech. Rep., Centre for Intelligent Machines, McGill University, 1996.

[36] G. Dudek and J. K. Tsotsos, "Shape representation and recognition from curvature", in *Proceedings of the 1991 Conference on Computer Vision and Pattern Recognition*, Maui, Hawaii, June 1991, pp. 35–41, IEEE.

[37] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics", *Autonomous Robots*, vol. 3, pp. 375–397, 1996.

[38] I. Park and J. R. Kender, "Topological direction-giving and visual navigation in large environments", *Artificial Intelligence*, vol. 78, no. 1-2, pp. 355–395, October 1995.

[39] E. Bourque and G. Dudek, "Viewpoint selection – an autonomous robotic system for virtual environment creation", in *IEEE International Conference on Intelligent Robotic Systems*, 1998.

APPENDIX A

Implementation

The architecture of the map building routine consisted of several components running in parallel. At the top level lies the main map building module which consists of computing the distinctiveness measure, building and connecting the nodes following our mapping criteria. The main mapping module communicates with five other processes:

- robot controller/simulator,
- robot,
- exploration module,
- vision system server,
- sonar system server.

It directly communicates with a robot controller/simulator developed in our labs, which opens a connection to a real or simulated robot. It also connects to an exploration module and the two model based mapping modules (sonar and vision system packages converted to servers) which also connect to the robot controller/simulator. Figure 1 shows the flow chart for these modules and Figure 2 shows the modules running.

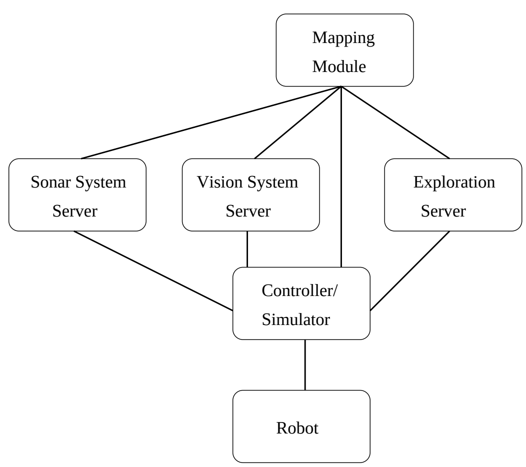


FIGURE 1. Flow chart of software modules.

These modules were all written in the C programming language along with some scripts, written in Perl, for data analysis and conversions.

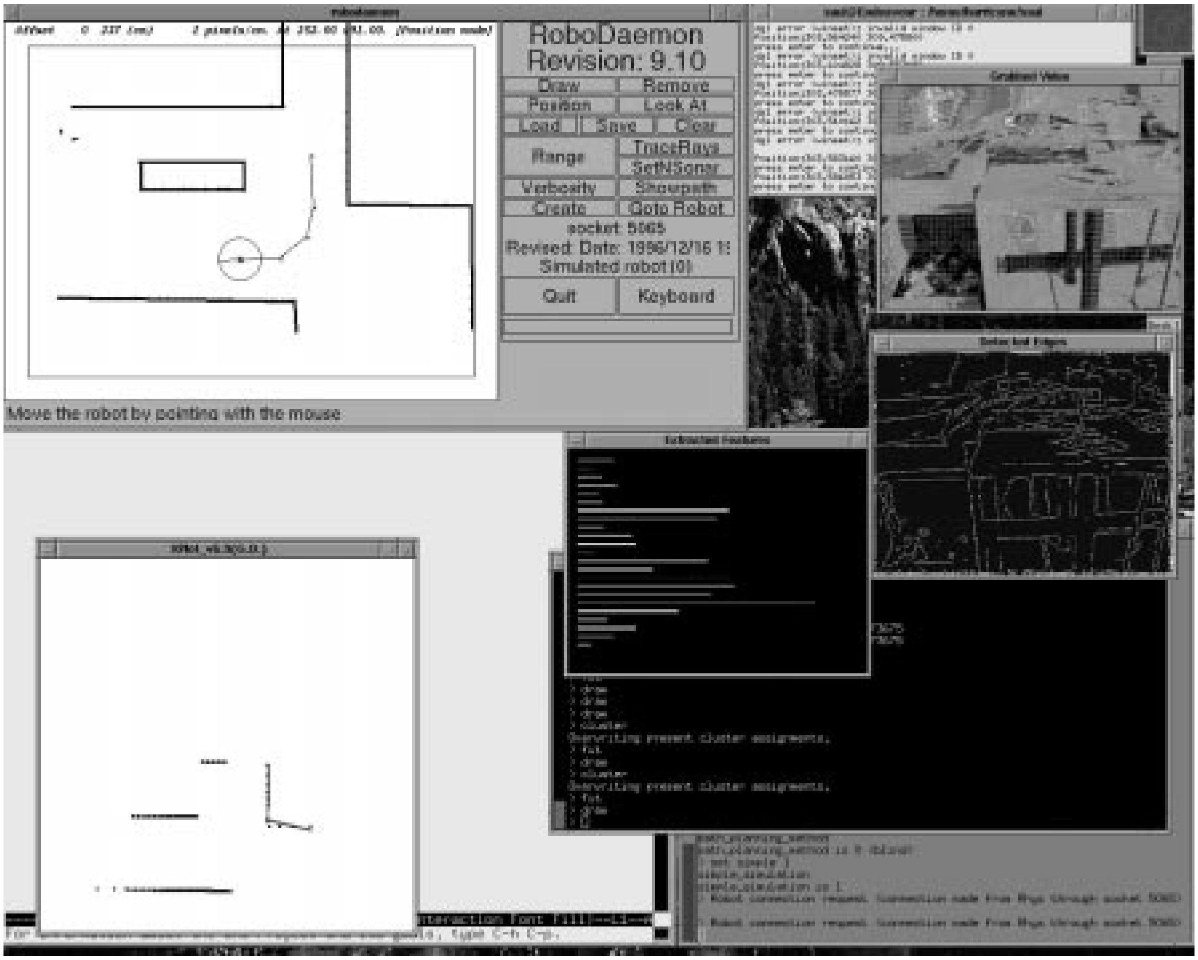


FIGURE 2. This screen shot shows the software modules running together. On the top left is the robot controller with a socket connection to the other modules. The bottom left window shows the output of the sonar system module after executing the line fitting algorithm (in the dark xterm window). The three upper to middle right windows show the vision system module, one window shows the captured image, another window shows the edge map and the last window shows a histogram of the extracted feature values.

APPENDIX B

Sigmoid Function

The sigmoid function is a function that provides a gradual change from a low plateau to a high one. It is used often as a a weighed normalizing function. It was shown in chapter 4 that it can also be used to describe the characteristics of sonar signal degradation (although, we used an exponential decay). The parameters of the function control the transition point (c) and the transition rate (m). Figure 1 show a plot of the high pass function and figure 2 shows the low pass function. We use this function to provide a consistent threshold based measure based on the parameters. For example, for the vision system, we considered that the slope between good features and the position should be about 45°, so we set the cutoff value of c accordingly. This was chosen such that desired features are ones that vary at least at that rate over the operators scale where all axis are normalized. If the axis were not normalized, a slope of 45° over the operator scale would be meaningless. Thus we must vary the parameters with respect to some normalizing factor and a desired target result. Similarly, the numerical result of the integral for the vector influence (equation 5.15) is dependent on the quantization of the seen line segment, i.e., the sensory resolution. We set the value of c according to a minimum line length seen at a certain distance, normalizing with the sensory resolution. If only a single line segment is seen that is further away or smaller than the measure would lie below the cutoff. Thus the meaning of these values can be portrayed by re-mapping them through the normalizer and setting the parameters accordingly.

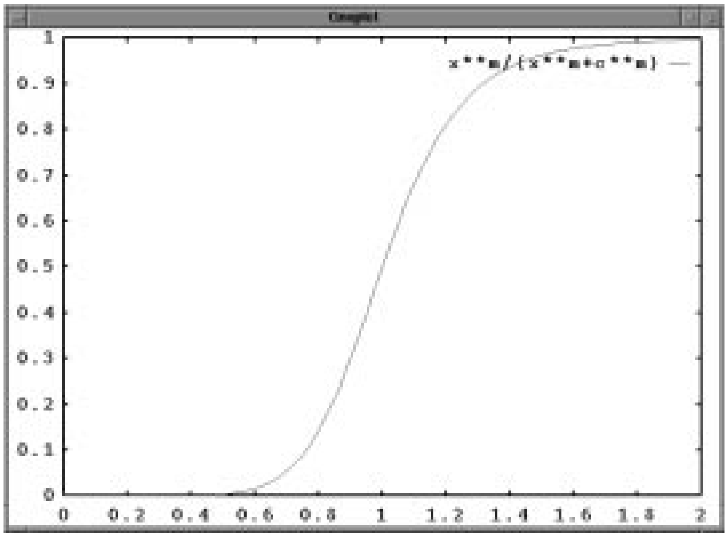


FIGURE 1. Sigmoid function passing large values: $x^m/(x^m + c^m)$ *with* $m = 7$ *and* $c = 1$. The vertical axis shows the function and the horizontal axis shows x.

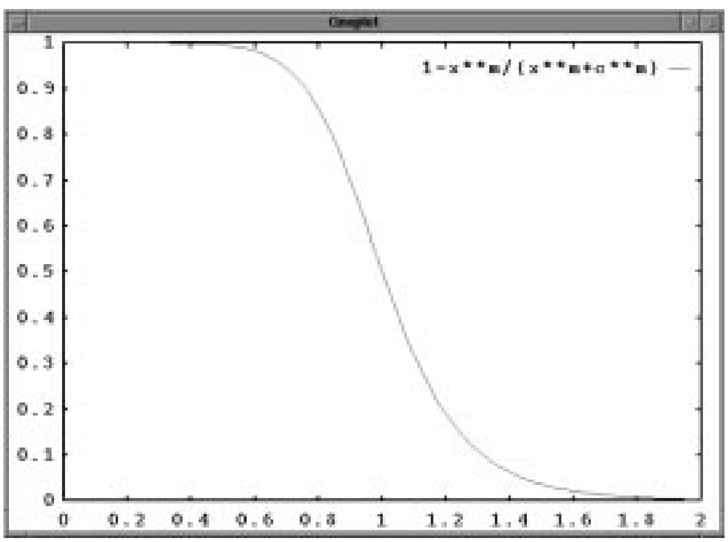


FIGURE 2. Sigmoid function passing low values: $1 - x^m/(x^m + c^m)$ *with* $m = 7$ *and* $c = 1$.The vertical axis shows the function and the horizontal axis shows x.