Universidade de Aveiro Departamento de Engenharia Mecânica
2008

CAETANO FILIPE COSTA DE NORONHA FERREIRA

NAVEGAÇÃO AUTÓNOMA E LOCALIZAÇÃO EM TEMPO REAL PARA UM ROBOT MÓVEL COM PERCEPÇÃO MULTI-SENSORIAL

AUTONOMOUS NAVIGATION AND MULTI-SENSORIAL REAL-TIME LOCALIZATION FOR A MOBILE ROBOT

Universidade de Aveiro Departamento de Engenharia Mecânica
2008

CAETANO FILIPE COSTA DE NORONHA FERREIRA

NAVEGAÇÃO AUTÓNOMA E LOCALIZAÇÃO EM TEMPO REAL PARA UM ROBOT MÓVEL COM PERCEPÇÃO MULTI-SENSORIAL

AUTONOMOUS NAVIGATION AND MULTI-SENSORIAL REAL-TIME LOCALIZATION FOR A MOBILE ROBOT

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Mecânica, realizada sob a orientação científica do Dr. Vitor Manuel Ferreira dos Santos, Professor Associado do Departamento de Engenharia Mecânica da Universidade de Aveiro da Universidade de Aveiro e de Jorge Manuel Miranda Dias, Professor Associado do Departamento de Engenharia Electrotécnica e Computadores da Universidade de Coimbra.

Apoio financeiro da Universidade de Aveiro no ambito duma bolsa de Doutoramento cedida através do Instituto de Investigação.

Apoio financeiro para a impressão do documento da tese obtido da Fundação para a Ciência e a Tecnologia no âmbito do Programa de Apoio a Execução Gráfica de Teses de Doutoramento (PAEGTD)

aos Satya, Adélia e Álvaro.

o júri

presidente

Prof. Dr. Aníbal Guimarães da Costa
professor catedrático da Universidade de Aveiro

Prof. Dr. José Joaquim de Almeida Grácio
professor catedrático da Universidade de Aveiro

Prof. Dr. Pedro Manuel Urbano de Almeida Lima
professor associado com agregação do Instituto Superior Técnico da Universidade Técnica de Lisboa

Prof. Dr. Urbano José Carreira Nunes
professor associado com agregação da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Prof. Dr. Jorge Manuel Miranda Dias
professor associado da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

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

Prof. Dr. António Manuel Ferreira Mendes Lopes
professor auxiliar da Faculdade de Engenharia da Universidade do Porto

Prof. Dr. Jorge Augusto Fernandes Ferreira
professor auxiliar da Universidade de Aveiro

acknowledgements

The conception and execution of any task is never the result of the effort of a lone individual. It is instead, fruit of the explicit and implicit contributions of many persons. This thesis is not an exception to this rule. It is therefore with great satisfaction that I thank all who have in some or the other way acted to allow me to complete this work.
Firstly, I wish to thank Prof. Vítor Santos, my academic supervisor and scientific guide over the course of the master’s and the phd program. I thank him for the faith that he reposed in me, for the patience and support that he showed me throughout the course of my association with him and for his cracking-of-the-whip and personal effort in this final phase, without which, the end of this chapter in my life would have been further delayed.

I thank Prof. Jorge Dias, my co-supervisor for the important insights and critical evaluations that he provided at various critical phases of my work. I also thank him for the support that he provided me in the form of a research fellowship that enabled me to wrap-up this work.

I thank Prof. José Grácio, for the invitation to enroll in the master’s program at Aveiro; at the time an unplanned event for me, and one that kicked off my association with this department, university and city. I also thank him for his efforts in arranging the necessary funding in the form of a Phd fellowship from the University of Aveiro.

Finally, I thank my family and the friends who supported me over this period for showing understanding and patience even when news was not forthcoming from me and when my spirits were not at their brightest. I shall remember all of them individually for the company that they provided me and for the care that I received.

palavras-chave

Navegação autónoma, localização, integração multi-sensorial, dados binários, cadeias de Markov, alinhamento de sequências.

resumo

O principio por detrás da proposta desta tese é a navegação de ambientes utilizando uma sequência de instruções condicionadas nas observações feitas pelo robô. Esta sequência é denominada como uma 'missão de navegação'. A interacção com um robô através de missões permitirá uma interface mais eficaz com humanos e a navegação de ambientes de maior escala e duma forma mais simplificada. No entanto, esta abordagem abre problemas novos no que diz respeito à forma como os dados sensoriais devem ser representados e utilizados. Neste trabalho representações binárias foram introduzidas para facilitar a integração dos dados multi-sensoriais, a dimensionalidade da qual foi reduzida através da utilização de Misturas de Distribuições de tipo Bernoulli. Foi também aplicada a técnica de cadeias de Markov ocultas (Hidden Markov Models), que contou com o desenvolvimento e a utilização dum modelo de cadeia de Markov original, esta que consegue explorar a informação contextual da sequência da missão. Uma aplicação que surgiu da aplicação do método de localização foi a criação de representações topologicas do ambiente sem ter que previamente recorrer à criação de mapas geométricos. Outras contribuições incluem a aplicação de métodos para a extracção de propriedades locais em imagens e o desenvolvimento de propriedades extraídas a partir de varrimentos dum medidor de distancia laser.

keywords

Autonomous navigation, localization, multi-sensorial integration, binary features, Markov chains, hidden Markov models, sequence matching.

abstract

This thesis evaluates the requisites for the specification of mobile robot 'Missions' for navigation within environments that are typically used by human beings. The principal idea behind the proposal of this thesis was to allow localization and navigation by providing a sequence of instructions, the execution of each instruction being conditional on the expected sensor data. This approach to navigation is expected to lead to new applications which will include the autonomous navigation of environments of very large scale. It is also expected to lead to a more intuitive interaction between mobile robots and humans. However, the concept of the navigation Mission opens up new problems namely in the way in which the sequence of instructions and the expected observations are to be represented.

To solve this problem, binary features were used to integrate observations from multiple sensors, the dimensionality of which was reduced by modelling the binary data as a Finite Mixture Model comprised of Bernoulli distributions. Another original contribution was the modification of the Markov Chains used in Hidden Markov Models to enable the use of the sequential context in which the expected observations are specified in the navigation Mission. The localization method that was developed enabled the direct creation of a topological representation of an environment without recourse to an intermediate geometric map. Other contributions include developments that were made in the characterisation of images through the application of local features and of laser range scans through the creation of original features based on the scan contour and free-area properties.

Contents

List of Figures

Nomenclature

α_c The mixture component coefficient for component c.

α_{reg} The weighting proportion of the motion model estimate for the prior probability of the next place-recognition.

λ The parameter set, $\langle N, M, \{\pi_i\}, \{a_{ij}\}, \{b_i(n)\}\rangle$, of the Hidden Markov Model.

$\mathcal{L}$ The likelihood of a [conditional] event.

$\mathcal{V}$ The FIM, or the complete set of views/vectors as collected during the Environment-Familiarisation stage.

π The initial probability distribution over the hidden states of the Hidden Markov Model.

π_i The initial probability distribution for the hidden state i of the Hidden Markov Model.

$\prod$ Represents the product operator.

$\sum_{k=1}^{K}$ Represents the sum operator with the index k varying from 1 to K.

Θ Represents the complete parameters of the Mixture model.

Θ^* The Mixture Model parameters obtained after terminating the EM algorithm.

Θ_c A single component c of the Mixture Model.

a_{ij} The probability of transiting from [hidden] state i to state j in the Hidden Markov Model.

$b_i(n)$ The observation or emission probability for the symbol b_i at the place n within the Hidden Markov Model.

$b_{i,j}(O_t)$ The observation or emission probability for the symbol O_t at the transition i, j within the Hidden Markov Model.

C a constant, represents the total number of components in the BMM.

c an index, represents a component in the BMM.

F A single [named] feature in a view/vector.

$H(V_k/K)$ The conditional information between the distribution of V_k, given that K is known.

$I(V_k : K)$ The mutual information between the distribution of V_k and K.

j An index, employed to denote a particular feature, j.

K A constant, typically indicating the number of Views in the original sampled Reference Sequence.

k An index, employed to denote a particular view, k.

k^* The estimated place as obtained by applying the Max-likelihood criteria to the Belief over the indices of the Reference Sequence.

K_s A constant, indicating the number of Views in the original sampled Reference Sequence, s.

k_s An index, employed to denote a particular view from Reference Sequence s, k_s.

M The number of [hidden] states in the Hidden Markov Model.

N The number of [visible] Observations/symbols in the Hidden Markov Model.

$P(V_{obs}/\Theta_c)$ The similarity between the observation V_{obs} and the component c of the Mixture model.

$P(V_k/V_{obs})$ The probability of the currently observed view/vector V_{obs} being matched to View/vector V_k in $\mathcal{V}$, the Reference Sequence.

$P(V_{obs}/\Theta)$ The probability distribution of the observation V_{obs} over all the components of the Mixture model.

T The total number of Observations in the Hidden Markov Model.

t The variable indicating the Observation number in the Hidden Markov Model, t varies from {0. . .T}.

V_k A single view/vector with an index k within the FIM, $\mathcal{V}$, composed of multi-sensorial features.

V_{obs} The currently visible View/vector with the features that are currently observable.

Z Hidden or incomplete data in a Mixture Mode.

z_k The vector from Matrix Z corresponding to the View/vector V_k.

z_kc The value of the Matrix Z corresponding to the View/vector V_k and component c.

${}_jV_k$ Value of the Feature 'j' in the View/vector V_k of the Reference Sequence.

${}_jV_{obs}$ Value of the Feature 'j' in the observed View/vector V_{obs}.

${}_j\Theta_c$ The value corresponding to Feature 'j' of Bernoulli Mixture Model Component 'c'.

#traj length of trajectory, number of views in the trajectory.

Camera #1 Camera looking forward in direction of motion.

Camera #2 Camera looking laterally to one side of robot.

LRF SICK Laser Range Finder facing forward in direction of motion.

BMM Bernoulli Mixture Model.

cost_j cost_j is a penalty term for feature j with values between 0 and 1.

EM Expectation Maximisation method as applied to evaluate the parameters of the BMM.

H(S) The entropy of the random variable S.

log() Represents the log operator.

traj trajectory, section of a topological path

Chapter 1

Introduction

Over the last two decades there has been enormous progress in the field of robot navigation, with techniques being borrowed from fields as diverse as biology, signal processing and data-mining, to name a few, to solve the problems of Map-building and robot localisation. With improvements in the computational power that is available on modern PCs and reductions in the cost and in the size of components, the past few years have seen an acceleration in the attempts to apply the lessons learned within University research laboratories to solve real world problems involving autonomous vehicles. Such attempts involve assembling a host of technologies to create complex albeit reliable systems. This endeavour is expected to continue over the foreseeable future with an ever greater emphasis on the simplification of the machine interface with humans and with other machines.

Autonomous Robot Navigation is an ongoing and still a difficult problem to solve. Although advances in the science of localization and mapping accompanied by great improvements in computing hardware have yielded satisfactory results for small to medium indoor environments, substantial challenges remain.

According to [Filliat 03] basic map-based navigation depends on three processes

- **Map-learning**: the process of memorizing the data acquired by the robot during exploration in a suitable representation.

- **Localisation**: the process of deriving the current position of the robot within the map.

- **Path planning**: the process of choosing a course of action to reach a goal, given the current position. The definition of 'path planning' varies with the time horizon and the nature of the motion or action that must be planned for.

Amongst the authoritative reviews of the process of creation of maps, Sebastian Thrun's [Thrun 02a] 'Robotic Mapping: A Survey' counts as a still valid and relevant introduction. As Thrun mentions, most of the successful state of the art methods in Localisation and Mapping are probabilistic in nature, albeit some methods are less overt than others in the representation of the uncertainties that affect sensing and robot control. In a similar vein Fox et al. [Fox 03], attempt to classify the well known approaches in mobile robot navigation focusing on the differences in the models that are used to represent the environment and the position of the robot. As a result of the legacy of ultrasound range sensors, many of the more successful maps still are actually probabilistic representations of free-space boundaries.

This thesis applies a selection of techniques that are borrowed from other disciplines to address the problems facing robot localization. These techniques are meant to aid mobile robots in the navigation of environments that are habitually frequented by human beings.

Localisation methods place the robot at that place (or at more than one place) in the environment which best explains its current sensor data. The choice of the Localisation method is usually a function of the type of map that is used.

We have addressed a particular type of localisation problem: positioning the robot somewhere along a known path. The information that the robot is given about its environment, the map, is given to it in sequential form, corresponding to the things that it will sense if it moves down the correct path.

By keeping the planning and the execution of robot motion out of the purview of the work described in this thesis, focus has been retained on the creation of a representation of the environment and on the localisation of the robot within this representation.

Our idea for robot localisation and the outline of the proposed solution are described in the next section. This description of the problem will be followed by a section that attempts to provide another way of looking at maps that have been described in the literature. This review focuses on maps from the perspective of map *events*. Section 1.3 will provide a backdrop for a more detailed description of the problem that this thesis addresses. The section 1.4 introduces the main ideas that will be tackled in the remainder of the document and presents a brief layout of the thesis document.

1.1 Problem Description

Imagine a situation in which our friend Juliana is journeying, by car, to a meeting at a house in the country. Juliana stops at a petrol station to ask for directions. The clerk at the petrol station provides Juliana with a sequence of descriptions of the environment that she will encounter as she progresses from a known landmark, say, the petrol station, to her final goal, as in Fig. 1.1. The clerk also provides Juliana with a sequence of instructions that Juliana must execute, instructions that are concomitant on the things that she will see as she successfully makes her way to her goal, the house.

The clerk tells Juliana that a little down the road, after passing the petrol station, she will come across a road junction. The road junction, is like many that Juliana might have previously come across. Our guide does tell her that this intersection is special because a bridge will be visible from it and because there will be some conspicuous trees that will be visible to the her right-hand side. Juliana is instructed to turn right at this junction and drive a few hundred meters till she arrives at her destination, at the house.

This sequential way of providing an agent with the expected observations against which the agent can localize itself and perform the required actions is associated with the execution of a definite mission or program. It does not require that the representation of the complete environment be provided at once and the actions that must be performed at each step are dependent on the observations.

Applications for such a way of representing the environment would include, for example, interfaces of social robots that interact with humans, receiving and relaying environment maps in a way that is more intuitive and efficacious to the completion of a mission. If Juliana came to be substituted with a computer that guides and runs the vehicle, a petrol station clerk (or a virtual

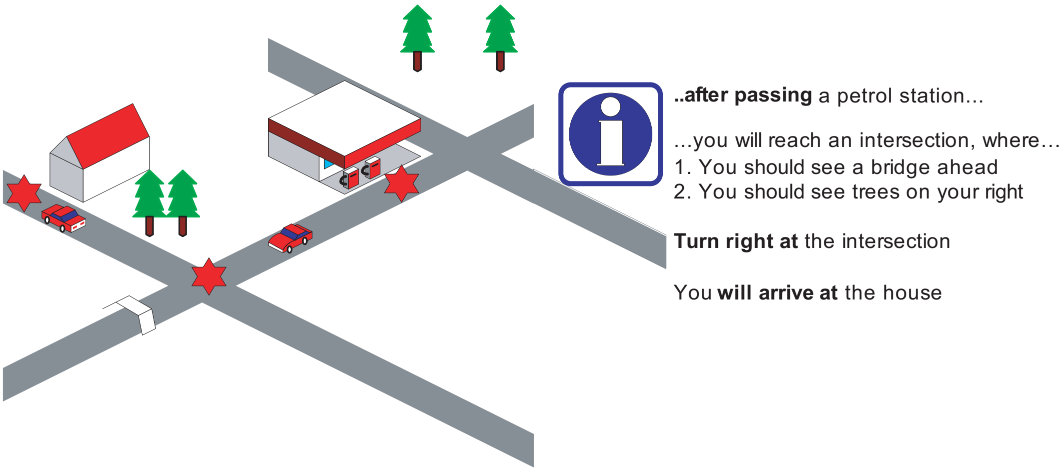


Figure 1.1: A sequential conveying of expected observations might be sufficient to successfully navigate along the Path. This is similar to the way instructions are given to people to allow them to complete a task or mission, in this case, arrive at the house.

clerk as the case may be) might still be able to provide a sequential description of a path and the instructions that are required to complete the mission.

Guiding a robot by providing only a sequence of expected sensor data opens a whole new set of possible applications that could involve a more intuitive interaction with humans. Such a concept, however, opens up new problems, namely in the question of how the sequence is to be represented and how data association is to take place.

This thesis seeks to contribute to the discussion of how to best exploit the sequential description of a environment to effectively perform localisation.

1.2 Taxonomy of Maps used for Robot Navigation

The map of the London Underground, seen in Fig. 1.2, is one of the most recognizable maps around. This map is instantly associated with the Metropolitan rail road and it has been adopted as a template by urban rail transport planners the world over. The purpose of such a map is to help travellers plan their journeys. The map is remarkable for the ease with which passengers can identify stations lying on the same line, plan transfers to other lines and evaluate the cost of the complete trip.

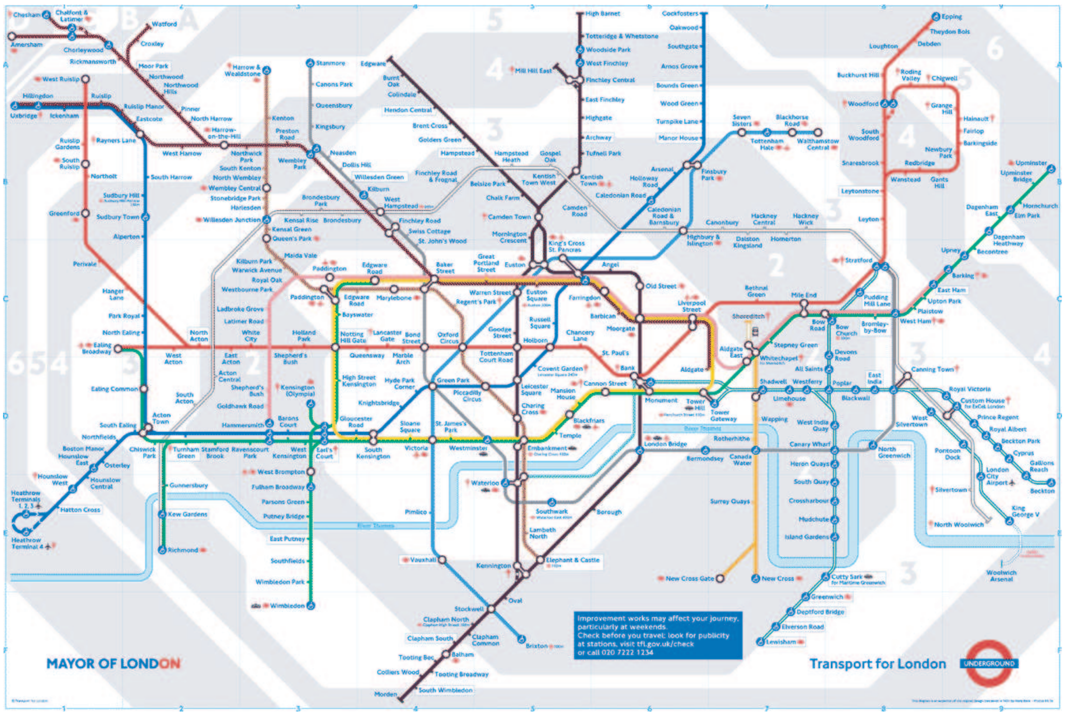


Figure 1.2: The 'official' map of the London Underground.

In the Fig. 1.2, along each line, the map indicates the names of the stations and the order of appearance of the stations. At most, the map maintains the proportions of the distances between stations lying on the same line. It has information about the bus and train services that users can access upon exiting a station. This map does not seek to accurately represent the distances between stations and the overall layout of the stations with respect to each other. This *topological* map of the underground does not reveal how far, geographically, a station on one line is from any other station on the same or on a different line. Also, the map provides information only on events occurring at certain finite number of places, the stations. Information about places lying between stations is non-existent.

There are applications for which this map is not the most appropriate representation of the tube network. For example, to a first time visitor to London, the Fig. 1.2 might not be sufficiently informative to decide whether it is worth just walking between a pair of stations, rather than wait for the next train that is delayed. The same map might, like-wise, not be an appropriate map for maintenance and emergency services.

Figure 1.3: The geographical layout of the London Underground.

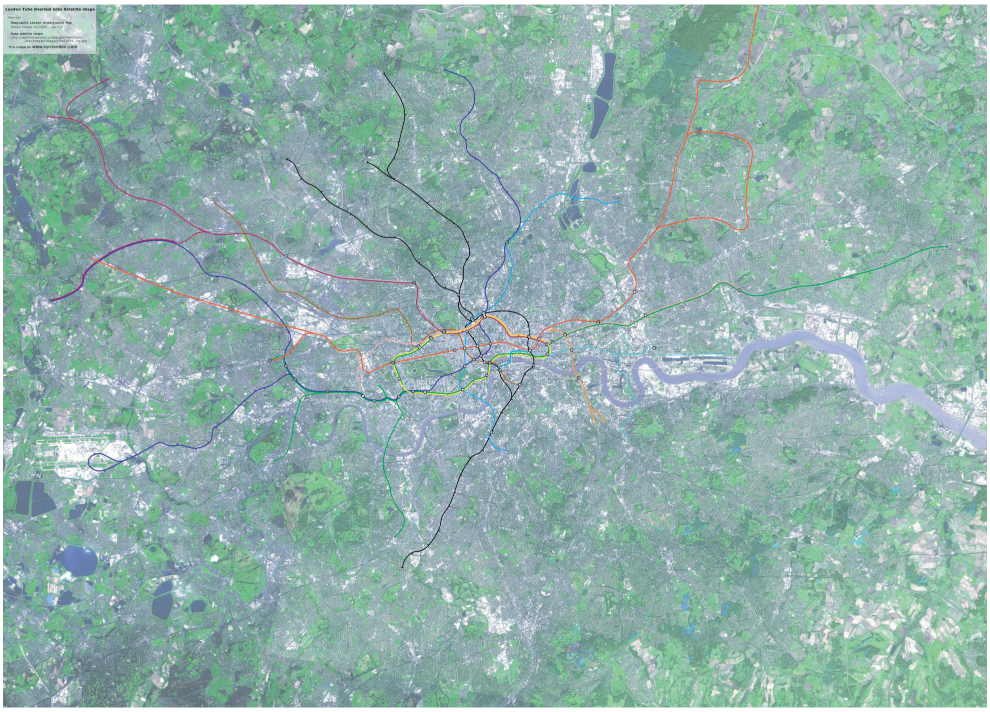


Figure 1.4: The actual geographical map of the London Underground with Surface features included.

For these users, a geographical map of the tube, like the one shown in Fig. 1.3, would be far more helpful. Such a *metric* map lays out the stations and the lines according to their geographical coordinates. This a map allows comparisons to be made of the distance between every pair of stations and of points lying on the lines, in between the stations. Additional information, such as ground-level features can be added to this new geographical map can also be represented in the same coordinate system, as depicted in Fig. 1.4.

The geographically accurate maps seen in Figs. 1.3 and 1.4 include large regions that are not covered by the underground network (and hence are not useful to commuters). The additional information has also resulted in lines that are difficult to follow, visually, and the accompanying text is sometimes small or uncomfortably positioned.

The additional information provided by the geometric or metric map improves the usability of the map by non-passengers. However, it is not the most useful map for commuters who simply want to execute a journey from station 'A' to station 'B' within the London Underground system. This example is an demonstration of a situation involving localisation where sequential, topological representation of the environment is more useful than precise metric positioning.

1.2.1 'Maps represent sensory and motion events...'

From the point of view of the theory of probability, maps represent events that can occur when a robot interacts with the environment. This interaction typically includes the *motion* of the robot and the *sensing* that can take place at any position. In other words, a map is a set of features, from *a sample space of sensor events* and a set of positions which the robot can occupy, *a sample space of positions*. Brief definitions of Sets, Sample Spaces and Events are included below.

Set: A set, A, is an aggregate or collection of objects. The members of the set A are called the elements of A, or $\in A$ and, if x is not an element of A, $x \notin A$...' [Hines 90]

Experiments and Sample Spaces: Sensing the environment is the execution of an experiment whose outcome cannot be predicted with certainty, being thus denoted as a random experiment. Despite the fact that the outcomes cannot be predicted with certainty, it is still possible to identify the set of possible outcomes, known as the *Sensory Sample Space* of the experiment.

Depending on the type of outcomes, Sample Spaces can be classified as discrete and continuous. Robot motion can also be viewed as another experiment wherein the outcome of a particular robot motion cannot be predicted with certainty.

Events: An event is the outcome of an experiment. Static maps contain events that can be sensed by the complement of sensors that the robot possesses. Sensory events can be simple, such as detecting whether a particular region in the environment is free or occupied, or complex, e.g. the detection of a particular 2D pattern from observed data that is acquired at multiple positions. Motion events are very summarily addressed in the course of this document.

Over the sections 1.2.2 through 1.2.4, three types of environment representations are reviewed with a view to scrutinize the differences that exist at the level of the representation of sensed events. In this review, relatively little attention has been given to methods that are used to create the maps and to the events that populate the map. For a discussion along these lines [Thrun 02a] provides an useful comparison.

An important point of discussion is the idea of Robot-centered versus World-centered representations of the environment. At the time of building maps of the environment most methods transform what the robot has sensed (which, by definition, is the robot-centered representation) into a world-centered representations, using the estimate of the position of the robot in the world-centered representation. Range sensors, for example, return a sample of distances to the nearest obstacles along some finite number of directions, thus returning robot-centered representations of the robot's neighbourhood.

Events on a geometric or topological map are usually termed, sometimes interchangeably, as landmarks, objects or features. These landmarks and features are sensory events the robot can identify, segment and recognize. The selection of the landmarks or features is a very relevant topic while discussing maps and a wide range of features and landmarks have been used within such maps. In robot platforms equipped with cameras local image features such as edges, corners, have been used. Topological maps have also included parametric groups of 2D and 3D point clouds produced by range finders and stereo cameras. The events might be all inserted into the same space of events or in different spaces.

1.2.2 Geometric Maps

World-centered or geo-referenced representations lead to maps such as the geometrical map of the Underground shown in Fig. 1.4 where all the features are inserted in the same coordinate system. Geometric maps allow the calculation of the distance that separates any two features that are included in the map. When used by robots equipped with range scanning sensors, geometric maps frequently include 'distance' events: i.e. the features that are represented must have definite coordinates and the robot must posses sensors that can measure the distance to the source coordinates of these features. In fact, as the first robots were equipped mainly with sonar and laser range sensors, geometric maps were used early on [Elfes 87].

Since geometric maps are, by their very nature, world-centered distributions, the robot-centered data must be translated and rotated appropriately in order to be incorporated into the world-centered representation. This incorporation procedure is usually performed by creating a physical model of the behaviour of the sensor in the environment and by making convenient assumptions about how to associate the different measurements.

In order to account for sensor noise and to allow an easy update of the map when new data becomes available, there occurred a shift towards discrete representations, to the so-called occupancy maps or evidence grids. Given the nature of range sensors in use, the limited computing and memory available and the need to map ever larger indoor spaces, the maps started out as a deterministic 2-D representations of the extents of the sensed open space. These were quickly improved to incorporate the uncertainty in sensor readings and the resultant inconsistencies that are bound to occur when the robot passes through the same stretch of environment multiple times.

Representing only the free-space boundaries in a geometric map or a single type of feature within a map presents both advantages and drawbacks. The advantages are the lower level of complexity required to register and incorporate the data from a single sensor. This is an important advantage since the sensor models may be very different and since the data from each sensor-centered representation must later be incorporated into a common map. The drawbacks of such maps are that the reduced variety of local regions in the maps results in a poor capability to determine how to register the sensor-centered data gathered at any instant with the map. This results in an increase in situations that suffer from the well-known problem of aliasing and from the general difficulty of data-association between the sensor-centered measurements and the world-centered representations. Additionally, in an effort to make the localisation of the robot tractable, non-linear motion equations are frequently linearised resulting in a gradual accumulation of an unbounded positioning error.

The Expectation Maximisation (EM) technique and other incremental mapping techniques have been applied to reduce the severity of these problems. Further evolution of the method might improve on the techniques required to perform registration. Other methods might also add to the capability of adding new features to the map and to the capacity to create, maintain and use larger maps effectively.

Geometric Maps enforce a consistency on the distances between every pair of places by defining a single coordinate system. If a sensory event must be represented in the geometric map, it must first be possible to represent it within the coordinate system of the map. This step ensures that the joint probability distribution of the events will be valid and that putting together sensed events from different sensors will result in consistent probability distributions. Therefore, data fusion requires the registration of the various sensors into the same single coordinate system within which all the features will be represented.

1.2.3 Topological Maps

Topological maps, on the other hand, *need not* have a consistent coordinate system in which all the map sensory events are represented. Topological mapping methods might use information about free-space boundaries or obstacle locations and in some cases local metric maps might be created and used in order to communicate results or interact with operators, e.g. [Silver 04] or to improve the place-recognition capability of the robot. The latter approach allows the use of place-recognition capabilities developed within the context of geometric maps, to create compact graph representations of the overall environment, appropriate for the creation of maps for larger environment [Thrun 98]. While geometric mapping is still popular, some researchers increasingly feel the need to adopt mapping techniques for what are called Large-scale environments.

Large-scale environments are defined as environments that cannot be observed all at once, using the sensors that the robot is equipped with [Savelli 05]. For this the drawback attributed above to geometric map-building methods must be addressed.

One such popular approach is to use a metric map created using range sensors and reduce it to a one-dimensional representation using a Generalised Voronoi Diagram. Such a method, while not having to store the range measurements within the topological map, results in the storage of a 'high-level' feature which can be easily sensed from the range scan sensor data [Choset 01], [Silver 04].

Increasingly, alternative methods have been developed to create Topological maps for navigation and Localisation. An important class of such methods are termed as appearance-based or view-based methods. In [Ulrich 00], Ulrich and Nourbakhsh, histograms are used as features to mark places in the environment. A distance metric based on the Jeffrey divergence between histograms is utilized as a metric and a set of adjacency Maps are utilised to account for robot motion. A generation of new methods for creating topological maps succeeded the seminal work called Topological SLAM by Choset and Nagatani [Choset 01].

Since topological maps typically contain sparser information than metric maps, they must contain better data-association methods, either in the form of a richer map-feature set or in the form of better global localisation algorithms.

In the context of a topological map for mobile robots, places are distinctive points in the environment that the robot can occupy or has previously occupied, at least once. The topological map can also specify whether a transition between a pair of places is possible and can include additional information regarding these possible transitions.

1.2.4 Hybrid Maps

Hybrid approaches to map-building are usually employed in applications that require the characteristics and advantages of both geometric and topological maps.

By definition, hybrid mapping would include a geometric representation and a topological representation, that is usually linked in some way to the geometric one. One of the earliest and

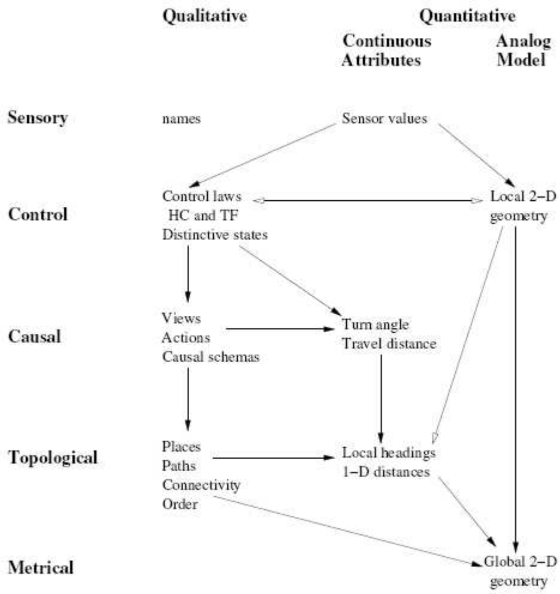


Figure 1.5: A Schematic of the Spatial Semantic Hierarchy [Kuipers 00]. Closed-headed arrows represent dependencies, open-headed arrows represent potential information flow without dependency.

possibly one of the most well known approaches in this category is the Spatial Semantic Hierarchy or the SSH developed by Ben Kuipers [Kuipers 00]. The SSH, shown in Fig. 1.5, is described as 'a model of knowledge of large-scale space consisting of multiple interacting representations, both qualitative and quantitative'. The representation of the environment is maintained in the form of a hierarchy of maps each of which allows some abstraction of the perception and interaction of the robot with the environment. These hierarchical'levels' include a Causal layer, a Control layer, a Topological layer and a possible (if enough information is available) a high-level Metrical layer. The advantages gained from using SSH or similar hierarchical model of representations is that incomplete information or uncertainty in the information is handled in different forms depending on which particular Localisation or navigation problem is to be solved. Local metric maps help to perform place recognition, [middle-level] topological maps help create consistent maps in the face of challenges such as loop-closing problems, and the global metric maps maintain an overall consistency in the global position of the robot. Savelli and Kuipers [Savelli 05], utilise a probabilistic modelling of motions behaviours to move the SSH from a rule based method to a graph-based topological map framework based on Bayesian networks.

There are also attempts to utilize graph-based approaches to solve particular problems that appear at the time of creation of metric maps. Methods such as [Folkesson 04], use graphical methods to maintain hypothesis for map expansion and closure, i.e. graph-like methods are used to maintain multiple map hypothesis of the main map which is geometrical.

There are works that enhance the applicability of metric maps and the ability of users to interact with these such as representing individual objects. In [Limketkai 05], Limketkai et. al. store the representation of objects (some of which might also be used by persons) using a technique called Random Markov networks.

As in the case described in the section on topological maps, in certain mobile robot systems, there is a need to maintain local geometrical maps around certain regions. The reason for maintaining these maps however is not to perform place recognition, but to allow better and faster planning of trajectories and re utilization of the map by robots equipped with different sensor configurations [Konolige 04].

1.3 Representing Places sequentially along a Path

A Mission, from the Latin Missum, refers to persons sent or appointed to perform any service; a delegation; an embassy. In the context of mobile robot navigation, a mission can be defined as *an ordered series of descriptions and instructions given to a robot that will take it from one place in the environment to another*.

In the context of this work, a Mission consists of a sequential description of situations and motion behaviours that the robot will sense along the way. We refer to the sequence of descriptions as the Reference Sequence. When a sequence of behaviours or actions are needed to move the robot from a point, A, to another, B, are added to the Reference Sequence, a robot Mission is defined.

The Mission is communicated to the robot in the form of a string of motion behaviours that are concomitant on the expected observation. Initial development of the specification of such mission strings was based on previously developed work [Santos 01] and is described in

Appendix C. The lack of previous experience with semantic representations and has led us to concentrate on the sequential representation of sensor information.

Thus, instead of manually creating a mission string and communicating it to the robot, the robot is driven along a path during an Environment or Path familiarisation phase. Since the motion of the robot has not been integrated into the representation of the environment, we will normally deal only with the Reference Sequence in the remainder of the thesis. It is our opinion that motion behaviours could be inserted into this Reference Sequence at a later stage to create a Mission if the capability of the system to recognise places and localise itself is verified.

Two distinct questions must be answered for a Reference Sequence to be specified in the form of a sequential description of the environment. These are:

1. How are observations represented in the Reference Sequence?

2. How can the sequential description of observations improve the localisation within the Reference Sequence?

These are questions for which, answers will be sought, over the course of the next two chapters. The exploitation of information, that is implicit in the sequential description of sensor data, that results in better place recognition is the key element that differentiates the approach described in this work from other map-building approaches in the literature.

1.3.1 Some Previous Work

The genesis of our attempt to navigate along a path by localising the robot within a Reference Sequence lies in previous work. A mission programming tool was developed for use on the robot platform at the Mobile Robotics and Automation Laboratory (LAR) of the Department of Mechanical Engineering, of the University of Aveiro [Santos 01]. This tool, the Language for Autonomous Mission Planning (LAMP) [Santos 01] could be used by an operator to set up a mission, using a qualitative description of the topology of the environment. The kernel feature of LAMP is that a robot mission may be planned and executed by utilizing the approximate position and layout of entities to trigger the beginning and the end of individual phases of the mission.

Although odometry is utilised to execute individual stages of the mission, the initialisation and termination of each stage resets the odometry effectively removing any accumulated error. Individual stages are set up either as closed loop feedback or as open loops. This method partially obviates the necessity of the robot knowing where exactly it is in the environment. LAMP still required a boot-strapping localisation procedure.

LAMP allows the environment to be described in terms of obstacles that the robot will sense around itself as it executes the various stages of its mission. The robot motions that can be included in a LAMP mission include: wall following, obstacle avoidance, in-place turning, executing a short trajectory in open-loop mode and crossing narrow openings. An illustrative mission comprising of 11 mission stages and the plan view of the result of each stage is described in Fig.1.6.

1. MOVP MV 2 0 PS 2 USL 500 SEN 1
2. MOVE LV 0 AV 10 ANL 90
3. MOVP MV 20 USL 1000 SEN 1
4. MOVE LV 0 AV 10 ANL 90
5. MOVE LV 20 DIL 1500
6. CROSS
7. MOVE LV 10 USL 3 00 SEN 1
8. MOVE AV 2 0 ANL 90
9. MOVP MV 2 0 PS 2 USL 500 SEN 1
10. MOVE AV 20 ANL 90
11. MOVP MV 20 PS 1 USL 300 SEN 1

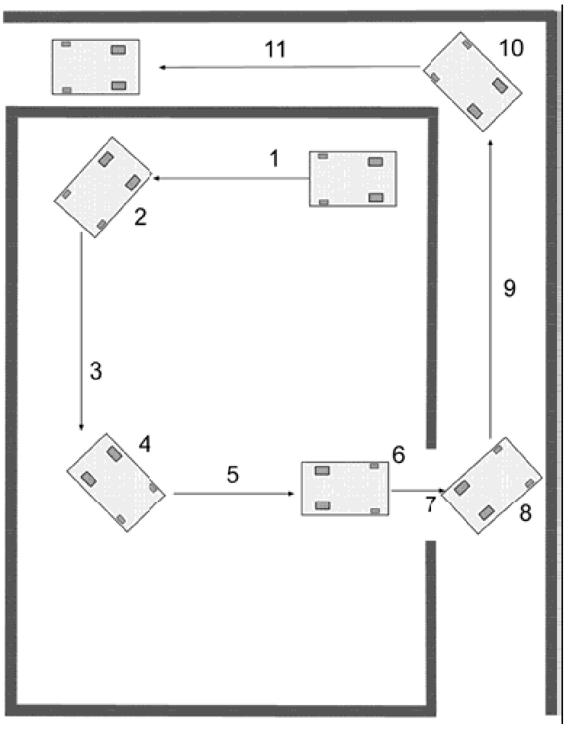


Figure 1.6: A navigation mission with associated LAMP code.

LAMP served to demonstrate the concept that the sequential description of the expected perception could be used to trigger the start and end of a sequence of motion behaviours that could propel the robot from the beginning to the end of a mission. It does suffer, however, from limited sensory capabilities that, in turn, limit its application in real-world environments.

LAMP handles its single feature in a trivial manner with the sonar sensors indicating the presence or absence of an obstacle. Its inability to handle more than one type of feature means that it cannot be applied to more conventional robot platform navigating robustly within a real-world environment. An expansion of the method to include multiple sensors would have to include more sophisticated methods to integrate the different features.

The application of LAMP-like methods would be much enhanced if realistic methods to create LAMP missions were developed and if vision and range scan features were used within the missions to provide robust robot localisation capabilities.

1.3.2 Individual Place Recognition and the temporal context of Places

Consider a hypothetical robot equipped with a special sensor that allows it to identify certain environmental features. With the aid of this sensor our robot can identify doors, corners and walls. To keep this exposition simple, it is assumed that the robot can sense these landmarks only if they lie close to it.

Supposing our robot is lead on a sightseeing tour of a building. At regular intervals of time our robot looks around, identifies the landmarks that are in sight and records the sighting in a table. The sequential description of what the robot found on the path has been represented in Table 1.1 and a plan-map of a section of the environment would appear similar to Fig. 1.7.

If the description of the observations seen from places 1 through 10 in the environment described in Table 1.1, were provided sequentially to a robot we would denote it as the Reference Sequence. Each place is associated with particular landmarks and with a motion behaviour. By correctly identifying the place at which it finds itself, the robot can recover the actions that are required to get it to the next way point/place. If the actions that led the robot from each place to the next place were added to the Reference Sequence we would obtain a robot mission. A mission that would allow the robot to move from the starting place 1 to the final place 10 would have been created.

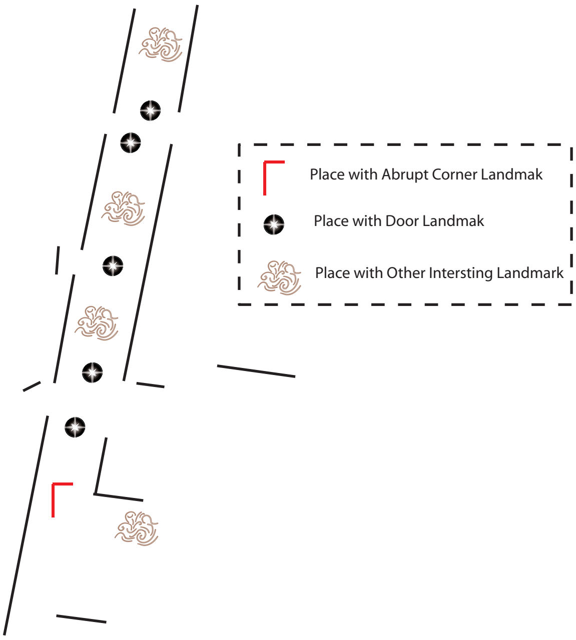


Figure 1.7: The figure illustrates a sequence of places, in a corridor, where each place is recognised because of the landmarks perceived at that place.

Table 1.1: Table of Places included in the Reference Sequence for the path depicted in Fig. 1.7.

	1	2	3	4	5	6	7	8	9	10
Corner Landmark	0	1	0	0	0	0	0	0	0	0
Door landmark	0	0	1	1	0	1	0	1	1	0
Wall landmark	1	0	0	0	1	0	1	0	0	1

Our hypothetical robot has the ability to distinguish between a limited number of landmarks. As a result it is not always possible to accurately distinguish one place from every other places. In particular, using a single observation, it is impossible to distinguish between the various ’door’ places, because the doors are ambiguous landmarks (and the same applies to the walls). In the context of the mobile robotic Localisation problem, this challenge is commonly referred to as ’perceptual aliasing’.

There is another problem, one that has not been described in this example, the problem of ’Scene Variability’, in which the same landmark appears differently at different times, either because of changing observation conditions, noisy sensor readings or as a result of the robot moving within dynamic environment (moving objects or changing nature of the objects).

The problems of perceptual aliasing and scene variability can be diminished by using a larger set of landmarks or by using combinations of landmarks, such that each combination of landmarks is unique. Unless a sufficient number of unique landmarks are available, it is going to be difficult to increase the size of the map whilst simultaneously ensuring reliable place recognition.

A logical improvement to using a *single observation* of landmarks would be to observe the order in which *multiple observations* of landmarks are made by the robot. In practice, the cost of implementing algorithms to identify different and unique landmarks is high, and the twin problems of scene variability and place aliasing is so common, that virtually all localisation algorithms make inferences from observing multiple observations. This accumulation of information is done in two ways.

The most common method is to utilise an estimation filter to accumulate the information gained by observing a sequence of observations. At the time of each observation, the evidence gathered over the multiple previous observations are added to the current observation in the form of a prior probability.

The other approach is to gather the combinations of features that are viewed while travelling along a path to create larger, and hopefully more unique combinations of landmarks. Such an approach can be utilised all the time as in the topological approach Kuipers [Kuipers 02] or occasionally, when there is a greater risk of place aliasing, such as when applied to the problem of loop closure while map building [Newman 06].

1.3.3 Modelling the Sequence as a Topological Path

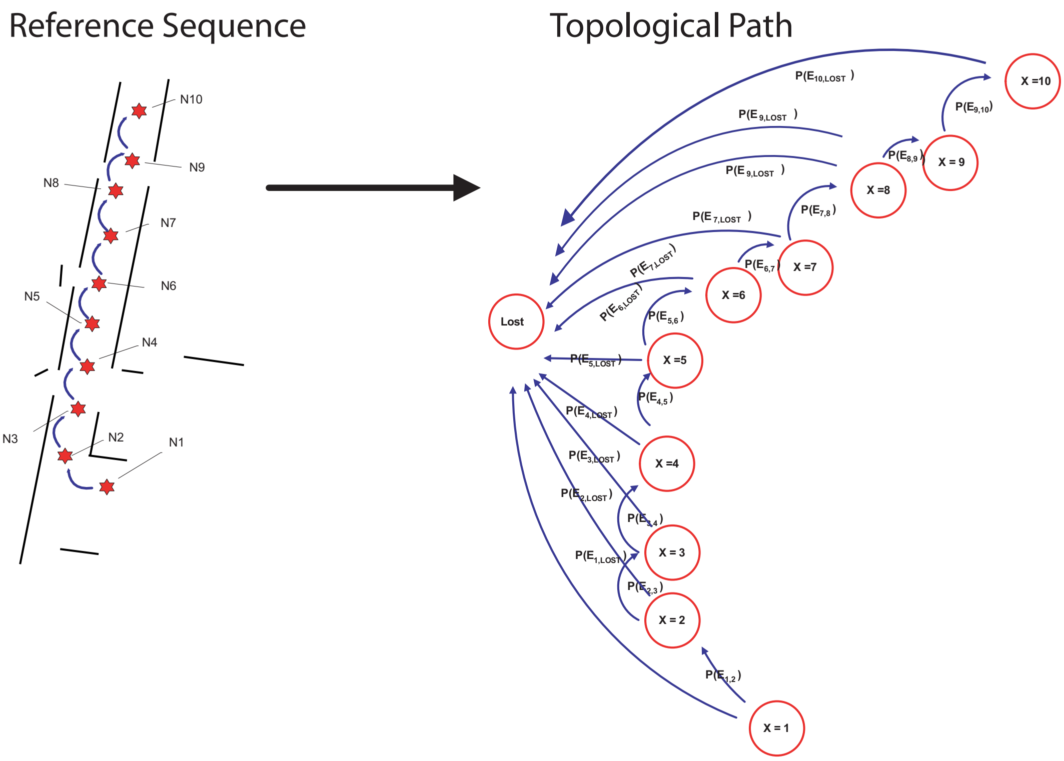


Figure 1.8: Representation of the Topological path as a Graph of Places arranged sequentially. The graph itself is created from a sequence of sequentially obtained sensor views.

Within the scope of the discussion from the last sections, a graph of places denotes an ordered set of places, along a path in the environment, at each of which the robot perception is defined. Importantly, the places that are represented in these graphs are not identified by their physical position in a common coordinate system. Some perception or landmark is associated with each place in the Topological path as seen in Fig. 1.8.

Localising the robot in the path and consequently in the graph of places that represents the Topological Path entails estimating the place that is currently occupied by the robot.

1.3.4 The Place of the 'Lost' Robot

The Reference Sequence is created by sampling the environment, which is a discrete procedure. It is not possible to take observations continuously and often it is not very feasible to sample the environment at very high rates. Doing so would seriously limit the size of the environment that can be usefully represented.

In this work, a definition of the *Place of the Lost Robot* (no relationship with Isaac Asimov's Little Lost Robot), or the $Lost\_Place$ was found to be useful. The $Lost\_Place$ represents the positions in the environment at which the robot does not know what combination of landmarks can be seen.

In the graph of places, the $Lost\_Place$, has also been added to the previously defined places. The Place of the Lost Robot is useful in two distinct situations: in the first, the robot has simply no recollection of a particular combination of features currently in view because of the problem of sampling. In the second situation, an unintended maneuver, or a substantial change in the environment results in the robot not recognising a place that is represented in the Reference Sequence.

Thus, the $Lost\_Place$ represents **all** the possible places that the robot might encounter in the environment that are not represented in the graph of places and the locations in the environment that fall outside this path.

In the literature, recognition at places that are difficult to identify is often done by attempting some sort of back-tracking. In this thesis, behaviours to recover from a failure to localise are not implemented and the robot simply declares itself as lost. In other words, when a robot is at a $Lost\_Place$ it is expected to eventually stop and declare itself as being 'lost'.

1.3.5 A hierarchy of representations

The proposal of the Reference Sequence as a representation of a topological path can be viewed in the context of map building algorithms that have been described in the literature. As a representation of the environment, the Reference Sequence contains the least amount of information.

The hybrid geometric and topological maps presented by Thrun in [Thrun 02b] have been used as a comparison to our approach. The different amounts of information that are maintained by different representations can be viewed as giving rise to a hierarchy as seen in Fig. 1.9. As seen in the figure, the higher we move up the hierarchy, lesser is the information that is

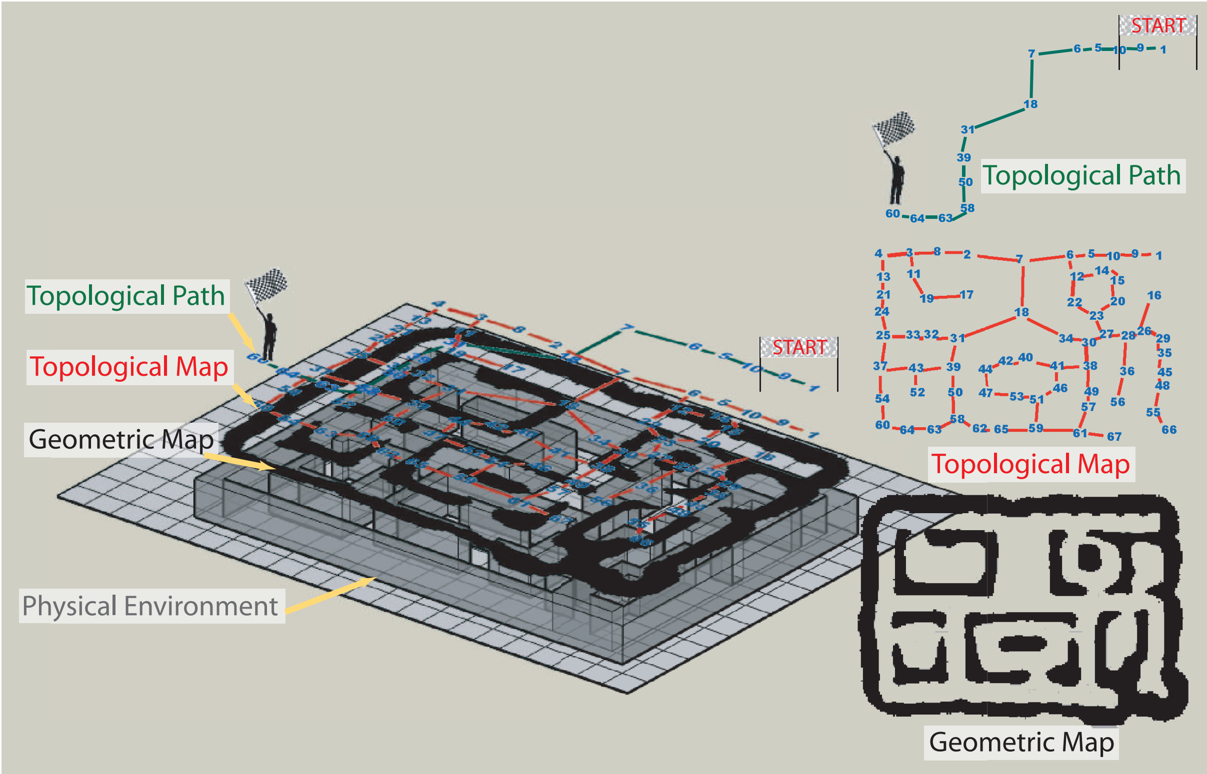


Figure 1.9: Navigating the environment along a topological path can be looked at as a representation containing lower information as compared to a complete geometric or even a complete topological representation of the physical environment.

maintained.

The geometric map attempts to represent all the information that can be extracted by the sensors within a single joint probability distribution. This joint probability distribution is broken up resulting in the graph structure of the topological map where the perception is stored at the various nodes of the graph. The Reference Sequence maintains information pertaining to only a single path in the global topological map.

1.4 Proposed Organization of this Thesis

This thesis document is chapter-wise organised according to three successive objectives of 1) designing an un-biased place recognition method, 2) making use of the sequential context in which place recognition takes place and 3) modification of these techniques to improve the usability of the methods.

When the robot navigates the environment using the Reference Sequence, a number of sources of uncertainty must be dealt with. There is uncertainty in the process of detection of the landmarks (due to noise and different perception conditions) and there is uncertainty in the accuracy of the motion along the Reference Sequence (due to dynamic environments). As compared to the term ’landmarks’, ’features’ are a more generalized form of representing the properties of current scene, since it avoids certain semantic difficulties that are associated with the use of the term, Landmark. A feature is taken to mean any artifact or property that can be extracted from a sensor view, be it from a Laser Range Scan or from an image or any other sensor data. This definition opens the way to using multiple features extracted from data from different sensors. Any feature that is chosen must be relatively robust to changes in the conditions of the observation. This thesis has made use of robust features that are extracted from images and from Laser Range scans.

Chapter 2 presents an algorithm to perform context-independent, non-informative prior, place recognition using a Bernoulli Mixture Model. In chapter 3, place recognition results are shown to improve when the currently observed view is compared with each view in Reference Sequence in the context of appearance with respect previous observations. In chapter 4, the problem of creating topological maps from individual Reference Sequences (topological paths) is considered. Chapter 5 begins with the description of the complete system as implemented on two robots and lays out the architecture of the system and a brief description of the applications and libraries that were developed. Finally, chapter 6 lists the main contributions of this thesis and the resultant articles that were accepted at peer-reviewed conferences and journals. The close of this last chapter also provides an opportunity to present issues that were addressed but which could be solved in useful time and which now indicate the direction for future work.

In the remainder of this section, the proposal for place recognition using sequences of views from multiple sensors is described.

1.4.1 Place Recognition using a single View

A key issue of using a method based on features is the choice of the frame of reference. The fact that features must be extracted from a number of different sensors and pooled into a single set of multi-sensorial feature forces us to choose between choosing a single common frame of reference or ignoring the ordering and position information that the features possess within the frame of reference. The features extracted from an image or from a Laser range scan, each have their frame of reference. The ordering of the features in the coordinate system of the frame of reference of the sensor can provide substantial information to aid the matching of a view.

Because the features to be integrated are very different from each other, a registration or sensor calibration procedure would be required to integrate each additional type of feature. Additionally, it would be impossible to integrate features that provide bearing/distance from the robot with other features that have no such obvious property. Also, because features from different sensors will not be inserted within a common coordinate system, the ordering information within a single view has been completely ignored while integrating the different features. If a need arises to include the information about the ordering between two or more features, this ordering can be explicitly included in the form of additional features.

The term 'Perception' is used to signify both, the act of perceiving or sensing the world and also the result of this sensing. Given some sort of environment representation, robot Localisation involves using the 'Current View' to help determine the likely 'Current Position'.

A schematic for such place recognition from a Single View is shown in Fig. 1.10. This figure depicts an Environment or Path Familiarisation stage at left in which a sequence of views are collected together. Each view is an index that refers to a particular place along the path taken by the robot during the Environment Familiarisation stage. Place recognition or localisation involves the recovery of this index by the comparison of the current observation with all the views in the Reference Sequence.

The current view is compared with each of the views in the Reference Sequence through a comparison of the features that appear in the views, using an appropriate algorithm. In order to be able to use the information from the sequence as efficiently as possible, it is important

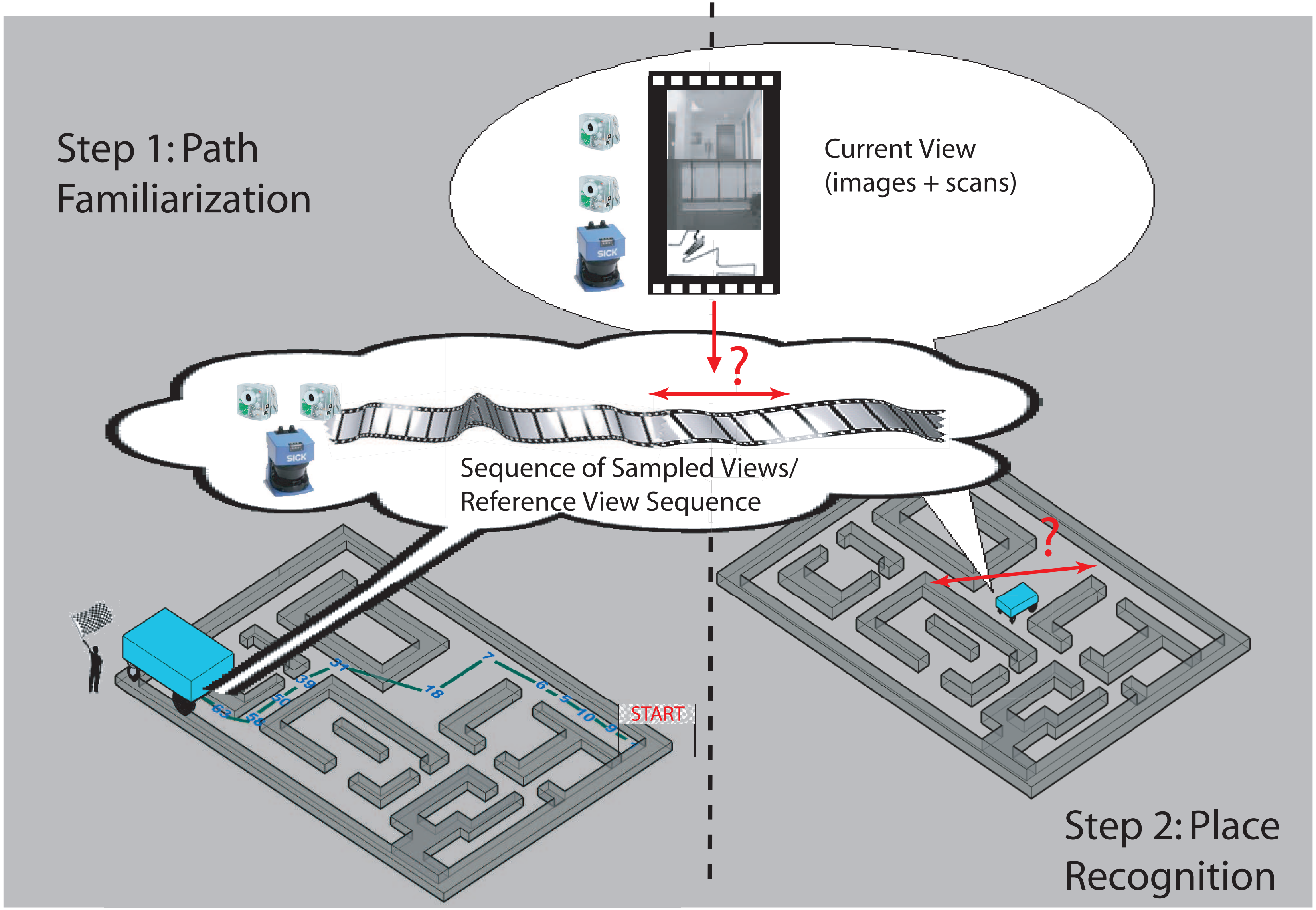


Figure 1.10: The image illustrates the recovery of the Reference Sequence View index using a Single View. The solution involves registering the current sensor data with a previously gathered sequence of Views to find the best ’fit’ or ’match’.

that the algorithm provide an unbiased estimate of the distance between the current view and each of the views in the Reference Sequence. When large numbers of features are employed to represent a view, it becomes increasingly difficult to design and use a metric that provides unbiased distances between views. This problem, often referred to as the curse of dimensionality [Bellman 61], results in the failure of matching. Given a noisy observation, it is more probable that any random view will be closer to the current view than the actual corresponding view in the Reference Sequence. For a distance metric to provide un-biased results on the high-dimensional feature data, some sort of dimensionality reduction is imperative.

The algorithm described by us, in chapter 2, as a solution to this problem is the reduction of the dimensionality of the 'space of features' using a Bernoulli Mixture Model.

An important assumption that is required throughout the work described in this thesis is that the initial Environment Familiarisation stage must be performed in an environment that is as reliable as possible. In subsequent travel through the environment, statistical treatment of differences in perception allow the robot to handle changes in the environment.

1.4.2 Place Recognition using the context of the View Sequence

As was mentioned in section 1.3.2, despite the use of a large number of features, the twin problems of scene variability and place aliasing implies that it might not always be possible to correctly and uniquely identify a place using a single observation.

A better way would be to make an inference inference of the position of the robot in the Reference Sequence after collecting observations over a finite number of positions, as shown in Fig. 1.11. This approach would still utilize the distance metric used in the case of a single view and would still require dimensionality reduction methods to be able to make unbiased estimates using a large number of features. Such a scheme could improve the performance of the method by using a Prior Probability to favour the chances of matching latter observations based on the estimation results of previous observations.

Utilizing multiple views implies that the motion of the robot must be taken into account in some way. In this work, a simple model for robot motion has been employed to consider the

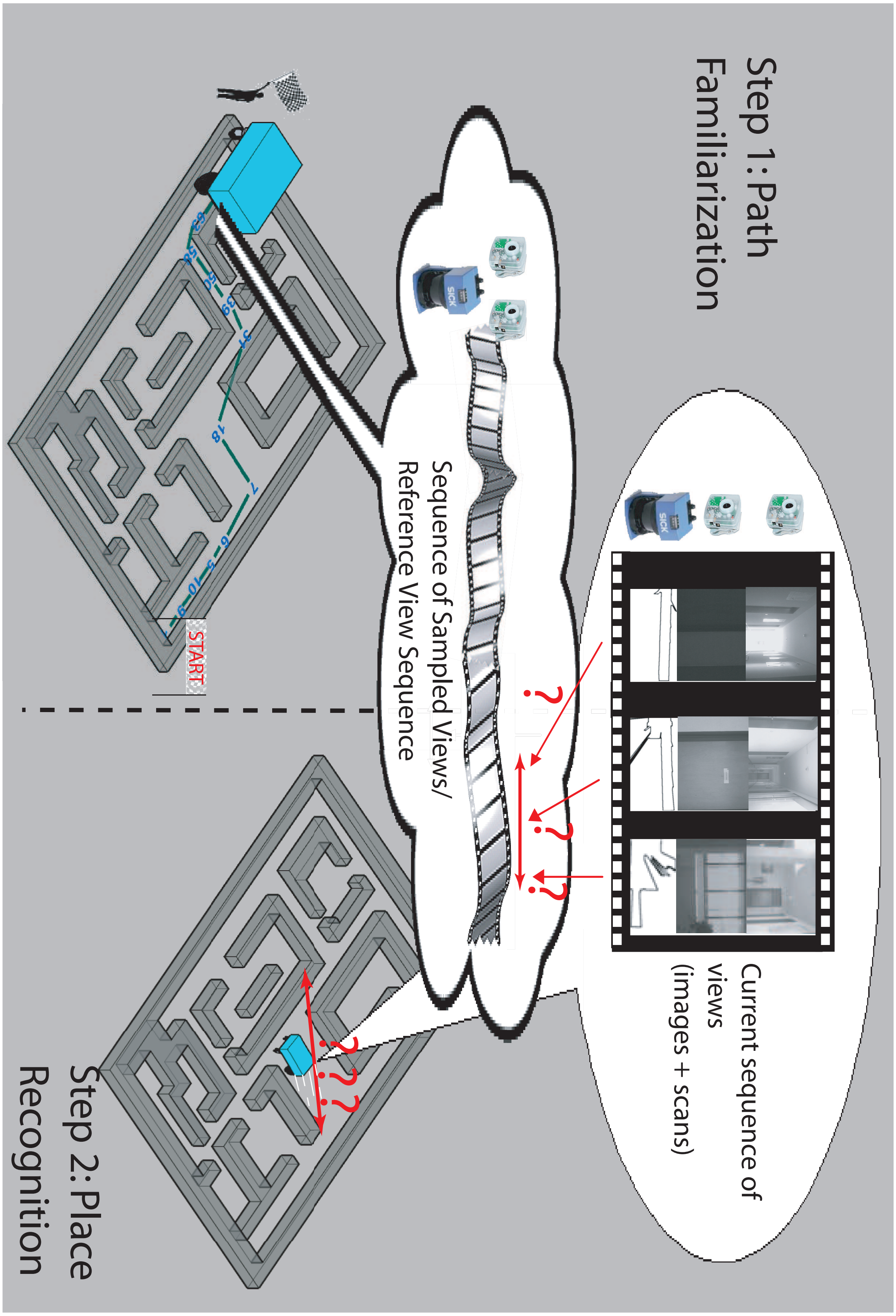


Figure 1.11: A schematic of the Place recognition procedure using the context of the Reference Sequence. The Single View matching described earlier is now extended to the case in which a chronological sequence of query views is available.

possible views as the robot moves along the Reference Sequence. This model leads the robot down the Reference Sequence and provides a very imprecise estimate of the next views that might be seen.

We propose to use a particular type of Dynamic Bayesian Network, the Hidden Markov Model, HMM to infer the position of the robot in the Reference Sequence from multiple observations. The application of HMM to this problem is described in chapter 3.

1.4.3 The Lost View

In an earlier section the concept of the Place of the Lost robot was introduced. When referring to views that are visible from the different Places, it becomes essential to ask the question: what will be the observed view at the $Lost\_Place$, or what will the *Lost View* be like?

The Lost View is a paradox, since, not knowing where the $Lost\_Place$ is, the robot cannot know what will appear at such a place were it to be lost. In this work it has been assumed that the view that is obtained at such a $Lost\_Place$ is the *average combination of features for the entire Reference Sequence*.

This is based on the assumption that, in the absence of any additional information, the distribution of features at a view taken at any place in the environment that is not included in the Reference Sequence should be the same as the distribution of features in the views in the Reference Sequence.

It is possible that one or more views in the Reference Sequence might also have this same, 'average' distribution of features, though, in the high-dimensional feature-space this probability is very low.

1.5 Summary

This chapter presented the proposal of this thesis which started out with the aim to extend previously developed work in which the robot executes a sequence of motion behaviours, conditional on the sequential description of the expected perception.

Before describing the approach in detail a review of the types of Maps used in robot Localisation, with special emphasis on separation of the distribution of the motion of the robot from the distribution of the sensory perception was presented. An important conclusion that can be made from the comparison of Metric and Topological approaches to the representation of an environment is the fact that the Metric map allows a joint probability of the events that can be sensed by the robot to be expressed within a single coordinate system such as on a piece of paper, whereas a topological map will use two or more conditional distributions to represent the same events.

The idea for place recognition by matching the currently observed sensory data with a sequence of views previously gathered during a Environment-Familiarization phase is presented in two parts, the first of which involves the use of a single view followed by the application of a View sequence to perform Place Recognition.

Chapter 2

A Model to Represent Individual Views

2.1 Introduction

The problem of assigning indices to combinations of features and the retrieval of this index upon presentation of a new observation, is found in many domains. It lies at the heart of applications involving detection of objects within a scene, image retrieval from a database, face recognition and gesture recognition, among others. These applications work by extracting features or properties that describe a scene or a face or some gesture primitive and then identify sufficiently unique combinations of features that allow the recognition of each object view, scene, face, gesture, etc. The question that we seek to answer is: 'How are observations represented in the Reference Sequence?'.

In most applications, a trivial comparison of features from a pair of views, is not enough. Sensor data is usually noisy and because there might be some changes in the views. Some of the features will be correlated with others and this correlation will vary among the features. Mathematical techniques must be employed to perform un-biased estimations using the reduced or noisy data.

Robot localisation must, almost obligatorily, be probabilistic in nature in order to account for the impossibility of modelling an activity as complex as perception. Among the methods that have been applied, 'Bayesian Inference' is, for many researchers, the preferred way of handling uncertainty in the perception process. As presented elegantly in [Reporter 00], Bayesian Inference is a tool that allows the integration of new evidence such that it may be used immediately (even as it slowly drips in). In mathematical terms, the probabilistic Localisation of a robot in an environment plugs in new evidence into the well-known Bayes equation (2.1) where $P(\text{Current View}|\text{Current Position})$ stands for the model of the system that defines the expected observations given a particular robot position. Localisation methods differ in the way they calculate the numbers to plug into this equation and the result that they draw from it.

$$P(\text{Current Position}|\text{Current View}) = \frac{P(\text{Current View}|\text{Current Position}) \times P(\text{Current Position})}{P(\text{Current View})} \tag{2.1}$$

2.2 Using Features to represent Places

Robot sense the environment using a variety of sensors [Borenstein 94]. In our work, we have been primarily interested in using Laser Range Finders and cameras. These sensors are fairly common on today's robotic platforms and the science of extracting scene information using these sensors has progressed considerably.

In an attempt to obtain more reliable features in the environment, many range-sensor based methods extract lines and other primitive features from the laser scan. Cox [Cox 91], attempts to match points extracted in the laser range scan with the lines in the map. He subjects readings from a LRF to an iterative procedure of rotation and translation until convergence results. Here, the objective function to be minimised is the distance of each scanned point from the nearest line, the environment itself being represented as a line model. The iterative procedure translates and rotates the scan till the sum of squares of these distances is minimised. The inverse transformation that produces the robot position estimate. Odometry data assists in reducing the iterations and provides localisation in the absence of new sensor data.

In Sequeira [Sequeira 93], a range scan is matched with a line-representation of the hall in which the robot finds itself. Utilising a modified version of Cox's algorithm, an attempt is made to match the scan obtained with the line representation. The paper reported very good precision in localisation and a good performance of the navigation system, considering the bulky hardware and small processing power. However, from the scan representations presented, the hall seems to be quite bare and contains few objects that significantly altered the rectangular shape of the hall boundaries.

Other methods make strong assumptions about the environment that generated the scans. For example in [Jensfelt 99] the assumption of orthogonal walls is made and the values of the distances to the walls are used to update the position estimate. The X and Y Cartesian coordinates are updated alternatively using different walls. The error in the angular orientation, a more serious problem in odometry-based systems, is corrected at every update. A reduced representation of the world model, limited to four walls representing the outer extents of the laboratory, reflects on reduced information content that is typical in 2D range scans.

Ribeiro and Gonçalves in [Ribeiro 96] utilise pairs of vertical edges in the environment (corners in the laser scan) to obtain a localisation estimate. The environment is scanned selectively in the direction in which the edges are expected. An environment model is utilised in order to obtain the absolute position of the edges and to choose from among various possible edges. An initial position estimate, obtained in case of a moving robot from odometry, is utilised to aid the search procedure and to strengthen robustness of the estimate.

Dynamic environments present a problem for Scan Matching algorithms. In [Bengtsson 98], Bengtsson and Jonasson present the ’iterative Dual correspondence’ algorithm, a method for matching consecutive scans so that the pose change can be recovered. This estimated pose change is then integrated using a Kalman Filter.

In [Arsénio 98], a LRF is mounted atop a Pan and Tilt Unit (PTU) and was utilised to obtain a depth picture of the robots surroundings. That comprehensive work includes an algorithm that has access to a 3-D representation of the hall. In that representation, objects with vertical edges, together with the sides that make up the edge, are represented, chosen on the basis of contrast and probability of being observed. Vertical edges are extracted from the laser scans and an attempt is made to match them with the vertical edges of the objects in the map. To simplify computation and improve results, some pre-processing of the data is carried out. In addition, the scanning for features based upon an initial estimate of the robots current position is performed in order to identify what objects might appear in the laser scan and what objects might be partially or fully occluded. Occlusion effects and the range of angles through which each edge is visible from each cell are taken into account.

Drumheller, in [Drumheller 87], describes a multistage line-extraction algorithm in which data from US sensors from all around the robot are used. Using interpretation trees, the lines scanned are then matched with corresponding features in an available map. In a similar vein, Dudek and MacKenzie in [Dudek 93], provide another method by which the scan data is viewed as being composed essentially of lines. An iterative matching scheme is devised that attempts to fit the scanned lines to actual lines existing in the model. The extraction of lines from the laser scan continues to be a popular approach in the extraction of information through a segmentation of objects using the laser scan data, [Nguyen 05], [Nock 03] and [Sack 04].

Some methods have tried to improve on the original line-feature extraction algorithms by performing some pre-processing. For example, Tang et al. [Tang 04], aim to reduce the roughness of the range finder data before isolating points by employing a method that is supposedly similar to the scale-space approach used in images. The line fitting algorithms is said to work better on this lower-frequency data.

Still other methods seek to parameterize laser range data, for example, by converting the scan into an ordered set of polylines, [Lakaemper 05]. The proponents of the system claim better matching properties between a pair of scans and easier integration of a new scan within an existing map.

Representing places only in terms of lines (and corners) provides a limited amount of information. Many places in the environment are found to have similar representations and these methods do not scale up easily to larger environments.

In [Tovar 03], Tovar et al. use a combination of places and motion paths to represent a portion of the environment that has been previously explored. Using only range measurements, the gaps between obstacles make up the features of interest and the behaviours that merge/close gaps or that create new ones are noted. The path to the goal is the combination of these behaviours. The environment for which the method has been tested seems to be small and the placement of obstacles appears to excessively facilitate the detection and characterisation of the above-mentioned gaps.

Other publications have expanded of the set of features that are segmented from laser scan data to include trees, kerbs and other features in select outdoor environments. Manandhar and Shibasaki in [Manandhar 01] extract roads, buildings, tunnels and other outdoor features by modelling 3D range data. In indoor environments too, composite landmarks including lines and other simpler features have been used, [Xiang 04].

There have been attempts to represent places in the environment with unique sets of the features, sometimes termed as ’fingerprints’. The fingerprints consist of a list of features (a string of symbols to be more precise) that lie around the robot. Each feature is represented in terms of a symbol and each place is denoted in terms of a string of these symbols. This approach can hope to draw from the wealth of string-based matching algorithms. A serious point of concern is that most of the simpler string-matching algorithms work on the principal of independence of bit errors and this might not be most appropriate approach when there are correlations between the presence or absence of certain features. Also, although the method aims to be multi-sensor, the features still need to be integrated into the same string implying that the features must still possess certain basic similarities.

In [Lee 00], for example, each feature that is extracted from the laser range scan is given a symbol and each scan is described in the form of a string, e.g. mMmMmMmMmDCm. The string alphabet, in this case (M)axima, (D)iscontinuity, (m)inima, (c)onnection), depends on the features extracted from the laser scan. Other methods use ’sections’ of the laser range scan so as to minimise the effect that changes in one part of the scan will have representation of the place. The idea is also the gist of the so-called ’fingerprint’ approach described in the section discussing Map Topology, [Lamon 03, Tapus 04a, Tapus 05, Tapus 04b].

Vision sensors or cameras have been used independently, see Appendix A for vision features, or coupled with Laser Range Finders to aid the segmentation of scan data. Arras and Tomatis in [Arras 99] attempt to introduce a vision sensor, a CCD camera, to a robot already having a localisation system based on a LRF. The stand-alone LRF-equipped system achieves good performance in rooms in which the environment is made up of distinct features. The performance of the system undergoes a drastic reduction in efficacy when presented with long corridors and situations in which the laser beam is subject to specular reflection. With the aid of a vertical-edge extraction procedure the method seeks to include data that is more reliable in situations where the LRF is more prone to provide unpredictable or highly ambiguous data.

A number of modern mobile robot platforms have employed multiple sensors with a view to minimising the situations in which a particular type of sensor can fail or provide less accurate data. In [Lamon 06] the configuration of the ’Smarter’ platform is described as including multiple laser range finders, omni directional and monocular cameras, inertial measurement units, differential GPS and other sensors. Information from these sensors is filtered into an estimate for the vehicle position, but only those features can be used to which a distance (or bearing) is associated.

2.2.1 Multiple Feature Integration

Perception is usually performed in environments that are atleast partially ’dynamic’ . Additionally, sensor data is often noisy or incomplete. Gathering ’good’ or unchanging features is one way of ensuring that there is some chance of localisation [Marsland 01]. It is even more important to build redundancy into the perception process. Robot localisation methods, should make use of a large number of features so as to be robust against the above problems.

Tebo [Tebo 97] says that ’*Sensor Integration is concerned with the synergistic use of multiple sources of information. Sensor Fusion is divided into three classes: complementary sensors, competitive sensors and cooperative sensors*’. While such a definition is useful in understanding the need for sensor integration, in practice, integrating measurements taken by sensors with different measurement and error models is not a trivial task since all the above three usually occur simultaneously, in varying degrees. Multiple sensors are usually employed to address the weaknesses of individual sensors and increase the sensory capabilities of the robot.

Two principal approaches to feature integration are possible; filter-based and wrapper-based. In filter-based methods, preception models for sensors that integrate the different sensors are assumed and these are imposed on the data. Such methods usually require a ’registration’ step to be performed in order to express the features from one sensor in terms of the feature space of the other sensor (usually distance and bearing). Wrapper-based approaches, on the other hand, attempt to facilitate the (usually NP-hard) mathematical procedures employed to approximate the integration of all features simultaneously [Kohavi 97].

Given that several thousands of features must be integrated, approximate techniques to reduce the dimensionality of the features seem to be most appropriate. A reduction in the number of correlated features also reduces the amount of redundant data and can reduce the time required for the procedure to run. Wrapper based methods are commonly employed in the literature, for example Ohba [Ohba 97] suggests the use of a 'Global Goodness' test to select windows of interest in an image. This criteria has the characteristic of testing the uniqueness of a single feature rather than of a combination of features (see also [Winters 02], [Gerstmayr 04]). Other approaches, including [Fleuret 04], [Peng 05], [Vlassis 00], have attempted to use methods based on Information theory to select a smaller subset of independent features.

The integration of features from multiple sensors is a topic of keen research in the area of mobile robot localisation. A typical combination of sensors is a camera and a LRF where discontinuities in the Laser range scan is associated with features in the image. For example, Castellanos et al. [Castellanos 01] employ vertical edges extracted from an intensity image and corresponding corners in the Laser Range Scan. The robot is obliged to know the rigid transformation that links the features in one sensor with those in the other. This form of [sensor] coupling requires a calibration process to obtain the transformation and the transformation itself may be of little use if the sensors have different ranges and accuracy. Vale [Vale 05] performs an evaluation of different sensor data for greater suitability to Localisation after travelling through the environment and evaluating how many times the robot localized itself. Interaction between the features extracted from different sensors is incidental and the set of features to be used must be decided before-hand.

Other approaches have been proposed to combine data from the sensors used in this work, and given the variety of features-based methods using vision and range scans, the combinations are many (see the bibliography maintained by Keith Price at http://iris.usc.edu/Vision-Notes/bibliography/match-pl502.html).

Methods that explicitly reduce the dimension of features with continuous values are common in many perception fields including face recognition, speech recognition, etc. Principal Component Analysis (PCA) and more application-specific methods derived from PCA such as the Eigen-Images constitute an important class of data-reduction methods. Mixture models are a common solution to modelling data that is believed to follow non-parametric distributions. In [Sajama 05], Sajama and Orlitsky demonstrate the use of Mixture models composed of Gaussian, Bernoulli and Exponential distributions as a solution to the classification problem. Clustering or classification methods based on Mixture models seek to identify features that are more correlated with members of their own group than with members from other groups. McLachlan and Peel [McLachlan 00] provide a good reference to the general topic of Finite Mixture models. While most of the work in the field is in the area of Gaussian and Exponential distributions, other distributions have also been discussed.

A central part of this thesis involves a different approach to the task of feature integration. After extracting different types features from each view, using diverse techniques, these features are represented using binary symbols, $[0, 1]$ through one of the following:

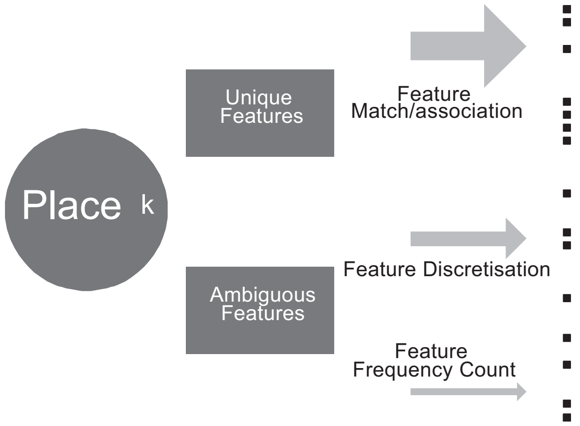
1. matching extracted features against a feature database to detect their presence (or absence),

2. categorising features, and

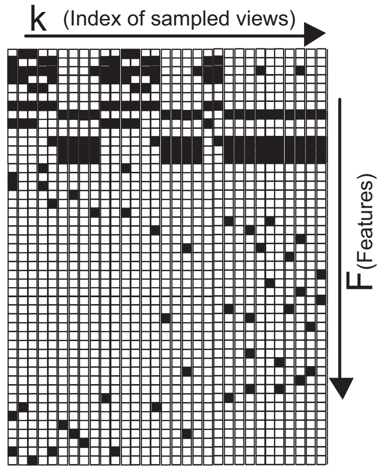
3. discretizing continuous-value features, as shown in Fig. 2.1a.

The ultimate goal of this step is to integrate large numbers of features into a common representation to perform place recognition and, ultimately, to achieve a a robust robot Localisation estimate. The robot is first led through the environment during an Environment Familiarisation phase. The sensors sample the environment, generating a sequence of views, which we call the Reference Sequence. This procedure was depicted graphically in Fig. 1.10 and the views of the Reference Sequence are indices for places in the environment that lie along the path covered by this Reference Sequence.

A matrix of binary features that represent the features extracted from the views of the Reference Sequence is denoted as the Feature Incidence Matrix, FIM , as depicted in Fig. 2.1b, where



(a) Creation of a view of binary Features.



(b) The Feature Incidence Matrix.

Figure 2.1: The process of categorisation of the features to create binary features is performed in diverse ways, depending on the sensor and on the type of feature.

each row denotes a particular feature that was extracted from one or more images or laser range scans. Each column represents a place at which sensor data was obtained. The presence of a ’1’, depicted as a black square in Fig. 2.1b, in any column signifies that the feature was observed in an image or laser scan taken at that place.

2.2.2 Using Features in Binary Form

As seen in the previous sub-section, the features that are obtained from the sensors reflect logical information (the presence or absence of a feature in the current observation), or numerical information (the number of doors or the free space as measured by the range scan). This idea of converting numerical (continuous or discrete) valued features into binary form is not new and can be seen in [Wang 05], [Fleuret 04].

Both of the types of features described above are represented using $[0, 1]$ symbols and treated like binary data. We have used different techniques, described Appendices A and B, to extract a large number of features from each view comprising of vision and laser scans. When SIFT or similar local features are captured from images, the number of features grows quickly into the thousands whereas the number of views might measure only a few hundreds. This large number of features makes the estimation of the current view from a comparison of feature values an illposed problem, as referred to earlier as the curse of dimensionality. A discussion on the nature of the features used within the views was initiated in section 1.4.1.

We can describe a '*feature space*' as a coordinate system where the value along any coordinate axis is discrete and can take values of 0 or 1. Each of the features from the very large number of binary features that are used to describe each view in the view sequence can be looked as an additional dimension. Each view is represented as a point in the feature space, and because of the presence of noise that can affect the sensors, there are regions and dimensions of the feature space that are very sparsely populated with views. Therefore, before comparing a view with all the views in the sequence, a data dimensionality reduction procedure should be performed on the feature space.

Data dimensionality-reducing methods such as PCA are not meant to handle binary data. Mixture-Models have been utilised essentially with Gaussian distributions but, more recently have been applied in the context of binary data in the form of mixtures of Bernoulli distributions. Articles such as [Wang 05] and [Kaban 00] go some way to demonstrate the usefulness of binary features. In [Gonzalez 01] the contexts in which the words are used in a sentence are converted into binary features. Mixtures of Bernoulli distributions have been used to model data containing binary features in [Gonzalez 01], [Juan 04] and [García-Hernández 04].

We have adopted a finite Mixture Model in which the individual mixture components are Bernoulli distributions to reduce the dimensionality of the feature space. Over the next sections, the technique to match a current view with each view in the FIM shall be described in greater detail and look at a concrete application where features that can be extracted from a Laser range scan and from multiple cameras can be integrated for better place-recognition results. Section 2.3 first introduces the Bernoulli Mixture model beginning with a description of the method following which an application to a toy example whose simplicity is meant to demonstrate the results achievable by the Bernoulli Mixture model. In section 2.4 examples are presented of some of the approaches that have been used to perform to extract features in range scan and in images.

2.3 Integration of Binary Features

Each view that is gathered by the sensors of the robot is converted into a column vector of binary symbols. This section describes a solution to the problem of matching a binary column vector with the FIM, $\mathcal{V}$. Each row of the FIM corresponds to a feature j and each column represents a separate vector V_k, identified by an index, k. Each entry in the FIM (2.2) might be represented as ${}_jV_k$ where the first subscript indicates the feature and the second subscript, the vector index. ${}_jV_k$ takes value 1 if feature j appears (is visible) in vector V_k and takes the value 0 otherwise.

$$\mathcal{V} = \begin{bmatrix} {}_1V_1 & {}_1V_2 & \dots & {}_1V_K \\ {}_2V_1 & {}_2V_2 & \dots & {}_2V_K \\ \vdots & \vdots & \ddots & \vdots \\ {}_NV_1 & {}_NV_2 & \dots & {}_NV_K \end{bmatrix} \tag{2.2}$$

Suppose we wish to retrieve the index of the vector that is most similar to an observed vector V_{obs}. A distance metric could be designed to evaluate the similarity between V_{obs} and each vector V_k in the FIM. One such metric could be the number of corresponding binary features in each vector that are unchanged, the 'Hamming distance'.

A direct comparison using a metric such as the Hamming distance makes the assumption that the individual features in each vector are independent. If the features were not independent, inferences that are made, based on this comparison, might be biased toward certain vectors in the FIM. In such circumstances a Mixture of Bernoulli Distributions is used to model the binary FIM and reduce the dimensionality of the vectors in the FIM before retrieving the index of the vector most similar to V_{obs}.

2.3.1 Formulation of the Bernoulli Mixture Model, BMM

Mixture models assume that there exist a finite number of parametric distributions which, when mixed together in a particular proportion, result in a distribution that best describes the data to be characterized. In this case, the observation V_{obs} can be assumed to be vector of binary features $\{0, 1\}^N$, obtained from a particular mixture of Bernoulli distributions, as in (2.3).

$$P(V_{obs}|\Theta) = \sum_{c=1}^{C} \alpha_c P(V_{obs}|\Theta_c) \tag{2.3}$$

In (2.3), Θ denotes the parameters of the distribution of the vectors that compose the Mixture. These parameters include the C component vectors, the Θ_cs, and the proportions in which these are mixed, the α_cs. Each α_c can also be looked at as a sort of prior probability of the component c within the complete mixture model, subject to the constraint $\sum_c \alpha_c = 1$. The term $P(V_{obs}|\Theta_c)$ can be determined using (2.4) where each Θ_c is a multivariate vector of Bernoulli probabilities each of whose N rows indicate the probability of success for a particular feature. Each $P(V_{obs}|\Theta_c)$ is a measure of the similarity of V_{obs} and the component Θ_c and since the features that make up the components can be assumed to be independent, it can be calculated as in (2.4).

$$P(V_{obs}|\Theta_c) = \quad \prod_{j=1}^{N} {}_j\Theta_c^{\,{}_jV_{obs}} (1 - {}_j\Theta_c)^{(1-{}_jV_{obs})} \tag{2.4}$$

To obtain the parameters of the component vectors and mixture priors, it is assumed that the component vectors are independent and the likelihood of the mixture satisfying the FIM is expressed as in (2.5).

$$P(\mathcal{V}|\Theta) = \prod_{k=1}^{K} P(V_k|\Theta) = \mathcal{L}(\Theta|\mathcal{V}) \tag{2.5}$$

The optimisation task to find the mixture that best explains this $\mathcal{V}$ can be expressed as in (2.6), i.e. to find the value of Θ that best satisfies the distribution of features in $\mathcal{V}$.

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}}\, \mathcal{L}(\Theta|\mathcal{V}) \tag{2.6}$$

The preferred method of solving the Mixture Model problem is the Expectation Maximisation algorithm. McLachlan ([McLachlan 00], page 19) states '...it will be seen that conceptualization of the mixture model ...(hidden data + component distributions)... is most useful in that it allows the Maximum likelihood estimation of the mixture distribution to be computed via a straightforward application of the EM algorithm.'. The EM method applied to the Mixture problem assumes that the data is only partially available. It becomes fully known through the use of a matrix of coefficients denoted henceforth as Z, called the 'missing data' or the 'hidden data' or still the 'unobserved data'. Two notations for Z are used: z_k is used to refer to the vector in Z that corresponds to the view k and z_{kc} to the element in Z that corresponds to the view k and the component c. If this Z is introduced to expression (2.3), the likelihood of the observations given the entire data can be expressed as in (2.7) and further simplified to (2.8).

$$\mathcal{L}(\Theta|\mathcal{V}, Z) = \sum_{k=1}^{K} z_k \log(\sum_{c=1}^{C} \alpha_c P(V_{obs}|\Theta_c)) \tag{2.7}$$

$$\mathcal{L}(\Theta|\mathcal{V}, Z) = \sum_{k=1}^{K} \sum_{c=1}^{C} z_{kc}(\log(\alpha_c) + \log(P(V_{obs}|\Theta_c))) \tag{2.8}$$

The EM algorithm proceeds in two stages: the *Expectation* stage attempts to reach the best value for the missing data Z, by keeping the parameters of the Mixture model constant (2.9), while the subsequent *Maximization* stage attempts to optimise the components and mixing parameters themselves by using the values of the 'missing data' obtained in the expectation step just performed (2.10), (2.11). The method then alternates between the two steps until some termination criteria is satisfied.

$$z_{ki} = \frac{\alpha_i P(V_k|\Theta_i)}{\sum_{c=1}^{C} \alpha_c P(V_k|\Theta_c)} \tag{2.9}$$

$$\alpha_c = \frac{\sum_{k=1}^{K} z_{kc}}{K} \tag{2.10}$$

$$\Theta_c = \frac{\sum_{k=1}^{K} z_{kc} V_k}{\sum_{k=1}^{K} z_{kc}} \tag{2.11}$$

As a termination criteria we have adopted a lack of change in the mean error when the Mixture parameters are applied to the original data. In the case of such applications, where the parameters of the Mixture models are required for the purpose of classification, the process is usually stopped quite early, when the reduction in the Mean Error in 2 successive iterations is not significant.

Mixture models used for classification make use of both, the Mixture parameters and the posterior probabilities over the components, the Z are used to evaluate the likelihood in the space of the vectors in the Reference Sequence as in (2.12) where $P(V_k)$ represent the prior probabilities on each index k.

$$P(k|V_{obs}) = \frac{\sum_{c=1}^{C} P(V_k) z_{kc} \alpha_c P(V_{obs}|\Theta_c)}{\sum_{k=1}^{K} \sum_{c=1}^{C} P(V_k) z_{kc} \alpha_c P(V_{obs}|\Theta_c)} \tag{2.12}$$

The *Maximum Likelihood Estimation* approach is used to obtain the matching vector, the index k^*, in $\mathcal{V}$ that best describes the vector to be matched, V_{obs}.

$$P(k = k^*|V_{obs}) = \max_k P(k|V_{obs}) \tag{2.13}$$

2.3.2 An Application of Bernoulli Mixtures to a Toy Problem.

To obtain a feel for the Bernoulli Mixture model, two toy FIMs are presented in Fig. 2.2. As was explained earlier, a row in the Feature Incidence Matrix denotes a single feature. Each column denotes a particular vector. The presence of a feature is indicated by a black square and the absence by a white square.

For simplicity and ease of visualisation, both the toy FIMs are created from two distinct combination of features. Each of the two combinations is a complement of each other meaning that the features that are present in one combination are not present in the other.

The FIM at left is easily recognised as being composed from a combination of two distinct vectors, each of which is a complement of the other. The features are perfectly (and consistently) correlated. These complementary vectors would be the expected result when the FIM is modelled as a Bernoulli Mixture model.

The FIM at right is identical in layout to the left FIM, but, in this case, some features have flipped in order to simulate the presence of noise. If this FIM is also modelled as a mixture model, one would still expect to see the original vectors as the main components given that the ’injected noise’ is quite small.

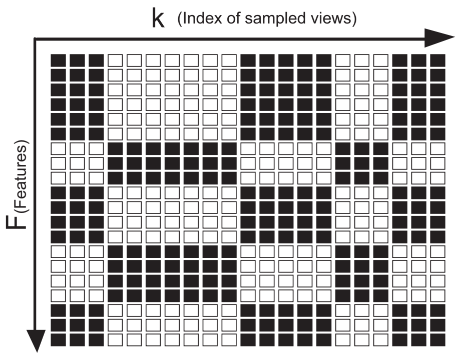
The two FIMs can be modelled as a Bernoulli Mixture model whose parameters are given by the EM algorithm. The left FIM in Fig. 2.2 was modelled using two components since it has been obviously created from two different column vectors. In an attempt to gauge the effect of the added ’noise’, the right FIM was modelled using a Mixture model with four components. Note that a greater number of components could be used, but four components should be able to demonstrate the effect of the small amount of noise.

After running the EM algorithm for Mixture Models, with two and four components respectively, component with values as shown in Table 2.1 and Table 2.2 and the mixture coefficients as shown in Table 2.3 and Table 2.4, are obtained respectively. As can be seen, most of the layout of the noisy FIM is explained by components Θ_1 and Θ_3 (the original components in the noiseless data) and the distribution of these components is quite similar to the corresponding components for the noiseless FIM.

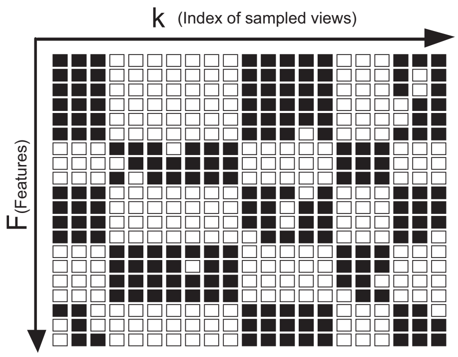
2.4 Real Features for Place Recognition

If we return to our localisation problem as described in section 2.2, and take each (column) vector in the FIM to be a vector obtained during the Environment Familiarisation phase, then the parameters of the BMM allow us to retrieve the index that is most similar to the currently observed view, V_{obs}.

This section and the next presents results from experiments performed using a ’Robuter’ robot platform by Robosoft and a Segway RMP 200. A LRF and two Unibrain Firewire cameras have been added as in Fig. 2.3. The forward-looking camera, Camera #1, looks in the direction of



(a) FIM for Noiseless Samples.



(b) FIM for Noisy Samples.

Figure 2.2: Sample binary FIMs, where the FIM at left shows vectors sampled from two populations. The FIM at right shows the same vectors with some added noise (the binary features in some vectors have been flipped). The black squares (value of 1.0) indicate the presence of features.

Figure 2.3: Arrangement of sensors on the Robuter mobile robot platform. The sensors used in this work include the forward-facing Camera #1, the lateral-facing Camera #2 and the LRF mounted on the front of the robot.

Table 2.1: Components for noiseless FIM shown in Fig. 2.2a

Θ_1	Θ_2
1	0
1	0
1	0
1	0
1	0
1	0
0	1
0	1
0	1
1	0
1	0
1	0
1	0
0	1
0	1
0	1
0	1
1	0
1	0
1	0

Table 2.2: Components for noisy FIM shown in Fig. 2.2b.

Θ_1	Θ_2	Θ_3	Θ_4
1	1	0	1
0.90	1	0	1
0.90	1	0	1
0.90	1	0	1
0.90	1	0	1
0.90	1	0	1
0	0	0.91	0
0	0	0.91	0
0	0	0.91	0
0.90	1	0	1
0.90	1	0	1
0.90	1	0	1
1.00	0	0	0
0	0	1	0
0	0	0.91	0
0	0	0.91	0
0	0	0.91	0
0.90	0.50	0	0.44
0.80	1	0	1
0.90	1	0	1

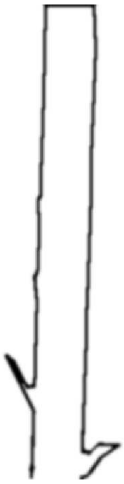
Table 2.3: Coefficients for noiseless FIM shown in Fig. 2.2a.

α_1	α_2
0.52	0.48

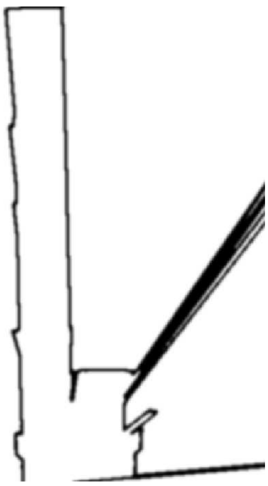
Table 2.4: Coefficients for noisy FIM shown in Fig. 2.2b.

α_1	α_2	α_3	α_4
0.44	0.06	0.48	0.02

(a) Scan 3

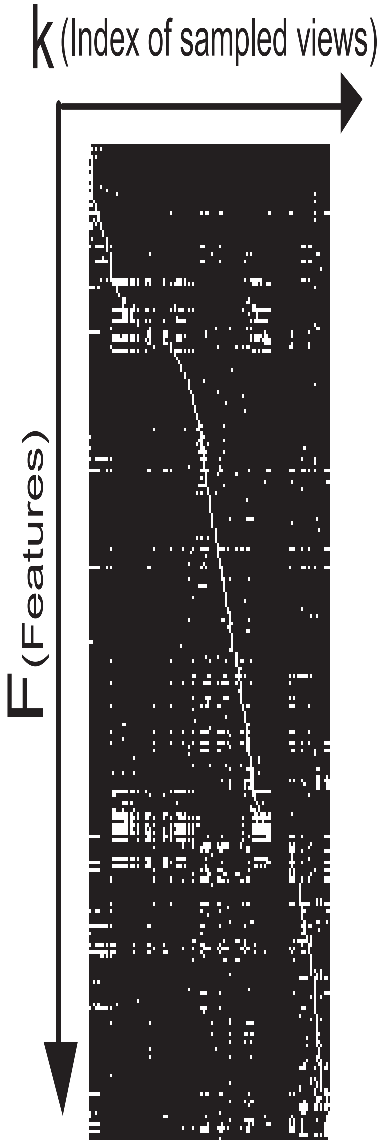


(b) Scan 29



(c) Scan 77

(d) Scan 107



(e) The binary Feature Incidence Matrix.

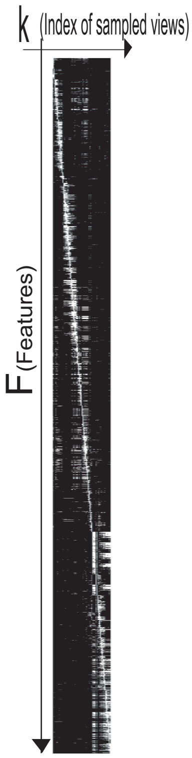
Figure 2.4: Representative laser range scans from a Reference Sequence of 118 scans taken along a Hallway. The Feature Incidence Matrix created from the features extracted from the 118 scans taken is also shown.

(a) Image 12.

(b) Image 32.

(c) Image 40.

(d) Image 63.



(e) The binary Feature Incidence Matrix.

Figure 2.5: Seen here are four representative images from a sequence of sixty seven images taken by Camera #2 as the robot moves inside a large room. The figure at right is the resultant Feature Incidence Matrix, FIM.

robot motion while the lateral camera, Camera #2 is mounted at a sufficient height to view posters and other texture appearing on the walls of the building. The cameras are capable of taking VGA-sized images and the SICK laser range finder provides a set of 361 range measurements taken through a 180 degree interval.

SIFT features were extracted from images captured from Camera #1 and Camera #2 and various types of features were extracted from the LRF. The presence of a particular SIFT feature in a particular view is indicated with a binary value of ’1’. The background on SIFT features and the details of the implementation in a library developed in the course of work performed on this thesis is detailed in Appendix A. In Fig. 2.5 a few images recorded by Camera #2, while travelling within a large room are displayed. The corresponding FIM for the SIFT features extracted from the whole set of sixty seven images is shown in Fig. 2.5e.

In the case of the Laser Range Finder, a number of different features were developed and utilised in the same fashion as the SIFT features from cameras. Details about the various features and how they are represented as binary features are included in Appendix B. Previously developed algorithms that allowed segmentation of door-like openings were extended and the extracted doors were classified according to their distance from the robot, resulting in the creation of new features. New feature extraction algorithms were developed to extract and classify walls. In a novel application of image description techniques, HU moment vectors were created from the profile of the Laser Range Scans and used as features.

The features that are continuous valued, are discretised and the discrete values converted into binary representation as discussed in section 2.2. Discrete-valued features such as counting the number of walls and doors, are directly converted into a binary representation.

2.4.1 Bernoulli Mixture Model applied to View sequences from a Single Sensor

The SIFT features from a single sensor (Camera #2 in Fig. 2.3) are converted into binary form. These SIFT features, extracted from a sequence of images lead to the creation of the FIM. The technique described in section 2.3 is employed to model this FIM as a Bernoulli Mixture model.

Two modifications that make the calculation of the mixture model feasible for the large number of features are are available. These are

1. The distance or metric between a component and a view is calculated using a modified expression described below, and

2. Only visible features from the FIM are used at any moment to calculate the BMM parameters.

When features measure in their thousands, it becomes necessary to impose limits on the probabilities obtained from the Bernoulli distributions. In this case the value that the elements of the components can take is restricted to 0 and 1. The purpose behind this idea is that, among thousands of features, it does not matter whether the probability of a feature appearing is 90% or 95%. The expression for calculating the likelihood becomes (2.14) instead of the earlier (2.4). The cost_j term is a penalty term for feature j with values between 0 and 1. It should take values close to 0 for invisible features and values closer to 1 for visible features. The expression that was developed independently during this work was found to be quite similar to one of the models proposed by Nadif and Govaert in [Nadif 98].

$$P(V_{obs}|_j\Theta_c) = \prod_{j=1}^{N}(\text{cost}_j)^{|_jV_{obs}-{}_j\Theta_c|}(1-\text{cost}_j)^{(1-|_jV_{obs}-{}_j\Theta_c|)} \tag{2.14}$$

For as short as a 100-meter-long stretch of indoor environment the system yielded a few thousand of features from a sequence of images obtained from a single camera. Instead of using all the features, visible and invisible, in the FIM it was found to be more advantageous to use only the visible features whilst calculating the parameters of the BMM. The calculation of the parameters is much quicker (due to the smaller number of operations required in the EM) and the parameters are more stable. In a personal communication with Alfons Juan, one of the authors of [Juan 04], this was put down to the possible difficulty of the EM algorithm to overcome local minima. The problem of local minima is more serious in higher dimensional (more features) problems. Better results in terms of faster convergence of the BMM parameters and a lower error of the parameters were obtained by using the reduced FIM (obtained by using only the currently visible features) to calculate the BMM parameters. We have also reason to believe that that the visible features contain greater information than the invisible features as depicted in Fig. 2.6.

By ignoring the visible features in (2.14) and by keeping the penalty term constant across the features cost, equation (2.15) is obtained, where cost takes values between 0.8 and 0.95. Thus, at the completion of each maximisation stage, *degenerate* Bernoulli Distributions where the vectors contain either zeros or ones are obtained.

$$P(V_{obs}|\Theta_i) = \prod_{j=1}^{N} (\text{cost})^{|_j V_{obs} - _j\Theta_i|} \tag{2.15}$$

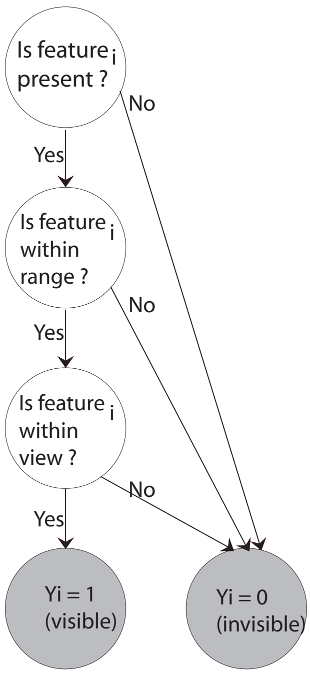


Figure 2.6: An invisible feature is more difficult to explain (more data is needed) as compared to a visible one. While the features that are currently invisible do contain useful data, using only visible features will require comparatively less data.

The entire procedure for comparing a sensor view with the Feature Incidence Matrix can be described as in Algorithm 1. The failure to successfully obtain the parameters of the BMM, results in the setting of the posterior probability of the matching as an uniform distribution.

The initialisation of the hidden data Z is performed by assigning some percentage of the components in Θ to the views in the FIM, V_k. As such these components partially reflect the the composition of the views to which they have been assigned. The remaining portion of the components in Θ are initialised randomly. In this work, no automatic adjustment of the number of components is performed. The number of components is previously fixed and restricted to a fraction of the number of views.

Algorithm 1 Evaluation of the posterior probability of the comparison of a view with the Reference Sequence

V_{obs} = current view (binary column vector).
FIM =Feature Incidence Matrix corresponding to K views V_k from the Reference Sequence.
Θ; % C components of the BMM, initialised with random parameters.
$\alpha = \frac{1}{C}$ % C mixture proportions, initialised with an uniform distribution.
Z; hidden data with C rows and K columns.
$N = 0$; number of iterations of the EM method.
cost $= \{0.8, 0.95\}$; cost penalty term set between $\{0.8, 0.95\}$.
$P_{res} = \infty$; residual probability of view match using current BMM parameters.
while $(P_{res} < \epsilon_{res})$ **do**
  $z_{ki} = \frac{\alpha_i P(V_k|\Theta_i)}{\sum_{j=1}^{M} \alpha_j P_j(V_k|\Theta_j)}$ %Calculate Z, Expectation Phase
  $\alpha_i = \frac{\sum_{k=1}^{K} z_{ki}}{K}$ %Calculate α, Parameter Maximisation Phase
  $\Theta_i = \frac{\sum_{k=1}^{K} z_{ki} V_{obs}}{\sum_{k=1}^{K} z_{ki}}$ %Calculate Θ, Parameter Maximisation Phase
  $P_{res} = 1 - P(V_{obs}|\Theta_i) = \prod_{j=1}^{N} (\text{cost})^{|_j V_{obs} - _j \Theta_i|}$ %Calculate residual error
  if $N = N_{max}$ **then**
    $P(k|V_{obs}) = \frac{1}{K}$; for all k % set uniform posterior
    EXIT
  end if
end while
$P(k|V_{obs}) = \frac{\sum_{c=1}^{C} P(V_k) z_{kc} \alpha_c P(V_{obs}|\Theta_c)}{\sum_{k=1}^{K} \sum_{c=1}^{C} P(V_k) z_{kc} \alpha_c P(V_{obs}|\Theta_c)}$ % Calculate posterior probability
EXIT

Mixture models were created for a FIM obtained from the Reference Sequence of sixty seven images taken with Camera #2 in the Mobile Robotics Laboratory at the Department of Mechanical Engineering, see Fig. 2.5. The posterior probabilities obtained from images captured while leading the robot over a similar path are shown in Fig. 2.7. These posterior distributions are defined over all the views contained in the FIM shown in Fig. 2.5e. Although the posterior probability distribution is not always very narrow, the method works remarkably well given that the images in the Reference Sequence are taken with so much overlap and have so much in common with each other.

The SIFT features work very well and allow the successful recovery of the views in most cases. For some views, however, e.g. in attempts 5, 7 and 8, the single sensor is simply not able to match sufficient features for a successful view index recovery resulting in imprecise and sometimes erroneous view index recovery.

2.4.2 Bernoulli Mixture Model applied to View Sequences with Multiple Sensors

The technique described in the previous sub-section has been extended to the case where the features arise from different, even dissimilar sensors. In an experiment that was performed using three sensors, two cameras and the Laser range scanner, the binary features from the three sensors are combined in the same FIM.

While the earlier subsection demonstrated the usefulness of binary features when these originate from the same sensor, this section demonstrates the application to robot platforms equipped with multiple sensors. Once again, the results are presented in the form of the posterior probability distribution over the views in the Reference Sequence. Representative images taken by Camera #1 are shown in Fig. 2.8, those taken by Camera #2 are shown in Fig. 2.9 and corresponding scans taken by the laser range scanner are shown in Fig. 2.4.

The results of matching are good, given that no additional constraint was imposed on the matching and that the prior probability for matching any image was assumed to be uniform. As seen in Figs. 2.11b and 2.11c, while the features from Camera #1 and Camera #2 are not very useful, the features from the laser range sensor help produce a good estimate when combined with the other sensors. In the case shown in Fig. 2.11a the presence of many unique features in Camera #2 help us to recognize the place correctly.

The effect of using the single camera over the entire Reference Sequence can be seen in Fig. 2.13a.

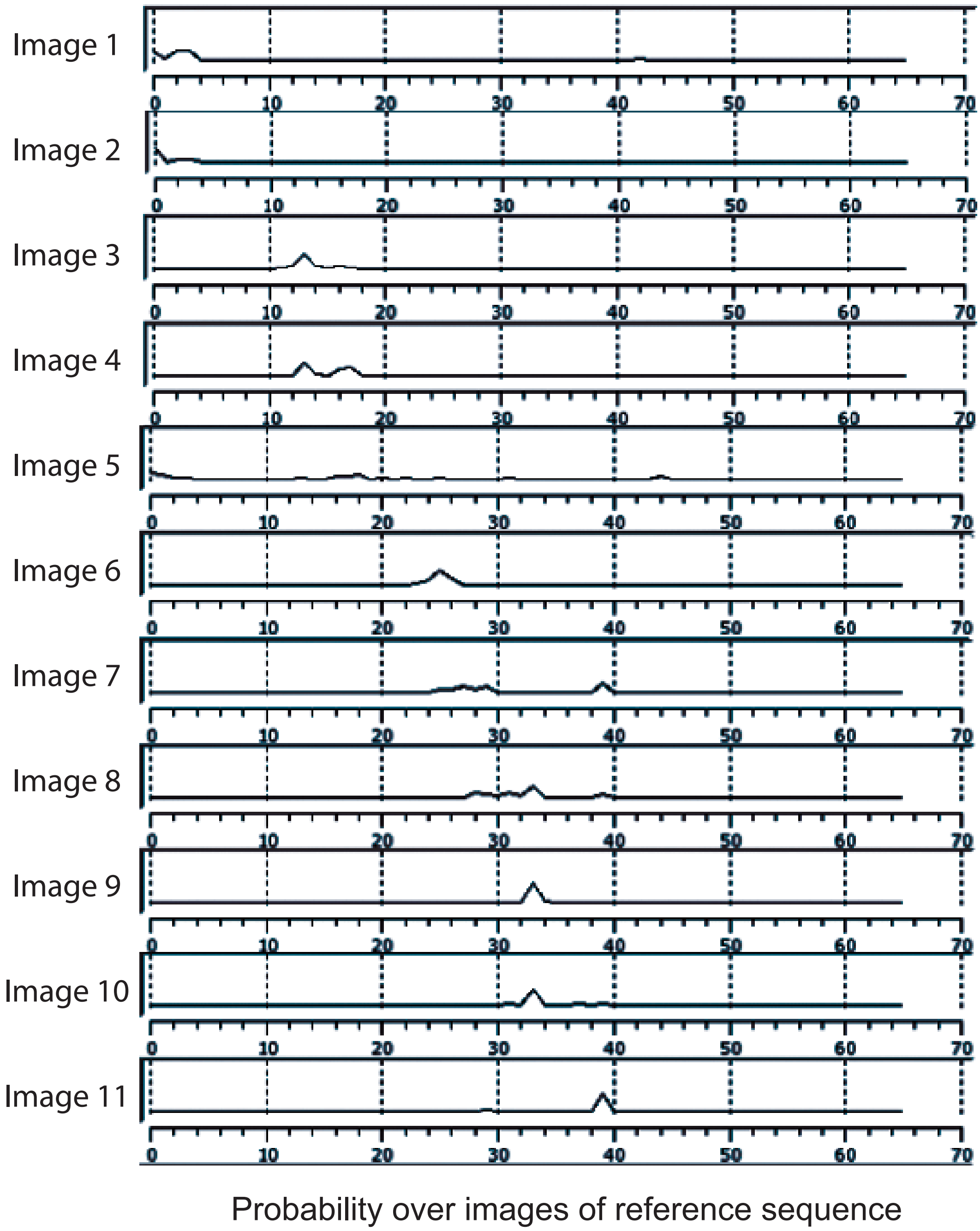


Figure 2.7: The figure shows plots of the posterior probability distribution over each of 67 images that are included in the FIM depicted in Fig. 2.5. The probability is plotted on the y-axis and the x-axis represents the index of the Reference Sequence view.

(a) Image 19.

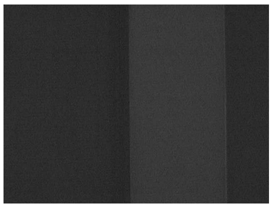
(b) Image 29.

(c) Image 77.

(d) Image 107.

Figure 2.8: Representative images from a sequence of 118 taken by Camera #1 along a hallway for the experiment demonstrating the use of multiple sensors, section 2.4.2.

(a) Image 19.



(b) Image 29.

(c) Image 77.

(d) Image 107.

Figure 2.9: Representative images from a sequence of 118 taken by Camera #2 along a hallway for the experiment demonstrating the use of multiple sensors, section 2.4.2.

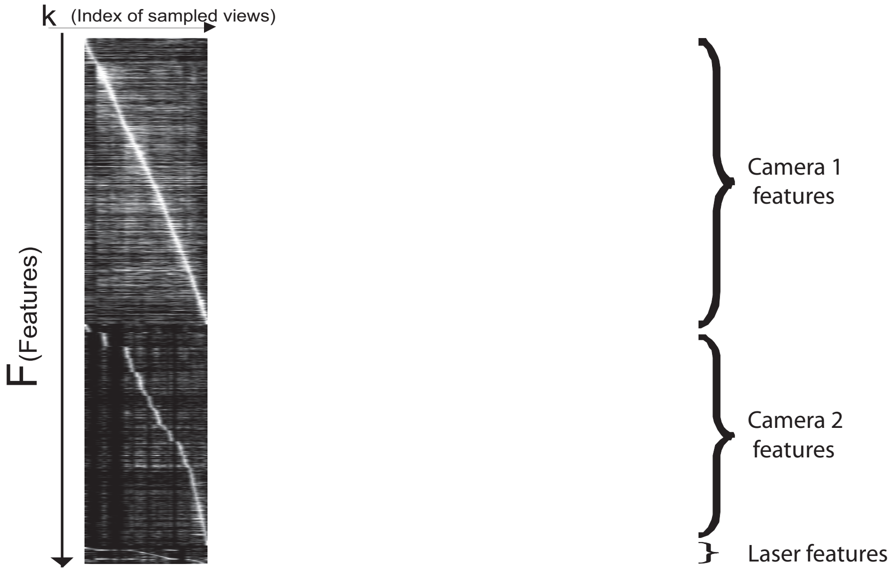


Figure 2.10: The Feature Incidence Matrix for features from the Camera #1, Camera #2 and LRF.

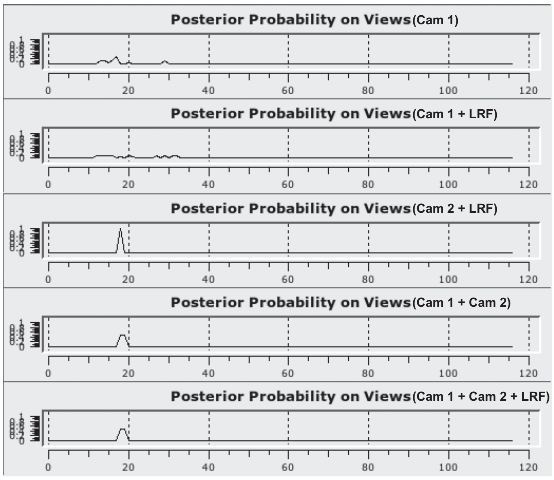
The effect of using the two additional sensors in the recovery of the view index over the entire Reference Sequence can be seen in Fig. 2.13b. It can be seen that there are now fewer views at which the place recognition is not achieved, as compared to Fig. 2.13a.

2.4.3 Bernoulli Mixture Model applied with a Prior View Probability Model

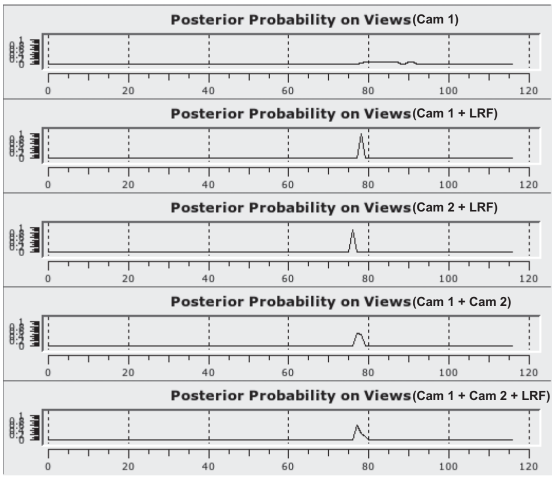
Until this point each view in the FIM has been assumed to be equi-probable. This is the case of the uninformative prior for place recognition. The probability on each view k is defined as $P(V_k)$. The terms $P(V_k)$ in (2.12) is the prior probability that V_{obs} is actually the same as a particular view k in the FIM and up until this point this $P(V_k)$ has been taken from an uniform distribution.

If some views were more likely to be observed than others, this additional information, the prior probability on each view, the $P(V_k)$ term in the right hand side of (2.12) might aid place recognition.

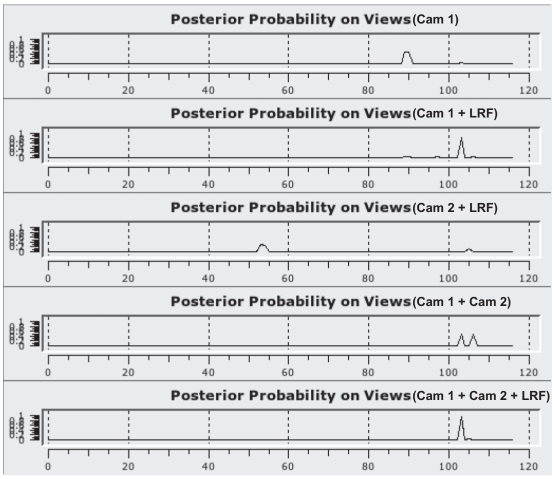
To obtain this prior probability for the next view, a simple robot motion model that updates the probability on every view can be used. This model assumes that the probability of being at



(a) For Image/Scan 9 of working sequence.



(b) For Image/Scan 39 of working sequence.



(c) For Image/Scan 52 of working sequence.

Figure 2.11: Results showing the posterior probability distributions over the views of the Reference Sequence for another sequence with different combinations of sequences. Each figure shows the result of the application for single camera Camera #1, for Camera #1 and the LRF, for the Camera #2 and the LRF, for two cameras Cameras #1 and #2 and, finally, for the Cameras #1 and #2 and the LRF.

any view after localising at a particular view is a function of the estimated amount of elapsed time or distance since the last place recognition attempt.

This probability comes from two distributions: the first distribution models a smooth transition to neighbouring views in subsequent observations and the second distribution tries to regularise the prior probability to account for an erroneous previous place recognition result and to recover from a robot 'kidnap'.

Given the ready availability of odometric data from the robot, the distance rather than the time was used as the parameter for the first distribution. This distribution displaces the current probability distribution on the views directly proportional to the distance covered since the last observation and with an uncertainty inversely proportional to the same distance covered, as described below.

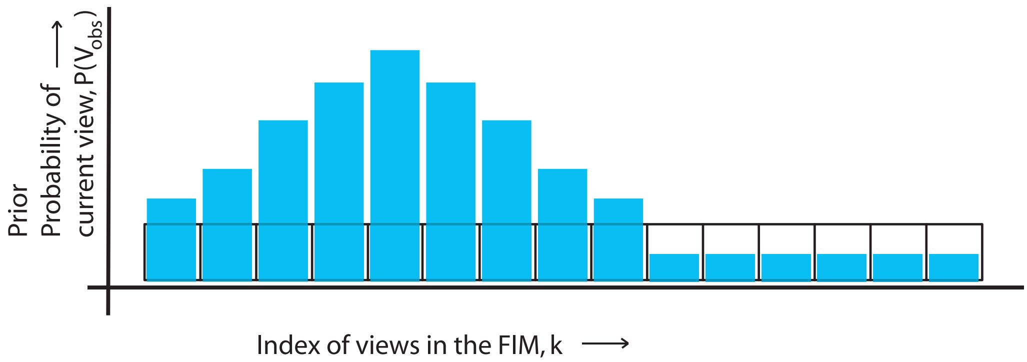


Figure 2.12: Prior probability distribution on the views in the Reference Sequence after the robot after has moved since the last observation.

The second distribution that contributes to the prior is a uniform, non-zero, probability added to all the views to reduce the effect of erroneous past place recognitions (2.16). This probability also reflects the prior that must be applied to the place recognition given that the robot is lost. A generic $Lost\_Place$ refers to the robot starting out from an unknown place.

$$P(k^t|Lost\_Place) = \frac{1}{K} \text{ for all k} \tag{2.16}$$

The probability is then defined as a weighted sum of the above two distributions as in (2.17), where α_{reg} is a parameter. The value of α_{reg} reflects the reliability of a localisation estimate obtained from the motion model and lies between 0 and 1. The resultant prior probability distribution for the current view, on all the possible views in the FIM, after the robot has moved since the last observation can be seen in Fig. 2.12.

$$P(k^t) = \alpha_{reg} \times P(k^t|d_{odo}) + (1 - \alpha_{reg}) \times P(k^t|Lost\_Place) \tag{2.17}$$

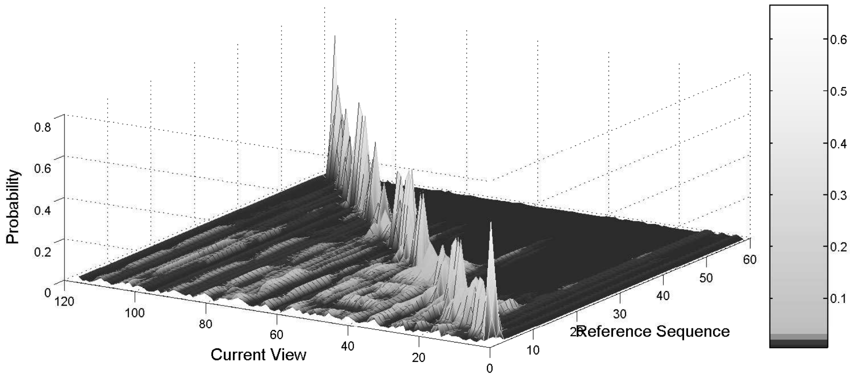
The improvement in the recovery of the view index over the entire Reference Sequence can be seen in Fig. 2.13c. The probability is concentrated quite tightly around the correct views as compared to the earlier case in which no model for the robot motion was utilised.

2.5 An Insight into the Information Provided by Multi-feature Views

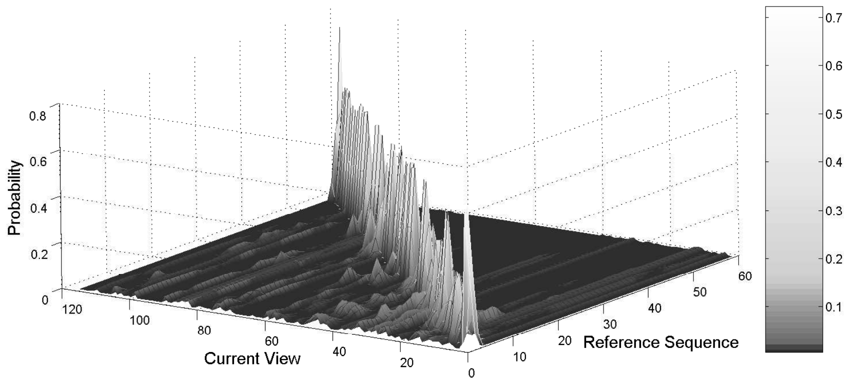
The Localisation within the Reference Sequence might be viewed from an Information Theory approach (see [Cover 91] for an excellent introduction to Information Theory). There is a uncertainty associated with position occupied by the robot in the Reference Sequence. Each View provides some information in the form of multiple features that could potentially reduce some of the uncertainty of the location within the Reference Sequence.

Let V denote a generic view in the Reference Sequence and it can take the values $V_1, V_2, ...V_K$ denoting one of K distinct and detectable Views in the Reference Sequence. The presence and the absence of individual features within each V is correlated with the position k of the robot in the Reference Sequence. This correlation may not be perfect however (the function that expresses the values of the features in terms of the view number is not injective), indicating that the same combination of features might be obtained at more than one view V_k in the Reference Sequence. The 'mutual information' term $I(V_k; k)$ denotes the overlap of information between V and k, i.e. it specifies how much the appearance of View V reduces the uncertainty of the robot position k. The way in which individual views can contribute to the reduction in the uncertainty of the position within the Reference Sequence is shown graphically in Fig. 2.14.

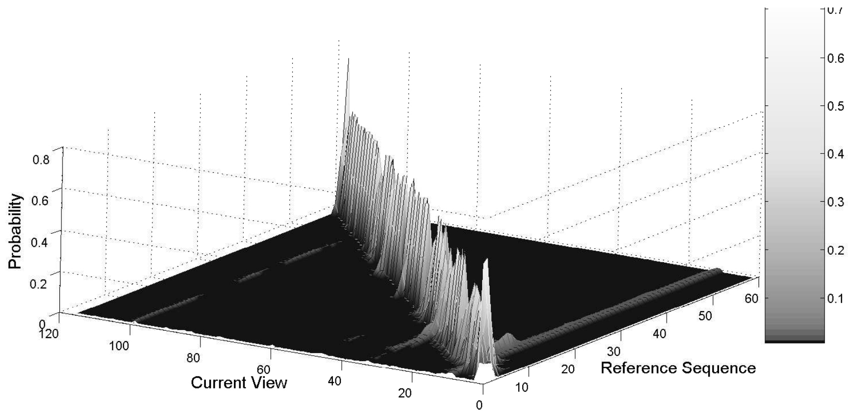
From the image in Fig. 2.14 it can be seen that the best results will be obtained when the



(a) Cam 1 only



(b) Cam 1 + Cam 2 + LRF



(c) Cam 1 + Cam 2 + LRF + robot motion model

Figure 2.13: These figures present an overview of the results obtained in sections 2.4.2 and 2.4.3. The figures show the posterior probability distribution over the Reference Sequence when the robot is driven along the same path over which the Reference Sequence was obtained, a second time.

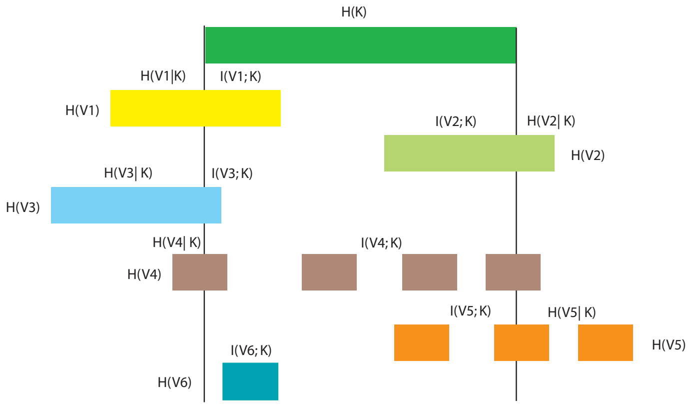


Figure 2.14: Multiple features potentially allow a reduction in the uncertainty in the Localisation within the Reference Sequence. This figure describes how individual features fs might aid in the reduction of k by being correlated with different parts of the distribution of k (different views of the Reference Sequence)

information provided by each of the views overlaps the least.

2.6 Summary

Robust place recognition has been obtained by resorting to sensor views containing a large number of features. It is important to highlight that the role of the Bernoulli Mixture model in this work is to reduce the dimensionality of the views and to describe the environment in some reduced 'space of features'. It should be easy to view this utilization of a mixture model as part of a solution to the generalised 'Data association problem'.

The addition of multiple sensors improves the place recognition capability of the robot immensely. In the example described in the earlier sections, SIFT features seen by forward-facing Camera #1 and located deep in the hallway persistently appear in the views and crowd out less salient, but possibly more useful, features appearing at the periphery of the images. In situations where localisation using this sensor fails, features from Camera #2 and from the LRF still allow for successful place recognition.

The place recognition method described in this chapter treats all the features in the same way. The Bernoulli Mixture model works by apportioning the individual components between blocks of features so that the variation can be accommodated. In the presence of features from multiple sensors, the components of the Bernoulli Mixture Model are especially sensitive to the much larger number of vision features (numbering in the hundreds per image) as compared to the LRF features (each scan results in a few tens of features).

While the results seem to be promising and sufficiently robust for application to localisation, improvements to the method are still required. In particular, methods to reduce the number of features, before application of the EM algorithm will result in the reduction of the time required to calculate the parameters of the BMM.

Chapter 3

Sequential Context of Views on a Path

3.1 Introduction

In chapter 2 the approach utilised to represent a view using binary features was presented. This approach utilised the information associated with matching a *single* view against the Reference Sequence to infer the location of the robot. But the Reference Sequence can yield additional information, information that is implicit in the ’order’ in which the views occur. This chapter aims to extend the method in chapter 2 by modelling this sequential order of the Reference Sequence and then simultaneously matching multiple views. The use of this ’contextual’ information of the Reference Sequence is depicted in Fig. 1.11. This chapter seeks an answer to the second question posed in section 1.3.1; How can the information that is inherent in the order of the sequence of observations aid the inference of the position of the robot in the Reference Sequence?

The information contained in the sequential context of the Reference Sequence is useful because, by using a single view, it might not be enough to disambiguate between similar places lying on a Reference Sequence. If a sequence of current views are to be used for localisation, the problem in chapter 2 modifies into a problem of aligning this sequence, called the query or localisation sequence, with sections of the Reference Sequence. The method that is used to model this additional information, a Hidden Markov Model, will exploit the sequential context information in two ways:

1. It will provide a consistent framework to use the prior probability of matching each view in the Query Sequence, and

2. Consequently, it will prevent inconsistent or unfeasible sections of the Reference Sequence from explaining the sequence of views in the Query Sequence.

Over the next few sections of this chapter, the problem of recovering the current position of the robot in the Reference Sequence is modelled in terms of a well-known and often utilised version of a Bayesian Network, the Hidden Markov Model (HMM). An original layout for the Markov Chain that is used to represent the hidden states in the HMM was developed.

Most autonomous robot localisation systems assume that the number of possible changes in the position of the robot, between observations, are finite and that future positions that could be occupied by the robot are predictable (probabilistically speaking). These systems assume that the robot motion is governed by a motion model and this model can be used to provide an estimate of the prior probability of the robot being at different places in the environment, just prior to making the next observation.

Usually, the distance covered by the robot since the last known position, as measured by the odometry or inertial sensors, is used as an unbiased estimator of the distance between the states. In the section 2.4.3, the distance covered by the robot since the last observation was utilised to generate an *informative* prior distribution that improved the place recognition results of each fresh observation (view). The motion model that was utilised in that section set the prior probability of the robot being at a particular place according to the distance covered by the robot since the last observation, Fig. 2.12.

This approach is actually using the sequential context information that exists between pairs of views when these are matched against the Reference Sequence. This happens thus:

1. The robot is assumed to be lost before the first observation.

2. The posterior probability after the first observation is obtained by applying the uniform prior in (2.12).

3. After the robot has moved and just before the next observation, the robot motion model is used to alter the posterior probability at the time of the last observation to provide an informative prior probability for the new observation.

4. The posterior probability is obtained by applying the informative prior once more using (2.12).

5. Repeat from 3.

An extension of this procedure is desired so that a query sequence that is longer than 2 observations.

In the next section a review of the application of HMMs to other areas, not related to robot Localisation is presented. In section 3.3, details regarding the calculation of the parameters required to model the Reference Sequence as an HMM, namely the transition between hidden states and the probabilities of observation will be presented. In section 3.4, results are presented with emphasis on the comparison with the results of place recognition without resort to HMMs. Finally, a discussion about why the HMM is an appropriate model for use with the Reference Sequence is initiated.

3.2 Sequence Matching using Hidden Markov Models

A Hidden Markov model (HMM) is a stochastic model of a process that can take a fixed number of mutually exclusive states. These states are taken to correspond to the hidden nodes of a Markov chain. The system is measured at a finite number of instances, where the time variable takes some discrete values.

As depicted in Fig. 3.1, the initial state is the state of the process at $t = 1$ and the final state is the state of the process at $t = T$. At every subsequent time, between 1 and T, the system jumps to another state or remains at the current state. Transitions are represented as arcs between the nodes and there is a probability associated with each such transition, called the state-transition probability. Impossible transitions between nodes have a probability of zero.

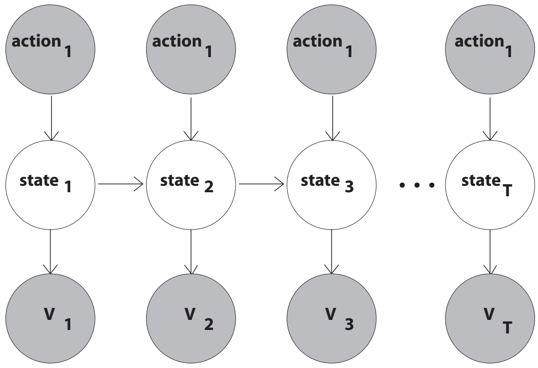


Figure 3.1: A classic representation of a HMM for Localisation in a Reference Sequence showing the [same] action that will propel the robot from one state to another until it comes to state 't', the current state.

While the system occupies one of the mutually-exclusive states at any one of the intervals from 1 to T, this fact is not visible and the states are said to be hidden. However, an output measurement is produced for each time T and this output is observable. The sequence of output vectors, O = ($O_1, ..., O_T$), is referred to as the observation sequence. This arrangement of hidden states and observations, seen in Fig. 3.1, is referred to as the Moore model as opposed to the Mealy model in which observations are visible during the transitions between the hidden states.

$$\lambda = \langle N, M, \{\pi_i\}, \{a_{ij}\}, \{b_i(n)\}\rangle \tag{3.1}$$

The notation used in Rabiner's [Rabiner 89] tutorial on HMMs for speech recognition has been utilised. The parameters λ of the HMM are specified as in (3.1), where M corresponds to the number of possible states, N the number of observation symbols, π represents the initial probability on the states, the a_{ij}s correspond to the transition probabilities between a pair of states i and j and $b_i(n)$ represents the probability of viewing symbol n at state i. Rabiner [Rabiner 89], also specifies three basic problems to be solved with HMMs, as reproduced in Fig. 3.2. This chapter deals with the second type of problem: what is the probable path (defined in terms of the sequence of views in the Reference Sequence) that the robot took, given what the robot has seen so far?

Problem 1: Given the observation sequence $O = O_1 O_2 \cdots O_T$, and a model $\lambda = (A, B, \pi)$, how do we efficiently compute $P(O|\lambda)$, the probability of the observation sequence, given the model?

Problem 2: Given the observation sequence $O = O_1 O_2 \cdots O_T$, and the model λ, how do we choose a corresponding state sequence $Q = q_1 q_2 \cdots q_T$ which is optimal in some meaningful sense (i.e., best "explains" the observations)?

Problem 3: How do we adjust the model parameters $\lambda = (A, B, \pi)$ to maximize $P(O|\lambda)$?

Figure 3.2: Rabiner's [Rabiner 89], description of problems for HMMs.

3.2.1 Application of HMMs to Robot Localisation problems

In the field of Robot Localisation, probabilistic methods have become the de-facto standard given their flexibility and the development of Bayesian Inference methods that have popularized the use of Bayesian Networks. Although it is quite impossible to exhaustively mention even the most important contributions in the field, the following works are mentioned because of similarities to the work presented in this thesis [Kosecka 04] [Theocharous 04] [Rachlin 05] [Nikovski 02] [Fox 99].

3.2.2 Application of HMMs to problems unrelated to Robot Localisation

One of the earliest application of Bayesian Networks in the form of HMMs was to solve the problem of speech recognition. Rabiner's tutorial [Rabiner 89] continues to provide a good introduction to the problem of speech recognition (and to the application of HMMs in general). The area has also seen the application of a number of modifications to the vanilla HMM to account for the complexity of the problem and the robustness that is required of any successful application. Modifications include the use of continuous variable duration HMM [Levinson 86]. The recognition of passages of music or the retrieval of the complete music using a shorter passage is another extension of this problem [Pikrakis 06].

Recognising what is written by one person or by multiple persons hand is a difficult problem because of the ambiguity that is introduced in the actual writing procedure. Besides errors in the spelling and the variations that are due to the persons individual style of writing, there is the difficult problem of identifying where a character ends and where another one begins. Besides the methods developed originally for speech recognition, one type of technique that has resulted in considerable success in the recognition of handwriting is the so called Variable Duration HMMs. In the handwriting recognition applications this technique allows for increases in the distance between hand written characters [Chen 95]. Other extensions of the variable duration HMMs can also be viewed as more complex Bayesian Networks [Vlontzos 92]. These techniques are built upon variations of the vanilla HMM model to perform an explicit modelling of the duration for which the system remains at the same state and allow for the use of larger dictionaries and impose fewer restrictions on the handwriting styles that can be recognised [Kundu 98]

The importance and sheer scale of the Human Genome Mapping project provides another example of the importance of the application of HMMs in Biology. The problem of matching described earlier is known in this field, of Bio-informatics, as the *protein sequence alignment* problem. While there have existed alternatives to the HMM profile-based methods such as pair-wise matching and Neural-Network based methods, HMMs ‘now provide a coherent theory for profile[matching] methods’ [Eddy 98]. With the human genome completely sequenced, profile-based (HMM) methods have often been used to compare a DNA sequences of as yet unknown function with others having known function [Conant 04].

The problem of sequence alignment of protein sequences actually bears much resemblance to the problem of robot Localisation within a sequence of observations. These are enumerated below:

- A protein sequence, as in the case of the Reference Sequence, is seen as a first-order Markov Chain where only the symbol sequence (corresponding to the view sequence) is visible. The chains are modeled as directed (left-to-right) graphs, where the transition between ’consensus nodes’ are represented using a transition probability Matrix.

- Profile HMMs allow the incorporation of information regarding the predisposition of protein sequences to contain spurious sub-sections (called insertions) as well as missing proteins (called deletions). Given that the Reference Sequence is composed of sampled views of the environment, insertions are expected in between the views in the Reference Sequence. Additionally, limitations in the perception process, incorrect executions of the required behaviours, and possible changes in dynamic environments might result in an unmatched (reference-sequence) view.

- Profile HMMs are suited to short-term correlations in the views [Eddy 98] rather than long-term correlations. This would be analogous to applying the HMMs to detect Localisation using short-term correlations but not to the problem of loop-closures.

- As in the case of the localisation with the Reference Sequence problem, the profile HMM method forces us to assume what would be the emission probabilities for the insertion states. The insertions are assumed to result in any of the views in the Reference Sequence with a uniform probability.

3.3 HMMs applied to Place Recognition in the Reference Sequence

At this point, we formalise a concept of the Reference Sequence as a connected graph. Each place in the environment that produced a view in the Reference Sequence is represented as a node in this graph.

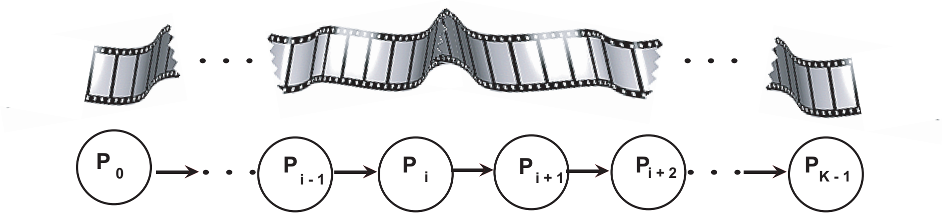


Figure 3.3: This figure depicts Reference Sequence in the form of a left-to-right graph composed of 'K' views in the order in which they were sampled during the Environment Familiarisation phase.

Transitions between the places are indicated by edges that connect pairs of nodes. The Reference Sequence was created by leading the robot along a path in the environment. As a result of this procedure, the graph is connected in a special way. Each node/place in the graph is connected to two other nodes/places. When we consider the sequential order of the Reference Sequence, there is one edge that could bring the robot to a particular node/place and another that takes the robot to the next node/place.Such a graph is called a straight, directed graph or a left-to-right graph as depicted in Fig. 3.3.

$$
\begin{array}{cc}
 & \text{(...to...k)} \\
\text{(From ...k)} & \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}
\end{array}
$$

Figure 3.4: The single-step transitional probability matrix for the Environment Familiarisation phase indicates a single known transition between views or places k along the Reference Sequence.

In this left-to-right graph (with no loops and overlapping edges), a transition probability matrix can be created to account for all the possible transitions between places/nodes in the graph. During the Environment Familiarisation phase, when the robot moves along the path which generates the Reference Sequence, the transition probability matrix might appear as in Fig. 3.4 since the only information that is known is the single-step transitions that are possible.

During localisation, the robot attempts to repeat these transitions as it moves along the path. This might not always be possible since the robot might skip a place and find itself situated at a node/place that lies later in the Reference Sequence. The rest of this chapter addresses these problems by proposing modifications in the left-to-right graph and detailing the application of the Hidden Markov Model, to find the most likely place in the Reference Sequence that is currently occupied by the robot.

The sequence of observations that is currently available, and which must be matched against the Reference Sequence is called the Query sequence or the Localisation sequence. These terms have been used interchangeably in this chapter and in the rest of this document.

The HMM has been applied to mobile robot localisation by a large number of researchers. Its popularity has been due to two convenient properties that can be incorporated in the HMM. The first property is that the position occupied by the robot can be modelled as a hidden variable (not directly observable state) which must be inferred from the outcome of another, observable variable, the sensor data. The second property is that this hidden variable is assumed to have the Markov property, where future states that the system occupies are dependent on the current state and independent of past states. The first property is useful when we assume that sensor data at each point is not unique as a result of which a prior probability is required to make the inference of the robot position. The second property applies quite well to the problem of robot localisation since a robot will usually move smoothly in an environment. It is also desirable because it makes the whole inference problem tractable by not requiring past positions to be considered.

A problem with the application of HMMs that is often not addressed by robot localisation systems is that the HMM might not make unbiased estimates when the sensor data at different places is highly correlated. The fact that a first-order Markov Model cannot, in general, capture the relationship between two observations [Dietterich 02] can pose a problem in Mobile robot Localisation. In our method, the dimensionality reduction method that has been previously employed to capture the correlations between the views in the Reference Sequence in chapter 2 is re-utilised as the observation model of the HMM. This utilisation of the Bernoulli Mixture model, BMM, as a dimensionality reduction method is expected to deal with the above problem and reduce its severity.

In subsection 3.3.1, the way to create the Reference Sequence is shown. The application of a Hidden Markov Model (HMM) to model this sequence so that a Dynamic Programming algorithm, the Viterbi algorithm, can be used to match the observation in the context of the Reference Sequence is shown.

3.3.1 Modifying the Graph of States for Place Recognition

The creation of the Reference Sequence during the Environment Familiarisation phase was previously defined as a sampling procedure. It is important to note that the place recognition procedure is also a sampling procedure. The 'sampling rate' at the time of localisation may be different from that during the Environment Familiarisation phase. The acquisition of views might be controlled by the distance travelled by the robot- a noisy quantity that is not always controllable in certain environments. Additionally, the Reference Sequence assumes that a behaviour executed at each view propels the robot to the next view in the Reference Sequence. Actions other than the one executed in the Reference Sequence will result in the robot seeing things differently as compared to the Reference Sequence. A view-based localisation method for a robot must recognise when the robot is lost.

Keeping this in mind, the Markov chain that represents the places in the Reference Sequence in the Environment Familiarisation phase has been modified to include additional nodes/places and some additional edges or transitions. This Markov chain is termed the Original Reference Sequence. The modified Markov chain for application to localisation, with some additional states, a fragment of which is shown in Fig. 3.5, is called the Complete Reference Sequence.

Any action other than the one performed in the Reference Sequence will take the robot to a some previously unvisited Place. To the original set of K nodes, another K '$Lost\_Places$' are added so that the possible values that the state can take are now described in the discrete by $M = 2 \times K$, which includes the K originally sampled Places and the K $Lost\_Places$ (there is a $Lost\_Place$ before the first state too). As a result the HMM depicted in Fig. 3.3 will be modified to Fig. 3.5. It is important to note that, if inference is to be made across multiple actions or behaviours, it will be reflected in terms of an increased complexity of the graph in Fig. 3.5. The '$Lost\_Places$' perform the following function:

1. They take into account the fact that the robot might see views that are not included in the Reference Sequence.

2. They indicate the last position at which robot was localized, directly indicating the probability of the robot being lost.

The Complete Reference Sequence with its $Lost\_Places$ is actually quite similar to some of the models used for profiling gene sequences [Durbin 98]. An example of a 'profile HMM' used in such application is shown in Fig. 3.6. The '$Lost\_Places$' correspond to the 'insertion' nodes in the profile HMM.

3.3.2 Applying the HMM on the Reference sequence

The Hidden Markov model is completely defined in terms of the three probability distributions: the transition probability matrix, the observation or emission probability matrix and the initial probability matrix.

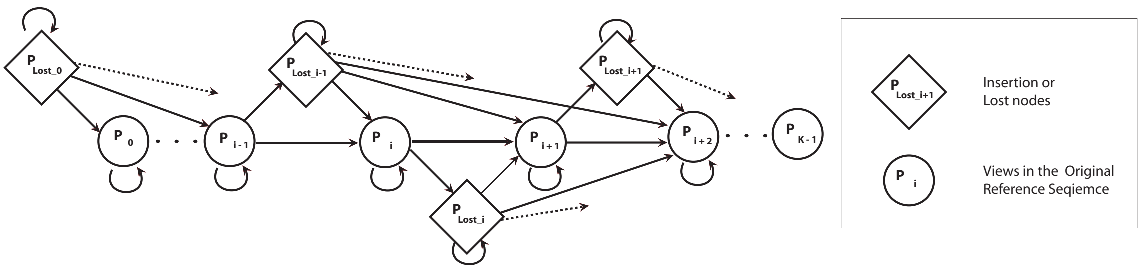


Figure 3.5: The figure depicts a modified Markov chain, with '$Lost\_Places$' inserted within the original Reference Sequence, to perform place-recognition. The dotted lines indicate the transitions to each of the Places in the original Reference Sequence which have not been drawn to avoid cluttering the figure.

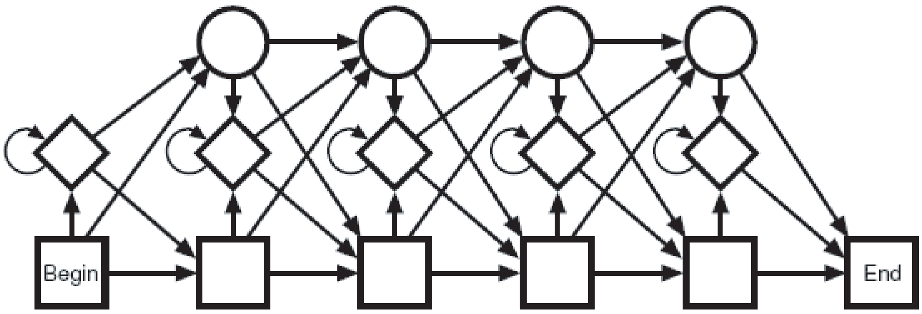


Figure 3.6: The figure describes a profile HMM [Durbin 98]. The lower-most layer of states correspond to the 'main' states, the diamond-shaped states represent the 'insertion' states and, the circles represent the 'deletion' states. The profile HMM is somewhat analogous to the Markov chain used to represent subsequent travel along the Reference Sequence where previously unseen views are observed.

The transition probabilities on the places, the a_{ij} s

The transition probabilities reflect the feasibility of moving from one place to any another place in the Reference Sequence. The transition probability is always conditioned on the possible behaviour that will take the robot to the next state or to on of the $Lost\_Places$. The transitions that are allowed from state are to the next state in the Reference Sequence and to the $Lost\_Place$ lying after the current state. If the robot is allowed to transition from each place to the neighbouring places and to the respective $Lost\_Place$, a transition probability matrix is drawn up, which can be represented in terms of Fig. 3.8b, where white represents a transition probability of 0 and black a transition probability of 1.

A robot motion model is used to apply a probability on every view, as a function of the estimated amount of travel that the robot has completed while performing a particular behaviour. The probability on each view k as $P(V_k)$ and the value of the probability on the view at any time t is defined as $P(k^t)$. The probability for each view k, $P(k^t)$ is then used as a prior probability before actually matching the view.

A model that modifies the probability distribution of the internal state Fig. 3.7. The vertical line through the bars indicate the value of the odometry as obtained by the sensor. The wider bar indicates the region in which the robot might be with higher, uniform, probability, while the thinner bar indicates the region that the robot has crossed or not arrived at with lower probability though non-zero probability.

The first distribution displaces the current probability distribution on the views as in (3.2), where d_{odo} is the distance covered since the last observation and $d(i-j)$ is the distance between place i and j.

$$P(i^t|d_{odo}) \quad = \quad \frac{\sum_{j=1}^{K} P(j^{t-1}) \times d(i-j)/(2 \times d_{odo})}{\sum_{i=1}^{K} P(i^t|d_{odo})}$$

$$\text{if } |d(i-j)| < d_{odo}$$

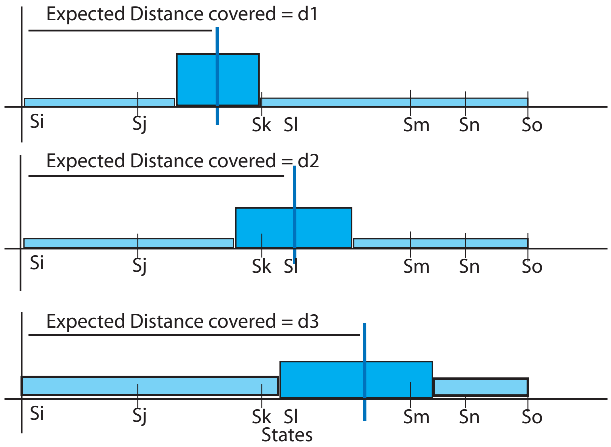


Figure 3.7: A model that for the probability distribution of the internal state.

$$P(i^t|d_{odo}) = 0 \text{ otherwise} \tag{3.2}$$

$$P(k^t|Lost\_Place) = \frac{1}{K} \text{ for all k} \tag{3.3}$$

$$P(Lost\_Place^m|Lost\_Place^n) = 0 \text{ ...for all m} \neq \text{n} \tag{3.4}$$

$$P(Lost\_Place^t|Place^{t-1}) = 0.5 \tag{3.5}$$

$$P(Place^t|Place^{t-1}) = 0.5 \tag{3.6}$$

With regards to the $Lost\_Places$, the sequence begins with $P_{Lost\_Place\_0}$ which indicates that the robot is completely lost or has never localised. Also, before every original place P_i, there is a $P_{Lost\_Place\_i}$. By moving forward from one $Lost\_Place$, the robot can transition from $P_{Lost\_Place\_i}$ to any node P_k where $k > i$. Similarly, from P_i the robot can transition to $P_k : k > i$ or to $P_{Lost\_Place\_i+1}$. The graph does not allow a single-step transition from one $P_{Lost\_Place\_i}$ to another $P_{Lost\_Place\_j}$.

The observation model, the $b_i(n)$ s

The Bernoulli Mixture model from chapter 2 is used as an observation model for the HMM. The components of the mixture model are estimated and the current view is compared to these components. The Bernoulli Mixture Model when used as the observation model, has two particularities:

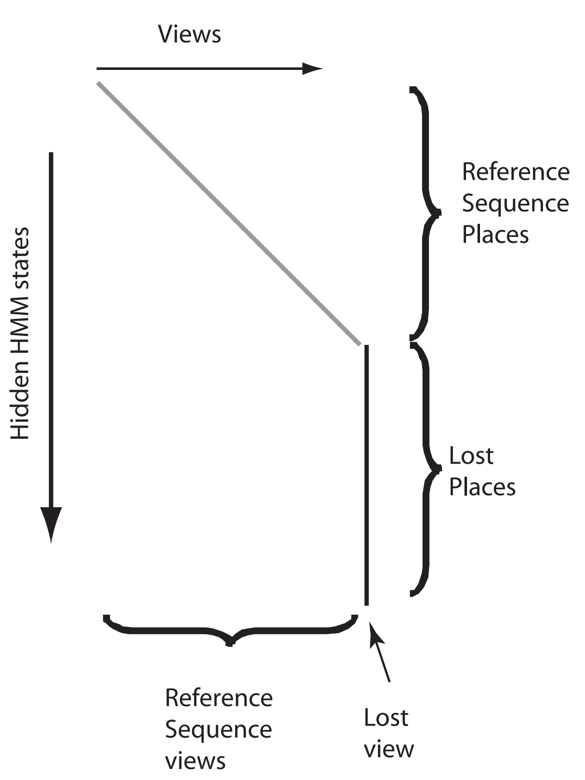
1. The observations '$O_i(n)$' at each and every state of the HMM are defined in the same space as the sampled views (3.7)from the Reference Sequence. This is not a property of HMMs but rather a result of the adoption of the Bernoulli Mixture model for matching pairs of views. Since the posterior probability for the Bernoulli Mixture Model is calculated in the 'space' of the views of the Reference Sequence and not in the space of the components, the observations have the corresponding number of events as in K.

2. Since the environment is sampled, not every observation might be assigned to a place in the Reference Sequence. Thus, when the robot is between places that were sampled earlier, it sees an un-defined view. This undefined view is called the 'Lost View' and any place that generates this view is defined as a '$Lost\_Place$' Fig. 3.5. In the absence of any information, the observation probability is arbitrarily defined over the set K at any $Lost\_Place$ as an uniform distribution (3.8). This statement is made on the assumption that the distribution of the features is maintained throughout the environment.

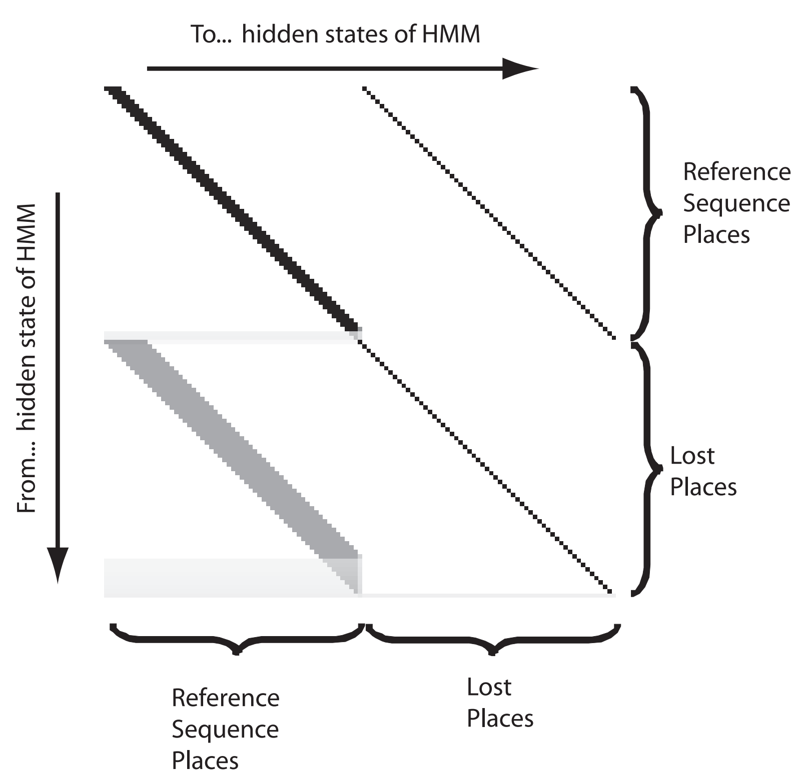
$$O_i(n) \in K \tag{3.7}$$

$$b_{Lost\_Place\_i}(n) = \frac{1}{K} \tag{3.8}$$

In the speech analysis field, given a set of training sequences, a standard way of obtaining the Observation Probability Matrix is to assume that they correspond to a normal distribution, each observation being centered around the symbol that is phonetically most similar to the underlying states, see Fig. 3.8a. This approach is acceptable because there exists previous information that



(a) The Emission Probability Matrix.



(b)

Figure 3.8: The figure at left graphically depicts the observation probability or emission matrix for the Hidden Markov Model, HMM. Besides the views from the Reference Sequence, a Lost View appears as an additional observation that is ’seen’ whenever none of the other views are seen (in practice, wherever the entropy of the posterior probability over the views of the Bernoulli Mixture Model is greater than some arbitrary threshold). This matrix is regularised (probabilities made non-zero) to avoid over-fitting to the Reference Sequence. At right, is the image of the Transition Probability matrix for the HMM. From each of the places in the Reference Sequence, the matrix allows transitions to the near neighbours. From the $Lost\_Place$, transitions are allowed to the places in the Reference Sequence. The transition probabilities are obtained from a simple model of the robot motion as explained in section 3.3.2

.

the phonemes are reasonably distinct since they are utilised in human language and do not come from sound clips with an arbitrary sequence of frequencies and amplitudes.

In the case of Place Recognition, it is not possible to ensure that different places in the environment will be sufficiently different. In this case the observation probability will be obtained by comparing the sequence of views with itself. Additionally, given that each View can be composed of hundreds of features, and given that the results of matching depend on which features are absent, it is not feasible to take into account all the emission probabilities for all combinations of features.

The emission probability matrix has been created using a distribution that is based on the similarity between views. Thus, if a view is different from all others, the emission probability for a robot at a position giving that view is very certain. On the other had if a particular view is similar to a number of other views, then the emission probability for the robot occupying that place will be spread among the various similar views.

This distribution is regularized to ensure that matching will occur despite the fact that some noise might exist in the observation vector.

The initial Probabilities State

At the start of the matching procedure or when the robot is completely lost, the robot always departs from the first place.

$$P(Lost\_Places_0^0) = 1 \tag{3.9}$$

It is natural that the probability at the beginning of the Localisation be concentrated in the first $Lost\_Place$, before the first state. Thus, the first hidden state is always matched to the $Lost\_Place$ $P_{Lost\_Place\_0}$. This first $Lost\_Place$ has a non-zero probability associated with the robot reaching each place in the original Reference Sequence, i.e. $P_0, P_1, \ldots, P_{K-1}$.

In the context of this work, where a mixture model is used to generate the observations, the above arrangement has some useful properties since, a concentration of the probability in the first $Lost\_Place$, $P_{Lost\_Place\_0}$, results in an uniform probability distribution over the places

1) Initialization:

$$\delta_1(i) = \pi_i b_i(O_1), \qquad 1 \leq i \leq N$$

$$\psi_1(i) = 0.$$

2) Recursion:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \qquad 2 \leq t \leq T, \quad 1 \leq j \leq N$$

$$\psi_t(j) = \operatorname*{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \qquad 2 \leq t \leq T, \quad 1 \leq j \leq N.$$

3) Termination:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = \operatorname*{argmax}_{1 \leq i \leq N} [\delta_T(i)].$$

4) Path (state sequence) backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \cdots, 1.$$

Figure 3.9: The 4 steps of the Viterbi Algorithm

$P_0, P_1, \ldots, P_{K-1}$ for the Bernoulli Mixture model.

3.3.3 Estimating the likely sequence of Places

The Viterbi algorithm is commonly used in the context of HMMs to determine the most probable sequence of hidden states that gave rise to a particular sequence of observations. In the context of the profile HMMS, where the HMMs are used to solve the problem of identifying substrings, the Viterbi algorithm is employed to find the 'best alignment', or the one with the highest probability, between two fragments of a string.

The algorithm itself is a type of Dynamic Programming algorithm [Forney 73], [Rabiner 89]. By focusing on only one hidden state at a time, the Viterbi algorithm calculates all the outcomes that could be possible for that state - and then keeps only the most likely one. After traversing the length of the HMM, the 'surviving' sequence is the sequence that is most likely to have generated the observation.

The algorithm is described in terms of 4 steps in the tutorial by Rabiner [Rabiner 89] and

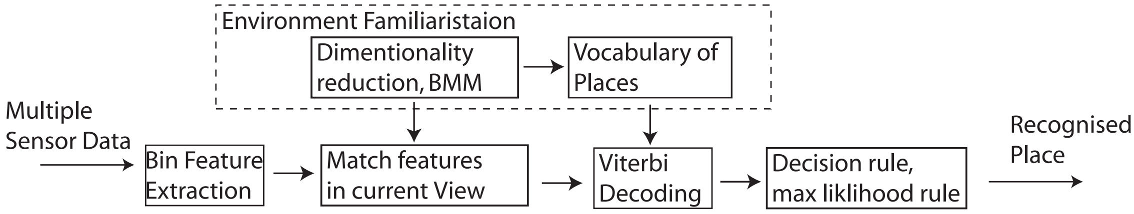


Figure 3.10: A schematic of the Algorithms used to perform place recognition

shown here in Fig. 3.9.

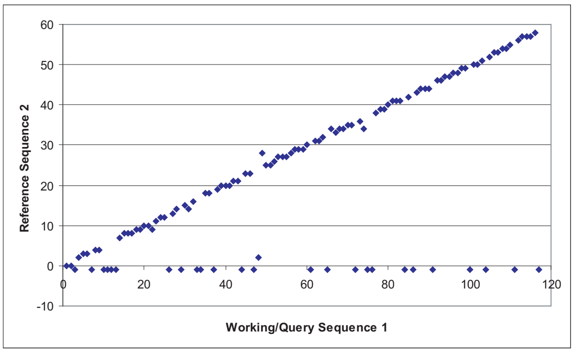
3.4 Experiments and results

The complete Place recognition algorithm is now built around the HMM. The HMM used the BMM as part of the observation model to perform dimensionality reduction. The schematic of the process is depicted in Fig. 3.10.

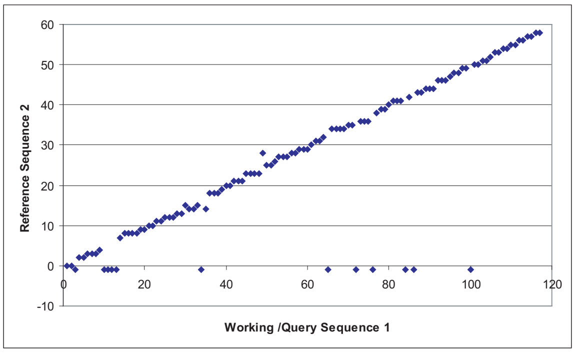
The results of applying the Viterbi algorithm for a set of 4 place recognition experiments are compared in Table 3.1. The Hidden Markov model that was 5 observations long. Only a single sensor, Camera #1 has been utilised in the place recognition experiments in this section so as to illustrate the improvement in results obtained by using a HMM. In the results presented in the table, two Reference Sequences, taken at very different times of the day (consequently in different lighting conditions) are used. As can be seen the percentage of successful place-recognition attempts varies greatly since they depend on the changes that occur in the environment.

The results of place recognition for individual runs of the place-recognition attempts are shown in Figs. 3.11 and 3.12. The results of place recognition are shown for three different situations. In the first situation, the place recognition is performed using the Bernoulli Mixture Model under an uniform prior distribution assumption. In the second case the same Bernoulli Mixture model is utilised, but this time an ’informative’ prior is applied. In the final situation, place recognition is performed using the Bernoulli Mixture model as the observation model of the Hidden Markov Model. In this last case, a prior probability is also available to improve the results of the Bernoulli Mixture Model.

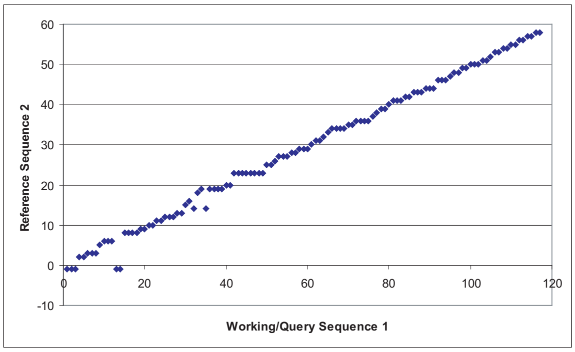
The results in Figs. 3.11 are obtained when place recognition was performed under rather



(a) Place recognition using a Bernoulli Mixture Model with a Uniform prior probability over all views of the Reference Sequence for matching each observation. At least 24 failures were recorded from a total of 117 observations.



(b) Place recognition using a Bernoulli Mixture Model with an informative prior probability for matching each observation. A total of 12 failures were identified from a total of 117 observations.



(c) Place recognition using a 5-observation-long Hidden Markov model. Each observation is matched using a Bernoulli Mixture Model with a prior given by the transition probability of the HMM. 6 failures were correctly identified from a total of 117 observations.

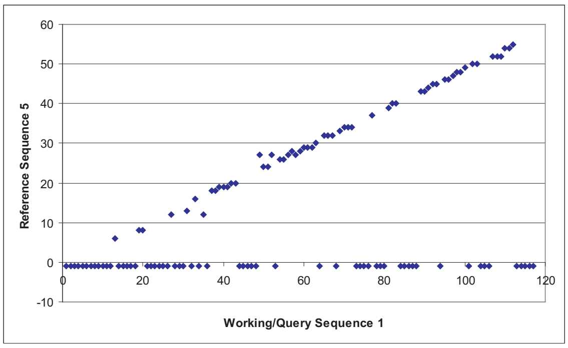
Figure 3.11: Comparison of place recognition with and without recourse to HMMs. The results represent two sequences taken in daylight, at times separated by an hour. Only Camera #1 and Camera #2 were utilized in these sequences.

similar conditions. The Reference Sequence was created at 16:00 along a hallway with natural lighting and the Query sequence for place recognition was obtained a couple of hours later. The environment was, on the whole, well-lit and there were very few changes to the environment (isolated persons walking down the hallways). It can be seen the reasonable results obtained using the simple Bernoulli Mixture are much improved when a HMM is introduced.

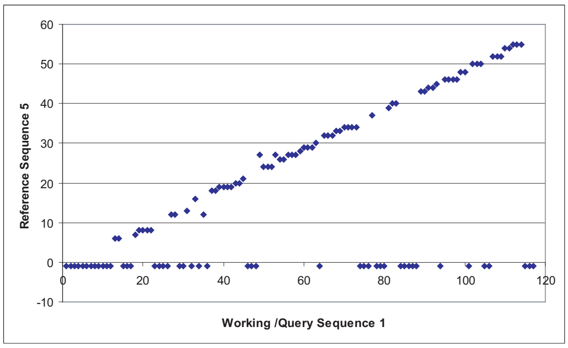
The benefits of employing the Hidden Markov Model are much more obvious when the place recognition is hard to perform as seen in Fig. 3.12. The fact that the Reference and Query Sequences were obtained under different lighting (light intensity and direction) conditions made Place recognition very difficult. In the figure at top, the place recognition results using only the Bernoulli Mixture Model are shown. A non-informative, Uniform probability over all the views of the Reference Sequence is maintained. The lower number of common features in the two sequence resulted in a high failure rate of 49%. At bottom, in Fig. 3.12c, the results of applying a 5-observation-long HMM, which resulted in a failure rate of 20% over the entire sequence, are shown. The figure in the middle shows the results of applying only the Bernoulli Mixture Model with an informative, non-uniform prior, which, itself was obtained from the procedure in Fig. 3.12c, with a success rate falling in between those obtained earlier.

3.5 The Reference Sequence as a Compact Representation of a Path.

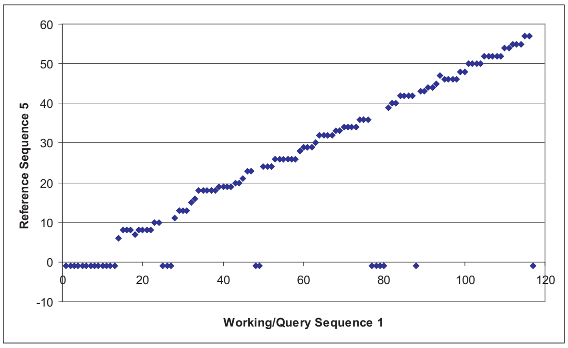
Any probabilistic representation of a robot environment must represent the joint probability either implicitly or explicitly, [Pearl 00]. As seen in chapter 1, while geometric maps aim to maintain a compound distribution of the events in the environment, topological maps break up the environment representation into 2 or more conditional distributions. With the increase in the size of the environment to be represented and with the addition of sensors and of sensor features, this joint probability (of being in a place and observing something) tends to become unmanageably large. The complexity of grid-based maps increases exponentially with the size of the map and special methods must be utilized to reduce the complexity of the Localisation process. The key



(a) Place recognition using a Bernoulli Mixture Model with a Uniform prior probability over all views of the Reference Sequence for matching each observation. A total of 57 failures were recorded from a total of 117 observations.



(b) Place recognition using a Bernoulli Mixture Model with an informative prior probability for matching each observation. A total of 46 failures were recorded from a total of 117 observations.



(c) Place recognition using a 5-observation-long Hidden Markov model. Each observation is matched using a Bernoulli Mixture Model with a prior given by the transition probability of the HMM. A total of 24 failures were recorded from a total of 117 observations.

Figure 3.12: Comparison of place recognition with and without recourse to HMMs. The results represent two sequences taken at different times of the day (morning and late evening), a fact that makes View matching rather more difficult. Only Camera #1 and Camera #2 were utilized in these sequences.

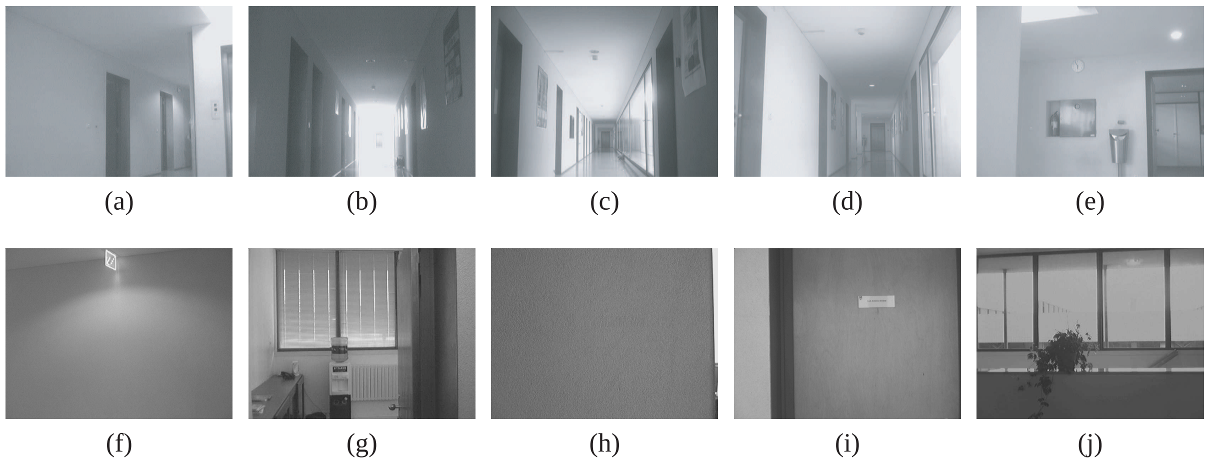


Figure 3.13: Some of the Images Sampled for the Experiments described in this work. The top row shows images captured from Camera #1 while the bottom row represents images taken by Camera #2.

Table 3.1: Comparison of Place-recognition successes with and without using HMMs.

Attempt	Reference Sequence	% success with No-HMM	% success with Pior Prob. and No-HMM	% success with HMM.
1	2	56	73	97
2	2	39	51	67
3	5	23	33	49
4	5	71	83	94

ability of the topological map to separate the conditional probabilities of the sensor data that, when multiplied with the marginal probability that represents the robot position, does not lead to such a large expansion in the map.

It is not relevant to debate whether the topological map is a high-level representation [Choset 01] or not. The topological map offers us the chance to break up the joint probabilities that express a map into two or more sets of conditional probabilities and express it as a graph. Various methods have been used to build Topological Maps. The nodes of the topological map usually represent convenient places where this conditional probabilities may be expressed. It matters less, how and where these nodes are located. Some mapping methods prefer to situate the nodes for ease of detection [Kuipers 91], others prefer to specify nodes more consistent with an approach that explores and segments the environment [Choset 01], [Thrun 98] or any other criteria.

The Reference Sequence exploits the fact that the robot must execute its motion along a path, in order to complete a mission, and represents the environment using only the conditional view distribution. The fact that the robot will arrive at the places in a sequential manner obviates the need to explicitly provide the transition probability matrix. For this reason we believe that a Reference Sequence and the accompanying sequence of instructions or behaviours, which make up the Robot Mission, is an efficient form of representing the desired motion along a topological path.

3.6 Summary

This chapter extended the method for matching a single current view for place recognition to a sequence of current views that includes the contextual information of each view in the Reference Sequence. The substitution of the single current view for a sequence of views was achieved by the adoption of a Hidden Markov model to align the Query sequence with the Reference Sequence.

The key modifications made to the method presented in the earlier chapter is that the views are now considered to be observations that are conditional hidden states (the actual position of the Robot in the Reference Sequence). The transition between these hidden states is modeled as a simple graph with the transition probabilities parameterised using a robot motion model, a model that generates transitions that are quite uncertain.

The Markov chain that is used to model the transitions between states in the Query sequence is a little different however with a set of 'insertion' states, $Lost\_Place$ that serve to accommodate those Places in the environment that have changes in appearance since the Reference Sequence, those places that were not sampled and situations in which the robot was plainly lost.

Chapter 4

Merging Topological Paths to create Maps

4.1 Introduction

The merging of smaller maps to create larger maps is a relevant topic in map building for robot navigation and there are various reasons we might be required to merge two or more smaller maps. One recent application of map merging has been in the integration of individual maps created by cooperating robots [Konolige 04]. We are particularly interested in map merging applications that use topological information to robustly merge different types of maps and a number of such works have been reported in recent literature. These works differ from each other in the nature of the sensory information that is included in the map and in the choice of the places at which merging is allowed to take place.

Like many localisation methods that use geometric maps, Dedeoglu et al. [Dedeoglu 99] track the position of the robot within a geometric map using a Kalman filter. As in the case of [Kortenkamp 94], pre-defined types of landmarks serve to make localisation more robust by allowing the tracking algorithm to periodically reset and annul the accumulated uncertainty in the odometry measurements. Often, topological information is obtained by abstracting out information from sensor data and the topological connectivity is used to merge the map, [Kuipers 00] being a well known example. Methods such as [Schmidt 06] selectively strip information from a geometric map to maintain only adjacency information representing a topological map. In another approach that uses only topological maps combined with some geometric information about the environment, [Huang 05] employ a method to merge two topological representations is presented.

The selection of candidate places at which merging of maps should take place is an important problem that must be solved before merging can take place. In the topological level of the Spatial Semantic Hierarchy [Kuipers 00] and again, in [Kortenkamp 94], a pre-defined set of indoor environmental characteristics such as junctions corners and doorways are utilised to create nodes in the map. The principal reason for using these so-called ’gateways’ is the reliability with which these type of places can be detected. The increased reliability that is associated with their detection makes these gateways suitable places at which maps can be merged with greater confidence.

The previous chapters of this thesis have developed the theoretical framework for localising the robot within a single path in the environment. Chapter 2 presented a method to perform place recognition by reducing the dimensionality of views containing a large number of different features. Later, in chapter 3, the use of the sequential context of the views in the Reference Sequence was found to have substantially improved the results of place recognition.

Let us suppose that the robot is led down two distinct paths within the environment and, at the end of Environment Familiarisation stage, it possesses two sequences of views. The robot can attempt to localise itself, simultaneously and independently, along each sequence using the techniques described in the earlier chapter. Besides being able to localize itself within each path, the robot must also be enabled with the capability to decide along which of the two paths it is currently travelling along, its ’global position’.

An extension of the concept of localisation along a single Reference Sequence is required to enable the robot to maintain its global position. Such a method will allow the robot to localise itself in a global map that is created by joining multiple, smaller topological paths.

We propose a method to create topological maps from multiple sequences of raw sensor data. In the current chapter, the complete environment within which the robot operates will be considered to be composed of multiple paths. A procedure that can maintain a global estimate of the robot position by successively locating the robot along multiple paths is presented. Using this method, the robot performs place recognition independently along each path, simultaneously evaluating the probability of it being along each of those paths.

Further, when the individual Reference Sequences overlap to some degree, the sequences can be ’stitched together’ along the overlapping stretch. All put together, these Reference Sequences make up the general topological map of the environment.

Two issues must be solved in order to arrive at this topological map:

- a procedure to maintain a consistent probability distribution across multiple paths must be developed and,

- a procedure to merge Reference Sequences, which includes a criteria for identifying overlapping segments of the individual paths, must be specified.

In the next section 4.2, a brief review of the procedure used to create the topological representations of each path is presented. In section 4.3, a procedure to simultaneously localise the robot within multiple sequences is detailed. This procedure builds on the technique developed earlier in this work and extends it to allow the robot to maintain the position of a robot along multiple topological paths. In section 4.4, an algorithm is presented to merge segments of data sequences into a generalised topological map. This algorithm allows the merging of these paths to create an unified topological map. In section 4.5, experiments that demonstrate the localisation along multiple paths and the procedure for merging topological paths are described together with the respective results. Some closing comments, in section 4.6, follow the presentation of results that were carried out on map-merging.

4.2 Place recognition along a Single Sequence of Views - a review

Before the presentation of the map-merging and localisation method, a brief review of the techniques that were detailed in the two previous chapters is presented. Section 4.2 summarizes the

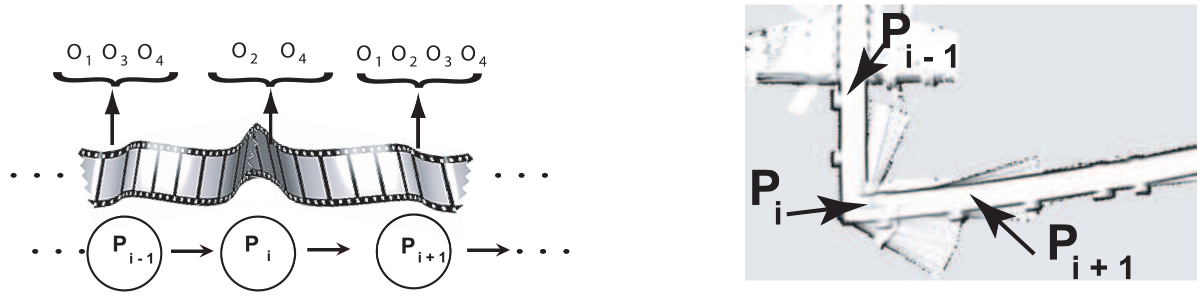


Figure 4.1: Sensory events in the topological are represented using a conditional distribution, where the experiment of viewing a particular sensory event is conditional on the particular state of the environment($O_1, \ldots, O_5$ are sensory events).

Figure 4.2: The robot is led through the environment on the Environment Familiarisation run, left. The 3D point clouds must be registered in the environment, right.

technique for localisation along a single topological path, as it was developed in chapters 2 and 3.

The algorithm used to integrate a large number of features and which is described in chapter 2 ensures that the place recognition should occur even when views are slightly altered. During the Environment Familiarisation phase, depicted at left in Fig. 4.2, the robot samples the environment according to a sampling plan, collecting a sequence of views from its various sensors resulting in the Reference Sequence. A repetition of the sequence of motion performed during the place recognition should propel the robot along the same Reference Sequence.

In chapter 3 it was shown how the sampled views which would normally be modelled as a left-to-right graph in Fig. 3.3, are augmented by the insertion of '$Lost\_Places$' as depicted in Fig. 3.5.

Any maneuver other than the ones taken during the Environment Familiarisation phase will take the robot to a place that was not sampled in the Environment Familiarisation phase. The $Lost\_Places$, in all a total of K in number, accommodate these possible views. Thus, each $Lost\_Place$ takes into account the fact that the robot might be seeing views that were not seen in the Environment Familiarisation phase.

The sequence begins with $P_{Lost\_Place\_0}$ which indicates that the robot is completely lost or has never localised. Also, before every original place P_i, there is a $P_{Lost\_Place\_i}$. By moving forward from one $Lost\_Place$, the robot can transition from $P_{Lost\_Place\_i}$ to any node P_k where $k > i$. Similarly, from P_i the robot can transition to $P_k : k > i$ or to $P_{Lost\_Place\_i+1}$. The graph does not allow a single-step transition from one $P_{Lost\_Place\_i}$ to another $P_{Lost\_Place\_j}$.

Subsequently, as the robot moves through the environment and need to localize itself, the current view is compared to the previously collected views and an inference is made of the current position of the robot. A Hidden Markov Model, HMM, is used to perform place recognition using the modified Markov Chain in Fig. 3.5 as a model for the transitions between the hidden states of the HMM. The Viterbi algorithm is commonly used in the context of HMMs to determine the most probable sequence of hidden states that gave rise to a particular sequence of observations. It is an inference tool that is associated with the process of making inferences in a HMM and is utilised to position the robot within the Reference Sequence by using the current sequence of observations.

The transition between the states is influenced by the transition probabilities between a pair of places in the graph shown in Fig. 3.5. A robot motion model is developed to evaluate the transition probability matrix. For each sequence of M observations, a simple distribution is used to model the transition probability distribution from each $Lost\_Place$ to the remaining original places in the Reference Sequence favouring places that lie closer in the Reference Sequence. The transition probability leading away from any of the original places in the Reference Sequence is uniformly split between the next original place (to the right) and to the corresponding $Lost\_Place$. The one-step transition probability from one $Lost\_Place$ to another $Lost\_Place$ is zero.

The first hidden state is always matched to the first $Lost\_Place$, $Lost\_Place\_0$. This $Lost\_Place\_0$, has a non-zero probability of reaching any place in the original Reference Sequence.

The observation model of the HMM is based on matching the current view with the views in the Reference Sequence. In the absence of any information regarding the view that will be visible at the '$Lost\_Place$', we arbitrarily define the observation probability as an Uniform distribution over the K views in the original Reference Sequence. The features from each view in the Reference Sequence are converted into binary form as described in chapter 2 and are represented within a Feature Incidence Matrix (FIM), $\mathcal{V}$. Due to the large dimensionality of the FIM, it is subsequently modelled as a Bernoulli Mixture Model (BMM). These parameters of the BMM are obtained by running the Expectation Maximisation(EM) algorithm.

The Mixture parameters and the posterior probabilities over the components, the z_i terms in (4.1), are used to evaluate the likelihood as depicted in (4.1), $P(V_k)$ representing the prior probabilities over each view k, in the Reference Sequence. As expressed in (4.2), the *Maximum Likelihood Estimation* is used to obtain the index k^* in $\mathcal{V}$ that best describes the object to be matched, V_{obs}.

$$P(k|V_{obs}) = \frac{\sum_{c=1}^{C} P(V_k) z_{kc} \alpha_c P(V_{obs}|\Theta_c)}{\sum_{k=1}^{K} \sum_{c=1}^{C} P(V_k) z_{kc} \alpha_c P(V_{obs}|\Theta_c)} \tag{4.1}$$

$$P(k^*|V_{obs}) = \operatorname*{argmax}_{k}^{K} P(k|V_{obs}) \tag{4.2}$$

So far we have dealt with only one Reference Sequence. When we are dealing with multiple Reference Sequences it is necessary to refer to $P(k_s|V_{obs})$ as $P(k_s|V_{obs})$, i.e. the probability of V_{obs} being matched against view k in the Reference Sequence s.

4.3 Simultaneously localisation in Multiple Sequences of Views

In the previous section, a consistent way of maintaining the localisation probability over a single path was reviewed. One of the key characteristics of that approach was the use of the Viterbi algorithm to exploit the information that is available in the sequential context in which the views are arranged within the Reference Sequence. The Viterbi algorithm allows the evaluation of

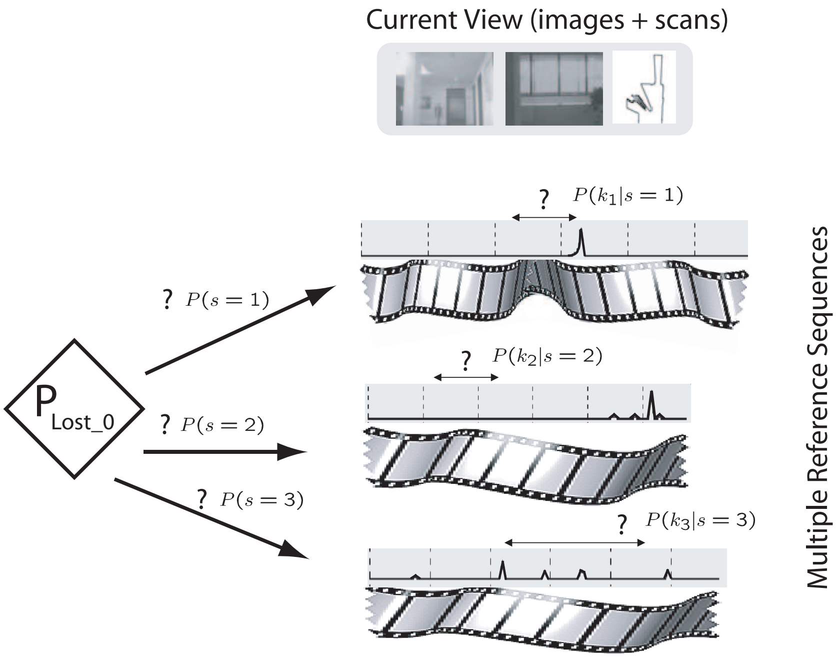


Figure 4.3: A schematic for the comparison with multiple trajectories (sections of different Reference Sequences or Topological Paths).

the transitions and observations within a single Reference Sequence that most likely led to the current observation sequence. However, the Viterbi algorithm does not allow a comparison of the likelihood of sequences for different paths, i.e. for two different HMMs.

We propose a more consistent way of handling localisation along multiple sequences that is based on the quality of the match of the sequence of observations for each Reference Sequence. Localisation of a mobile robot using such an approach, depicted graphically in Fig. 4.3, will reflect the fact that each data sequence is obtained by separately sampling different paths of the environment.

If more than one Reference Sequence is available for the robot to localise itself against, this method will maintain the global position of the robot by modelling its global position as a combination of two, independent probability distributions. The first distribution is the marginal distribution, of the robot being along a particular Reference Sequence, s. The second distribution, independent of the first, maintains the position of the robot within that particular Reference Sequence. The application of this global positioning scheme is depicted in Fig. 4.3 for three independent Reference Sequences and is expressed in (4.3). The proposal of this method is influenced by the corollary proposed by Montemerlo in [Montemerlo 02] which says that, given that the path is known, the sensor observations are conditionally independent on the robot path.

$$P(k_s, s|V_{obs}) = \frac{P(s) * P(k_s|V_{obs}, s)}{\text{constant}} \tag{4.3}$$

The second term, $P(k_s|V_{obs})$, in the numerator on the right-hand-side of (4.3), is the probability with which the current observation matches the views in the Reference Sequence. This is the distribution (4.1), that was first presented in chapter 3 and was summarised in section 4.2 in the current chapter.

The first term, $P(s)$, in the numerator on the right side of (4.3), denotes the probability of the robot being along each of the Reference Sequences. We have developed a measure that reflects the marginal probability distribution of the robot being within each Reference Sequences. This measure is based on the *quality* of the output of the place recognition procedure. The correct path along which the robot is moving should have the lowest uncertainty given the current observations.

In the next two sub-sections, the expression for the term $P(s)$ is developed in terms of the uncertainty of the localisation estimates within each Reference Sequence. In section 4.3.1, the expression is first developed for the case of a single, isolated, observation. The equivalent expression is then extended, in section 4.3.2, for the case where a sequence of observations as used within the Hidden Markov Model, HMM.

4.3.1 Calculating $P(s)$ when using a Single Observation

We have used the uncertainty of the place recognition results along a particular Reference Sequence as a proxy for the probability of the robot travelling along that same sequence. The binary entropy of the probability distribution of a discrete random variable is a commonly used measure of the uncertainty of that distribution. This measure, reproduced for a variable X in (4.4), can be evaluated for any probability distribution such as for the estimate for a single view localisation, $P(k_s|V_{obs}, s)$. We have defined the probability of the robot travelling along a particular Reference

Sequence according to (4.5).

$$H(X) = -\sum P(x)\log(P(x)) \tag{4.4}$$

$$P(s) = \frac{1}{\text{constant}}\Big[1 - \frac{1}{\log(K_s)} \times \sum_{k_s=1}^{K_s} P(k_s|V_{obs}, s)\log(P(k_s|V_{obs}, s))\Big] \tag{4.5}$$

$$\text{constant} = \frac{1}{\sum_{s=1}^{S} P(s)} \tag{4.6}$$

Expression (4.5) is based on the 'normalised entropy' of the distribution $P(k_s|V_{obs}, s)$. This expression is a measure of how uncertain a place recognition algorithm is within the particular Reference Sequence, s. The *less uncertain* a probabilistic distribution is over a Reference Sequence, the higher the probability that the robot finds itself within that particular Reference Sequence. The normalising term $\frac{1}{\log(K_s)}$ allows the comparison of trajectories of different lengths, where K_s denotes the number of places in the Reference Sequence, s.

With the arrival of new sensor data, the probability of the robot being along the *correct* Reference Sequence should begin to improve to the detriment of the other Reference Sequences. If the current observation does not match, sufficiently well, with one of the views in the Reference Sequence, the uncertainty of the localisation results will be high and the corresponding probability of the robot being along that Reference Sequence will be low.

4.3.2 Calculating $P(s)$ when using an observation sequence

The expression for the case of a single observation, developed in the previous sub-section is quite straight-forward. In the case of a sequence of observations, the expression must be developed more carefully since the uncertainty refers to the results of the Viterbi Algorithm.

When a particular sequence of observations is utilised to obtain the most likely sequence of places that were occupied by the robot, the uncertainty of the possible sequence of places must be evaluated.

The sequence of places that the robot occupied within a Reference Sequence is called the robot trajectory. The trajectory is defined as the section of a topological path that the robot has travelled along, with the section beginning at an initial state i, ending at a final state j and having no intervening states equal to j. According to the above definition, the trajectory does not include loops through the environment.

As reviewed in section 4.2, for each Reference Sequence, the motion of the robot is modelled using the Hidden Markov Model (HMM) and the Viterbi algorithm provides the most likely sequence or trajectory , $\hat{\text{traj}}_s$, for the Reference Sequence s. The algorithm works by calculating all the possible sequences of places that explain the sequence of observations. The expression for the uncertainty of the inferred trajectory that is equivalent to (4.5), is less straight forward to compute since it requires the evaluation of the entropy of the trajectory, $H(\hat{\text{traj}}_s|V_{obs})$.

The brute-force evaluation of this quantity would require the calculation of all the possible trajectories in the Reference Sequence that explain the sequence of observations. The value of $H(\hat{\text{traj}}_s|V_{obs})$ would be obtained by the application of the probability distribution of these sequences of states that explain these observations in expression (4.4). An efficient method to calculate this quantity was presented in [Hernando 05]. This method extends the Viterbi algorithm to calculate the entropy of the sequences of hidden states as each observation arrives. The procedure described in [Hernando 05] is adopted for the calculation of the uncertainty of the trajectory. The calculation of $H(\hat{\text{traj}}_s|V_{obs})$ requires some modification of the data structures and some additional computations within the Viterbi algorithm.

The final expression for $P(s)$, for the case involving a sequence of observations, is shown in (4.7) where $H(\hat{\text{traj}}_s|V_{obs})$ is calculated using the method outlined in [Hernando 05]. The constant term in (4.8) is used to create an expression for a regular probability distribution and the normalising term, $\binom{K_s}{\#\text{traj}}$, allows the $P(s)$ to be calculated independently of lengths of the individual paths, as explained below.

$$P(s) = \frac{1}{\text{constant}}\left[1 - \frac{1}{\log(\binom{K_s}{\#\text{traj}})} \times H(\hat{\text{traj}}_s|V_{obs}\right]. \tag{4.7}$$

$$\text{constant} = \frac{1}{\sum_{s=1}^{S} P(s)} \tag{4.8}$$

The normalising term, $\frac{1}{\log(\binom{K_s}{\#\text{traj}})}$, adjusts the entropy for the number of combinations of states that are possible in each Reference Sequence for an observation length of $\#$traj. The expression $\binom{K_s}{\#\text{traj}}$ denotes the binomial operator that is used to account for the different combinations of state sequences of length $\#$traj that are possibles in a Reference Sequence with a total number of K_s original views.

4.4 Merging Topological Paths

The previous section has presented a method to globally localise the robot within a set of paths. The representation of each of the paths is obtained by leading the robot through a particular section of the environment. If the robot is successively led through the environment, each time along a different path, the topological representations along multiple paths that together make up the topological map of the environment.

The advantages of merging multiple Reference Sequences are two-fold:

- **Localisation can be performed on independent Reference Sequences:** The localisation method developed in the previous section assumes independence between the individual Reference Sequences. This independence is achieved by identifying the overlap between the Reference Sequences created during the Environment Familiarisation phase. This step helps guarantee the corollary proposed by Montemerlo in [Montemerlo 02].

- **Possibility of localisation boot-strapping:** The map merging procedure is also required to enable the localisation method to move from one Reference Sequence to another. The merging procedure makes pair-wise comparisons between segments of different Reference Sequences leading to the discovery of overlapping segments. These overlapping segments allow the identification of possible transitions from one Reference Sequence to another, via the overlapping segments.

This problem of maintaining the consistency of the merged global map shares some of the challenges faced by other map-building methods described in the literature. The Simultaneous

Localisation and Mapping, SLAM, algorithms and their variants maintain the relation between features through the use of the position and error correlation matrices [Thrun 05]. SLAM is well suited to the process of incrementally mapping an environment. An extension of the map is performed by adding the position of each new feature and by updating the error matrices that represent the uncertainty of existing features. In most map building procedures the merger of two maps or of two, previously separate, sections is a difficult problem. This problem is often referred to as the loop-closure problem. In the case of SLAM, loop-closure involves the conciliation of the position and error matrices for the two separate maps/sections whilst simultaneously associating features from one of the maps/sections with the other.

The problem is a difficult one to solve since the position is not accurately known and, often, other means such as image features in the case of [Newman 06], must be used to select those places where merging will occur and other places where only position updates will be performed. Once the position at which the maps must be merged is established, methods such as [Carpin 07] can be used to locally search for and optimise the transformation that best ’fits’ one map into another.

Where the environment is sufficiently rich in sensorial detail, assumptions are made about the ’slow’ variability of the environment. The similarity between places can be usually established by comparing changes in the sensor data over a short path. In [Schmidt 06], Schmidt et al. use the ’width’ or smaller dimension of the environment, as measured using ultrasound sensors as the robot moves, to verify the similarity of places in the environment. A similarity term is defined for the width of passable sections in the environment and changes in this width are used to split regions in the environment.

In [Howard 06], Howard et. al. employ the projection of a higher dimensional structure, a manifold, onto a plane to help resolve the problem of place aliasing and scene variability. Importantly, representing the sensor measurements on the manifold rather than on the planar map allows merging of places that are previously visited by the robot to be put off until more information is available. The authors re-cast the problem of place-aliasing as a non-symmetric projection problem where each position in the environment (in the planar projection) can be represented as a different place on the manifold.

Hierarchical representations of the environment have been used in the literature since the introduction of the Spatial Semantic Hierarchy, SSH, by Kuipers [Kuipers 77]. These typically seek to identify previously visited or already-mapped places with help of the topological representation. For example, in [Kuipers 02], an extension of the SSH architecture, Kuipers and Beeson seek to merge topological information by testing the various merging hypothesis. The environment is sensed using very imprecise sensors and is subsequently abstracted in terms of corners and junctions. The hypothesis for merging are evaluated by moving the robot around the environment such that loop closure hypothesis can be verified.

In [Stewart 03] a geometric map is broken up into a number of smaller, ’more unique’, environment layouts such as junctions and door openings and then strung back together in the form of a Dirichlet process. The novel method dwells on the problem of estimating the likelihood of previously unseen environment layouts and builds the prior probabilities of the different possible environment layouts.

Although not related to mobile robotic applications, in [Scarlatos 93], Scarlatos tackles some of the issues that are faced by map building algorithms regarding the efficient representation of large environments. Although the emphasis is on the development of appropriate data structures, chapter 6 of the same publication, addresses the problem of merging spatial representations including, for the case in which the representations are not of the same type. Although the use of line, point and area-based methods for merging the accurate maps used in that work is of interest, the maps that are available in mobile robotics are comparatively far from accurate given the noisy sensory readings and the accumulation of mapping errors.

In the context of our work, at a coarser or ’higher’ level, the environment is viewed as being composed of multiple paths. Each path in turn is made up of a sequence of views. This representation is depicted in Fig. 4.4, by using the topological map described in [Thrun 98]. The proposed method of building a global topological map from a merger of paths can be contrasted with the method described in the article [Thrun 98], where the ’global’ topological map is constructed after building the complete geometric map, by abstracting out certain properties of the ’global’ geometric map.

The merging of Reference Sequences into one global topological map comprising multiple

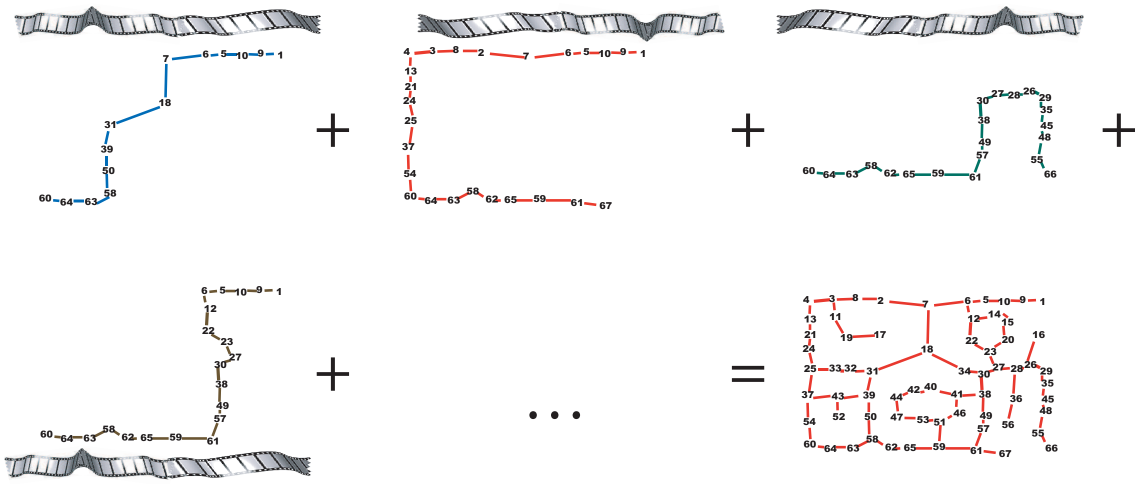


Figure 4.4: The global topological map can be viewed as a collection of multiple, non-overlapping paths through the environment. This process of creating a topological map by merging the topological representation along multiple paths can be contrasted with methods such as [Thrun 98], where the 'global' topological map is constructed by abstracting out the 'global' geometric map.

topological paths is performed using a two-stage procedure.

1. Identification of Merge Candidates - Measuring View Similarity

2. Path Merging - Verifying Reference Sequence Overlap

In the first step, *Identification of Merge Candidates*, each view in a Reference Sequence is separately compared with every view in another Reference Sequence to generate hypothesis where possible merging could be performed. This step results in the creation of a view-similarity matrix that indicates how similar each view in a Reference Sequence is with every view in the Reference Sequence it is being compared with.

In step 2, *Path Merging*, segments of a Reference Sequence around a pair of candidate views that are identified in the earlier stage are aligned to verify whether an overlap actually exists. It is important to point out that the first step is essential since it identifies places in the pair of Reference Sequences that are really similar. If the step 2 were to be performed on every pair of sequences, without first testing for similarity, a forced alignment of views that are actually not similar might occur.

4.4.1 Identification of Merge Candidates - Measuring View Similarity

A method that merges paths to create a global topological map of the environment must identify possible places at which a pair of sequences *'cross over'* each other. A view-similarity algorithm can identify possible individual instances of such overlaps. The definition of similarity varies greatly in the literature and this definition is often a function of the type of environment and the nature of the sensors being used.

In [Ishiguro 96] the Fourier components that make up Omni-directional images are compared for an evaluation of similarity. Ho and Newman [Ho 05] and Posner, Schroeter and Newman [Posner 06] use place similarity measures to improve the robustness of loop-closure and map merging algorithms that normally employ geometric information. In the latter publication, the similarity between two sequences of views has been depicted using a view similarity matrix, as seen in Fig. 4.5. It is pertinent to note that although Posner et. al [Posner 06] have noted that algorithms that make use of the [sic] 'temporal ordering of data', are still not very common, their own method still does not make explicit use of sequential context of the views.

A recent article by Zivkovic et. al [Zivkovic 07] also attempts to use information obtained from near-by scenes to define places in the environment. As in the case of [Posner 06], a clustering approach is used to group images and represent places in the environment by using a typical set of images for that place. In [Thomas 00], Thomas and Donikian hypothesize a hierarchical set of [topological] representations that represent the environment using similarity of places. The developers of these methods claim that such labeling of (similar-looking) places is in line with the spatial concepts that humans employ.

In our approach, each view in a Reference Sequence is compared with each view in every other Reference Sequence. The posterior probability distribution that is obtained by comparing each view in Reference Sequence s_i with another Reference Sequence s_j, for all the views in s_i results in the 'view similarity' matrix. This similarity matrix indicates how similar a particular view is to each view inside the other Reference Sequence. The similarity between a pair of views is calculated using a metric based on the number features that the views have in common after having reduced the dimensionality of the feature space using the Bernoulli Mixture Model.

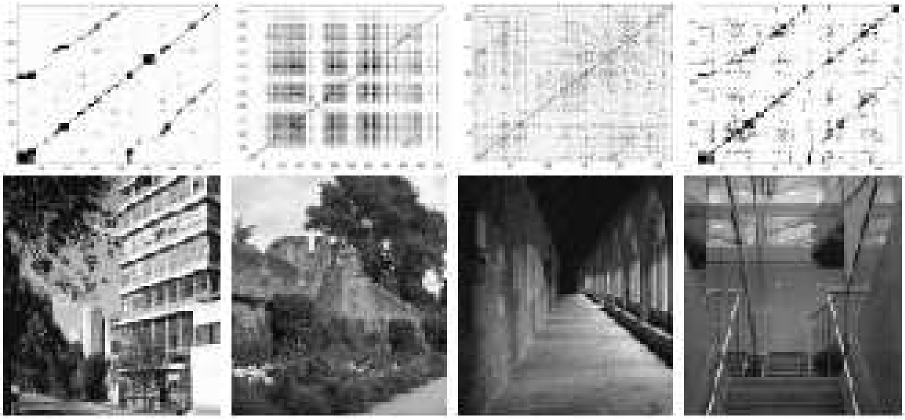
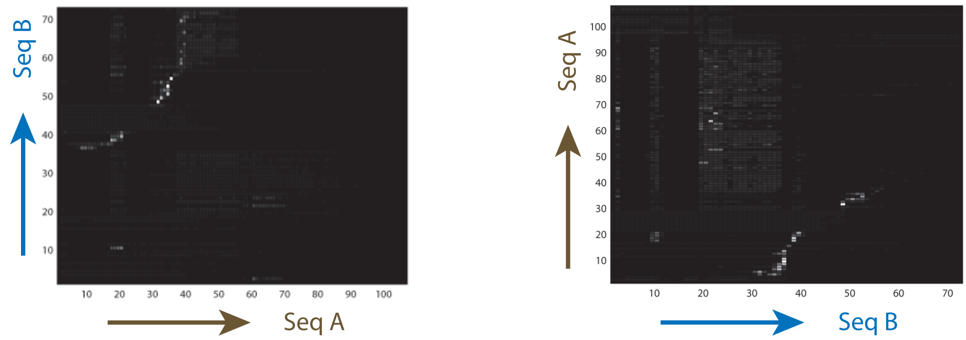


Figure 4.5: The calculation of View similarity to improve geometric map merging has been utilised in [Posner 06]. The view similarity information is used to add robustness to map-merging algorithms that normally use geometric or spatial information. Black positions in the matrix indicate greater similarity while white indicate lower similarity.

An example of such a comparison is seen in Fig. 4.6 for two sequences taken at the Department of Electronics and Computer Engineering, DEEC at the University of Coimbra. The figure at left in Fig. 4.6a depicts the view similarity matrix when a Reference Sequence ’B’ is compared to another Reference Sequence ’A’, and in Fig. 4.6b, Reference Sequence ’A’ is compared to Reference Sequence ’B’. As is visible in Fig. 4.6, and as it might be expected, the two matrices are not completely similar given that the features used to evaluate the similarity are different. In Fig. 4.6a, the features found in the Reference Sequence A are used to calculate the view similarity between the Reference Sequences whereas in Fig. 4.6b its the features from Reference Sequence B that are used. But the similar views in one Reference Sequence are consistently evaluated as being similar to views in the other Reference Sequence. These high similarity places are candidates for possible overlaps between parts of the sequences as described in section 4.4.2.

4.4.2 Path Merging - Verifying Reference Sequence Overlap

The aim of this step is to verify that overlapping between the sequences does take place and to split the environment into distinct regions. The overlapping regions, common to both Reference Sequences, are removed from one of the Reference Sequences and maintained only within a



(a) Reference Sequence B compared to A. (b) Reference Sequence A compared to B.

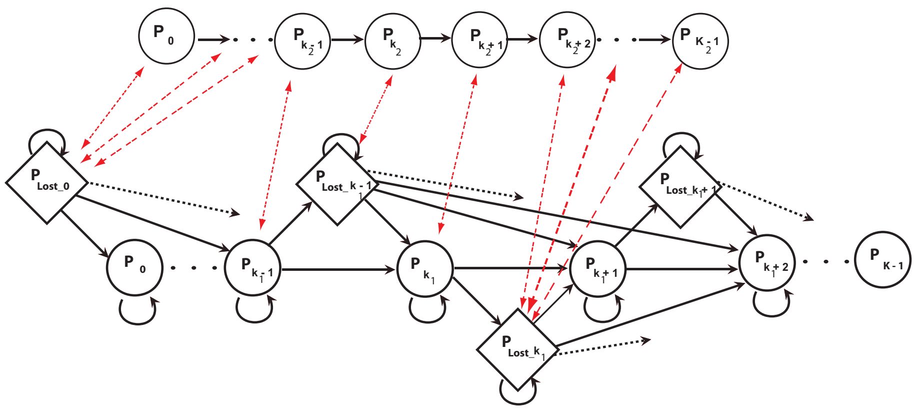
Figure 4.6: View similarity matrices are shown for experiments described later in section 4.5. The similarity (lighter indicates greater similarity) between views is calculated using a metric based on the number features that the views have in common after having reduced the dimensionality of the feature space using the Bernoulli Mixture Model. In this figure white values in the matrix indicate greater similarity and black indicate no similarity.

single Reference Sequence. Thus, no two reference Sequences maintain the same part of the environment.

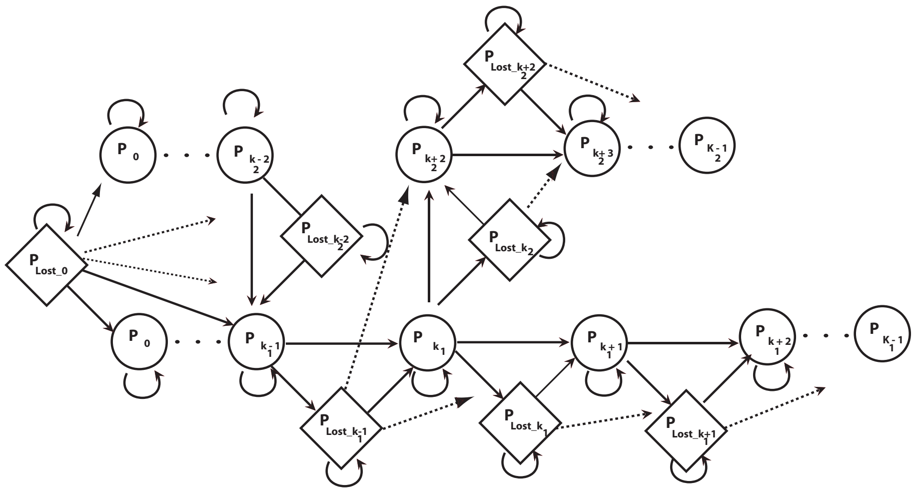
The evaluation of place-similarity must be robust to noise, to some changes in the view-point and to some changes in the dynamic environments. This procedure can handle for slight differences in the viewpoints of the robot as it travelled over the overlapping region in the Reference Sequences.

A simple algorithm that tests the hypothesis for merging segments of the new data sequence into the topological map is presented in Algorithm 2. The algorithm entails an alignment of segments of the Reference Sequences in the neighbourhood of the nodes which were deemed to be similar from the view-similarity matrix calculated in the previous step. If the similar nodes are relatively close to each other, suggesting that the non-similar intermediate nodes results from sensor noise, that part of the paths are considered to be overlapping.

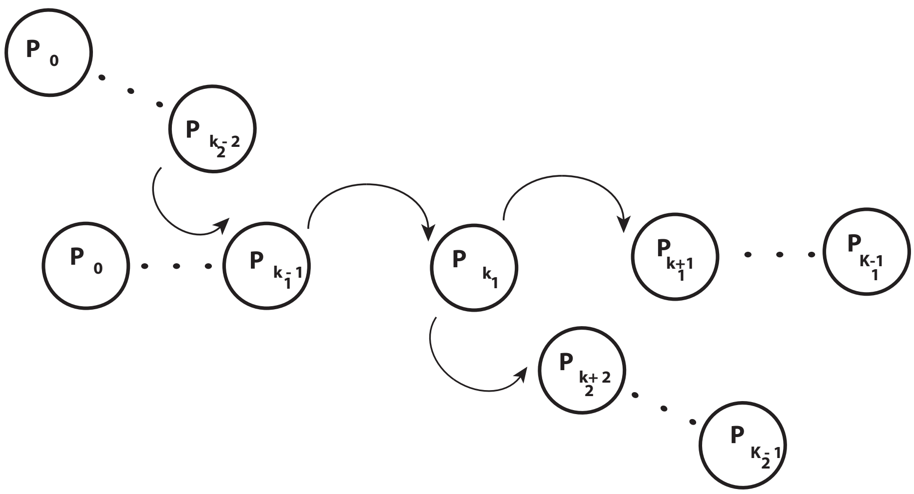
As seen in chapter 3, each place in the Reference Sequence is limited by a pair of '$Lost\_Places$'. The role of the $Lost\_Places$ is extended to incorporate possible overlap with other Reference Sequences. Merging of a new data sequence into a Reference Sequence is achieved by modifying the Markov chain shown in Fig. 3.5 to expand the role of the $Lost\_Places$ in the graph. These $Lost\_Places$ are now allowed to 'absorb' segments of the new sequence that must be merged.



(a) Finding overlap in the data sequence with an existing topological path.



(b) The final topological map.



(c) The places in the final topological map.

Figure 4.7: Merging of Segments of a data Sequence with the topological representation of another Path.

Algorithm 2 Merging Hypothesis For Topological Paths

$H_{new} = NULL$ // the new sequence is completely separated
$^{min}t_{new}$ // the minimum sequence length for declaring overlap
Require: $V_{new} \geq^{min} t_b$ // the new sequence has min length
$T_{new} = V_{new}$ // all new views are potential match
while $(T_{new} > 0)$ **do**
  $i \leftarrow V_{new}^1$
  $j \leftarrow V_{new}^{1+^{min}t_{new}}$
  if TEST$(t_{i,j})$ **then**
    // test for no overlap
    $H \leftarrow t_{i,j}^b$ //no overlap, current trajectory in B is added to Hypothesis
    remove $V_{new}^1, \ldots V_{new}^{1+^{min}t_{new}}$ from T^b //current trajectory removed from further tests
  else
    // overlap confirmed
    remove $V_{new}^1, \ldots V_{new}^{1+^{min}t_{new}}$ from T^b //current trajectory removed from further tests
  end if
end while
return H // the non-overlapped sequence is returned

Without loss of generality, Fig. 4.7 shows the effect of matching a segment of a Reference Sequence 1, that partially overlaps with another Reference Sequence 2. In this case, two of the views in Reference Sequence 1 were matched with part of the Reference Sequence 2 and an intermediate view was matched to an intervening $Lost\_Place$.

Those views occurring before the successful match are absorbed within the first or global $Lost\_Place$ and those views occurring after the successful match are absorbed (matched against) in the $Lost\_Place$ occurring after the match.

4.5 Experiments

In the above sections a procedure to maintain the global position of the robot in the map that is made up of multiple paths was presented. A procedure that can be used to merge overlapping sections of multiple paths to create a topological map was also described.

The algorithms and procedures are still in a relatively early stage of development and our localisation application does not use all the information that is available from the merging of the topological paths, namely the improved transition probabilities that are obtained from merging. Ongoing work seeks to include this information within the HMM algorithms that perform place recognition along each path. Additionally, the calculation of the uncertainty of the HMM trajectory inferred by the Viterbi algorithm was implemented and tested only on small Reference Sequences. For the experiments reported in this section, it was not possible to run the modified Viterbi Algorithm, as a result of which, $P(s)$, was calculated using the algorithm outlined in section 4.3.1.

A first experiment with two Reference Sequences was carried out to demonstrate the global localisation procedure. This experiment served to underline the need to identify and subsequently remove the overlap between Reference Sequences.

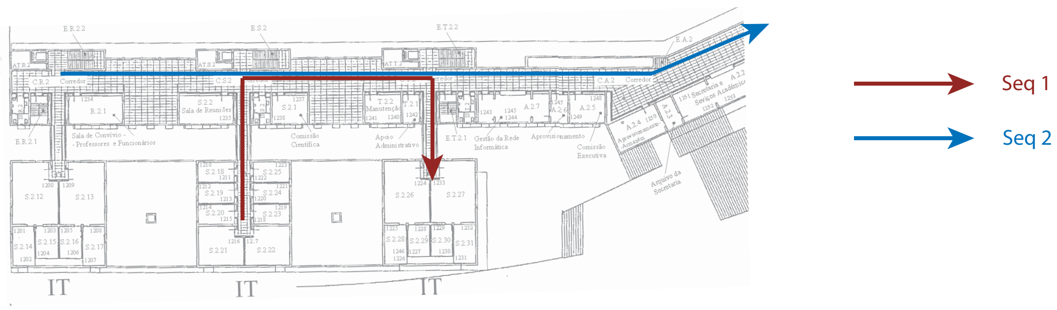
Two Reference Sequences were obtained by leading the robot along a corridor environment as shown in plan view in Fig. 4.8. The two paths having an overlap somewhere in the middle of the corridors where the robot briefly travelled along the same path. Each sequence consisted of 640x480 gray-scale images taken by two cameras mounted on a mobile robot. One of the cameras is facing forward and the other is facing away from one of the sides of the robot.

The two-step merging procedure described in section 4.4 was run and the overlap in the two topological paths was identified according to the algorithm presented in Algorithm 2. The second Reference Sequence was merged with the first Reference Sequence over a single overlapping segment, between views 16 and 32 in sequence $\#1$ and views 22 and 46 in sequence $\#2$, see Fig. 4.9a.

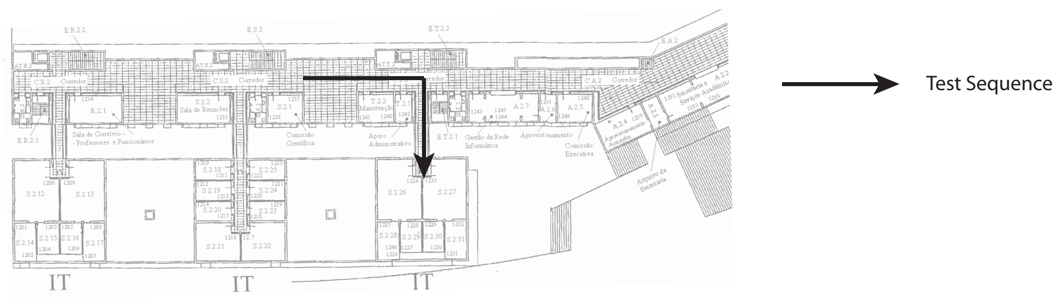
During a subsequent run, the robot is positioned at a place that lies on the overlapping segment of the paths and is driven along the path corresponding to the $sequence\#1$. The overlapping segment of the two Reference Sequences was not removed and are maintained within both Reference Sequences.

The robot localises itself in the global topological map and the probability of being on each of the original sequences is plotted and shown in Fig. 4.9b, it is impossible to tell whether the robot is localised in both Reference Sequences or in no sequence at all. To enable the use of the HMM models within a generalised topological map, overlapping segments should be removed.

To demonstrate the effect of the removal of overlapping segments, a second experiment was

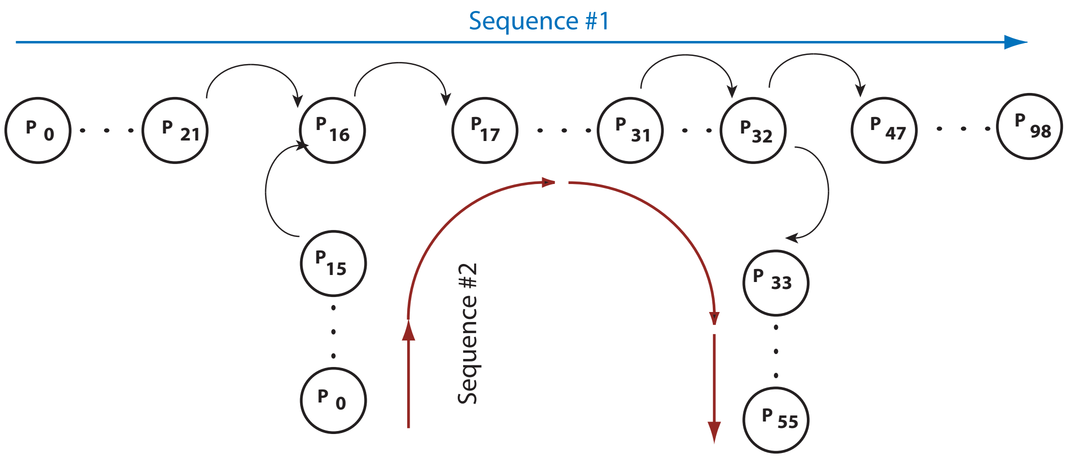


(a) Layout of the 2 Reference Sequences that were recorded on the second floor of the DEEC Building.

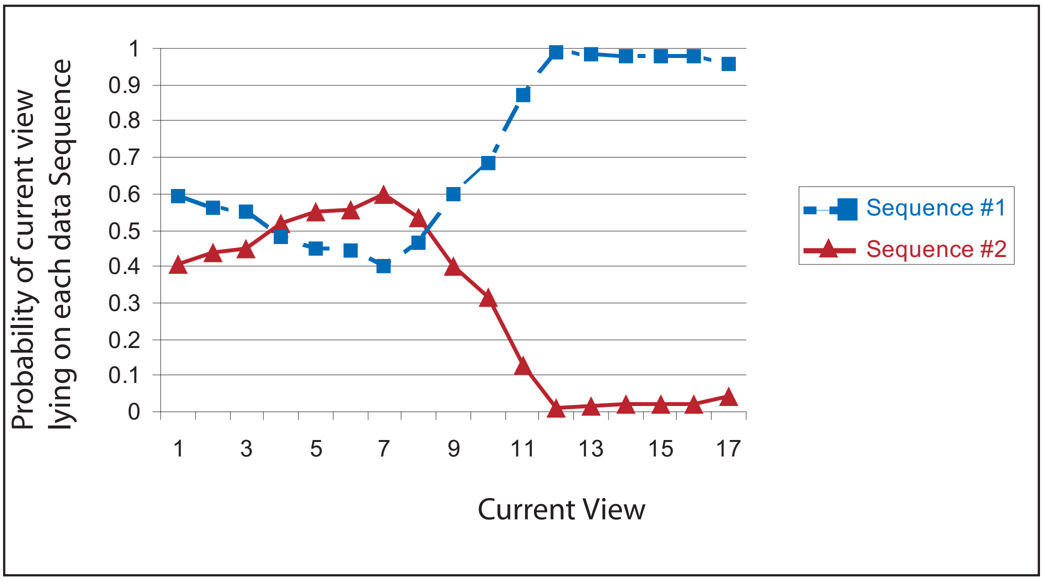


(b) Layout of the test path on the second floor of the DEEC building over which localisation was performed.

Figure 4.8: Two sequences with overlapping segments were gathered from the environment, as shown at top. Subsequently, the robot was driven over part of the region covered by the two Reference Sequences, as shown at bottom.



(a) Sequence #2 consisting of 98 views is merged to Sequence #1 consisting of 55 views. The views or places in the common segment have been numbered according to Sequence #1.



(b) The figure plots the probability of the robot being on each of the original Reference Sequences.

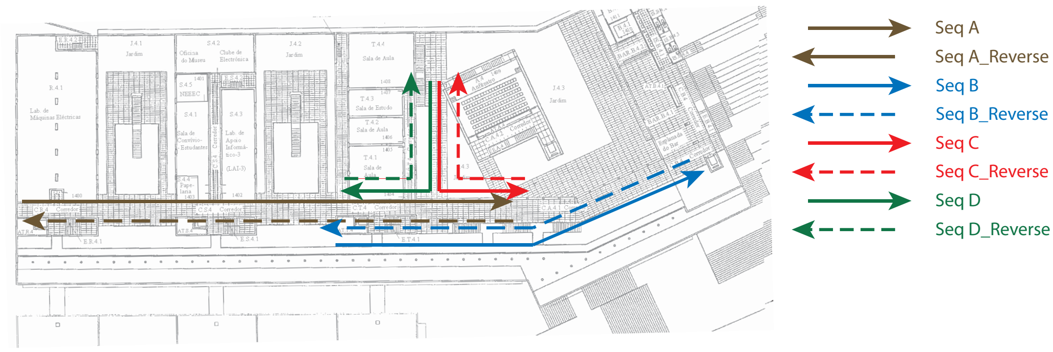
Figure 4.9: Merged Topological graph and probability distribution. While the robot is on the overlapping region, either sequence is probable. Upon reaching the place where the paths separate, the probability of the robot being on the incorrect path rapidly reduces.

carried out on a longer stretch of environment consisting of 8 Reference Sequences, collected by driving the robot forwards and backwards along 4 paths in the environment. The 4 paths cover a distance of approximately 300 meters and overlap to various extents. The plan view of the environment is seen in Fig. 4.10a. Sample images from six of the eight Reference Sequences are shown in Fig. 4.11 (Camera #1) and Fig. 4.12 (Camera #2). The 8 paths are named thus:

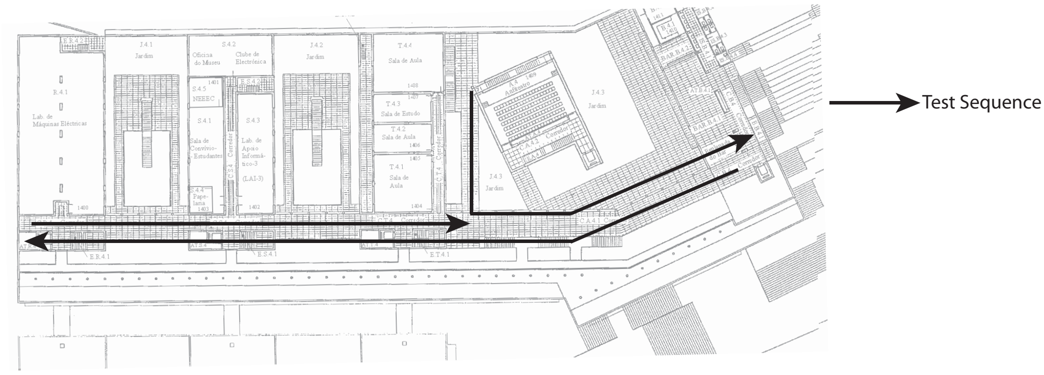
- Reference Sequence A Forward: moving forward along path A.
- Reference Sequence A Reverse: Retracing in reverse along path A
- Reference Sequence B Forward: moving forward along path B.
- Reference Sequence B Reverse: Retracing in reverse along path B.
- Reference Sequence C Forward: moving forward along path C.
- Reference Sequence C Reverse: Retracing in reverse along path C.
- Reference Sequence D Forward: moving forward along path D.
- Reference Sequence D Reverse: Retracing in reverse along path D.

The merging of the 8 Reference Sequences is performed as outlined in section 4.4. The view similarity matrices for some of the sequences that overlap with each other are seen in Fig. 4.13. As can be seen in that figure, the view similarity matrices for some pairs of Sequences indicate an overlap quite clearly. This is true especially in the case of Reference Sequences A and B, Reference Sequences A Reverse and B Reverse, Reference Sequence C Reverse and D Reverse, Reference Sequences B and D Reverse. The view similarity matrices for the other pairs of Reference Sequences do not appear to indicate overlap despite the robot having physically covered the same path. This result indicates that it might not always be possible to identify overlapping segments and to merge the respective Reference Sequences.

The procedures outlined in the previous sections is followed to merge the Reference Sequences. The candidates for map-merging in the view similarity are then verified for actual overlap according to the procedure outlined in section 4.4.2 and the overlapping segments are



(a) Layout of the 6 Reference Sequences that were recorded on the fourth floor of the DEEC Building.



(b) Layout of the test path on the fourth floor of the DEEC building over which localisation was performed.

Figure 4.10: Eight sequences with overlapping segments were gathered from the environment, as shown at top. Subsequently, the robot was driven over part of the region covered by the 8 Reference Sequences, as shown at bottom.

removed from the *longer* Reference Sequence. This is a simple criteria for the removal of overlapping sequences and was adopted on account of its simplicity.

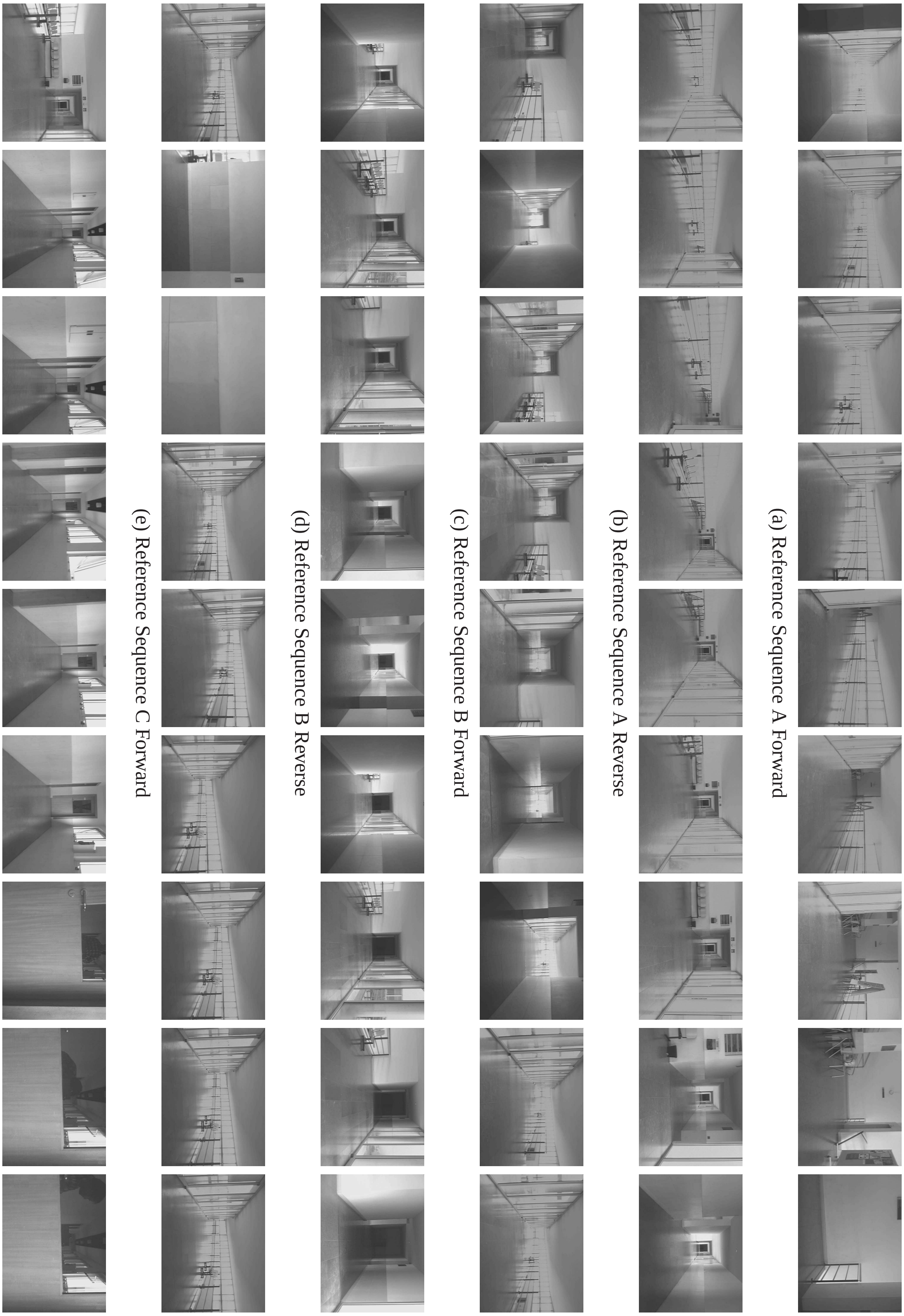
Subsequently, the robot is driven along part of the environment covered by the 8 Reference Sequences. The path covered by the robot is shown in Fig. 4.10b. Localisation was performed simultaneously along all 8 Reference Sequences with the overlapping segments removed. The global position of the robot is obtained by combining the probability $P(s)$, of the Robot being along a Reference Sequence s and the probability distribution $P(k_s|V_{obs}, s)$ over the views in that Reference Sequence. The probability distribution $P(s)$ is plotted in Fig. 4.14 over the entire test path. The robot is positioned on the Reference Sequence with the greatest value of $P(s)$.

It can be seen in Fig. 4.14, that when the robot is travelling along the part of the environment covered by Reference Sequences A Reverse, B Reverse, C Reverse, the values of $P(s)$ are quite stable and a single Reference Sequence is consistently selected, leading to correct global localisation of the robot over the topological map.

At other times, as can be seen in Fig. 4.14, there is a large variation in the probability distribution $P(s)$. This occurs partly because, even when there was no overlap, some of the paths are quite similar to each other, for example in Reference Sequence A and B both cameras were facing regions of the environment that were very similar and partly because at other places, very few features could be extracted, for example in Reference Sequence C and D the sidewards facing cameras were looking at a texture-less wall.

Such observations lead us to believe that the merging of Topological maps will be dependable only at those places where the views in the Reference Sequences are quite distinct, i.e. wherever localisation is *less uncertain*. This is a different problem in its own right and has not been addressed in this document.

We can also see how the removal of the overlapping segments allows the identification of those places where the localisation is successful and others where it fails. The situations which result in similar values of the maximum $P(s)$ for multiple Reference Sequences are an indication that the robot cannot localise itself.



(a) Reference Sequence A Forward

(b) Reference Sequence A Reverse

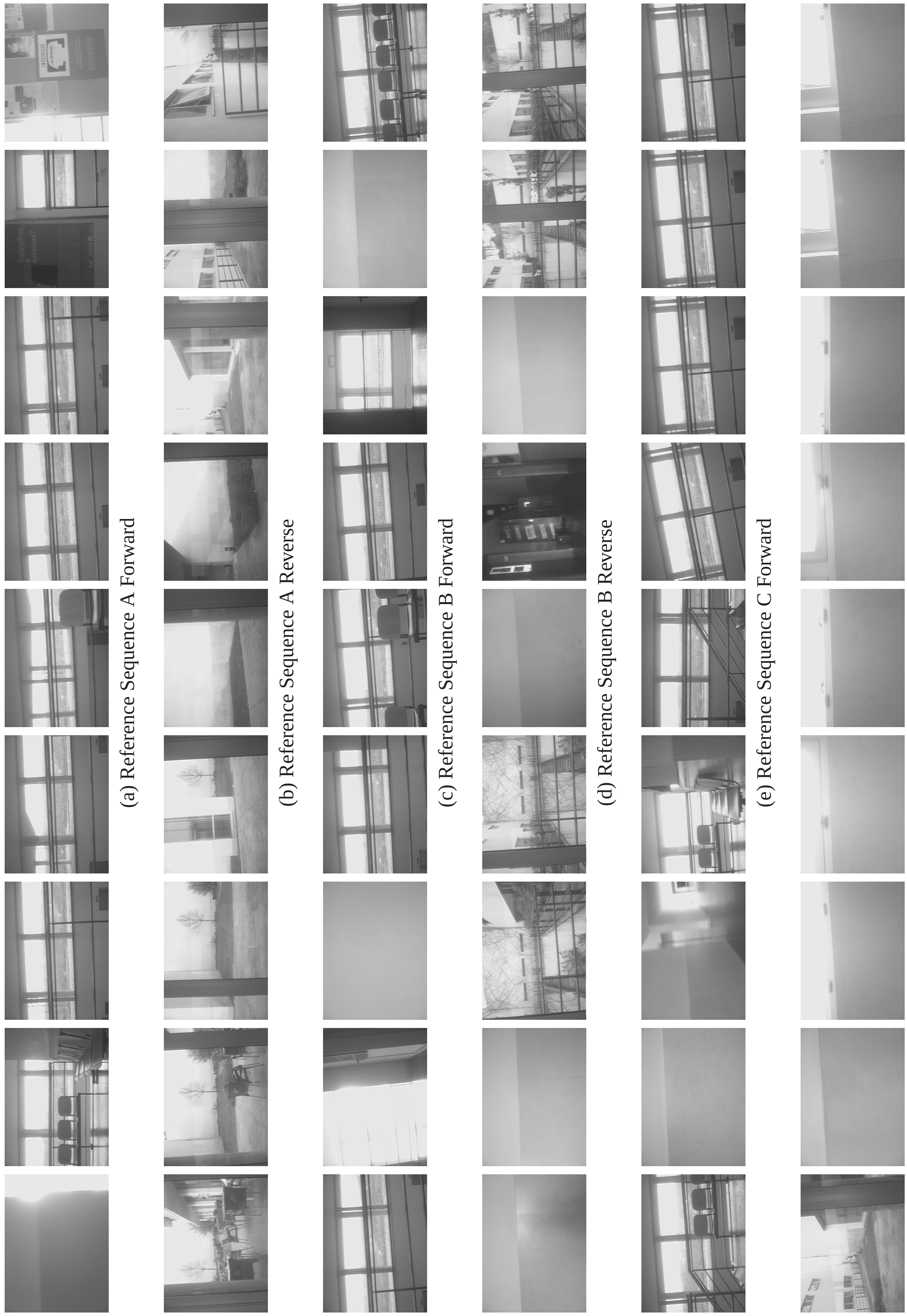
(c) Reference Sequence B Forward

(d) Reference Sequence B Reverse

(e) Reference Sequence C Forward

(f) Reference Sequence C Reverse

Figure 4.11: Some of the Images from Camera #1 that are included in 6 of the 8 Reference Sequences captured prior to merging.



(a) Reference Sequence A Forward

(b) Reference Sequence A Reverse

(c) Reference Sequence B Forward

(d) Reference Sequence B Reverse

(e) Reference Sequence C Forward

(f) Reference Sequence C Reverse

Figure 4.12: Some of the Images from Camera #2 that are included in 6 of the 8 Reference Sequences captured prior to merging.

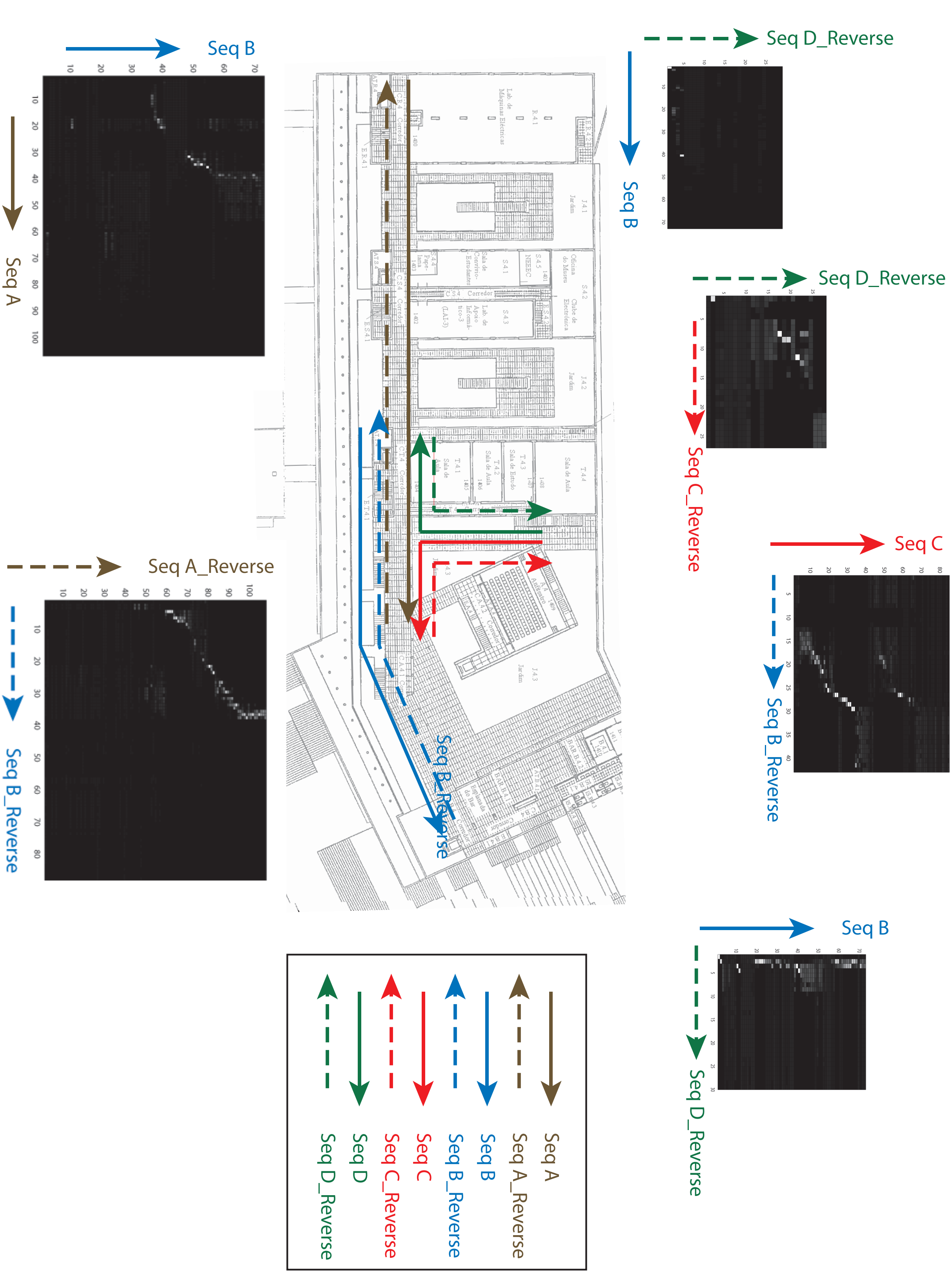


Figure 4.13: The Reference Sequences are compared pair-wise to obtain hypothesis for merging. Similarity between views in the sequences are indicated by high values in the View-Similarity Matrix.

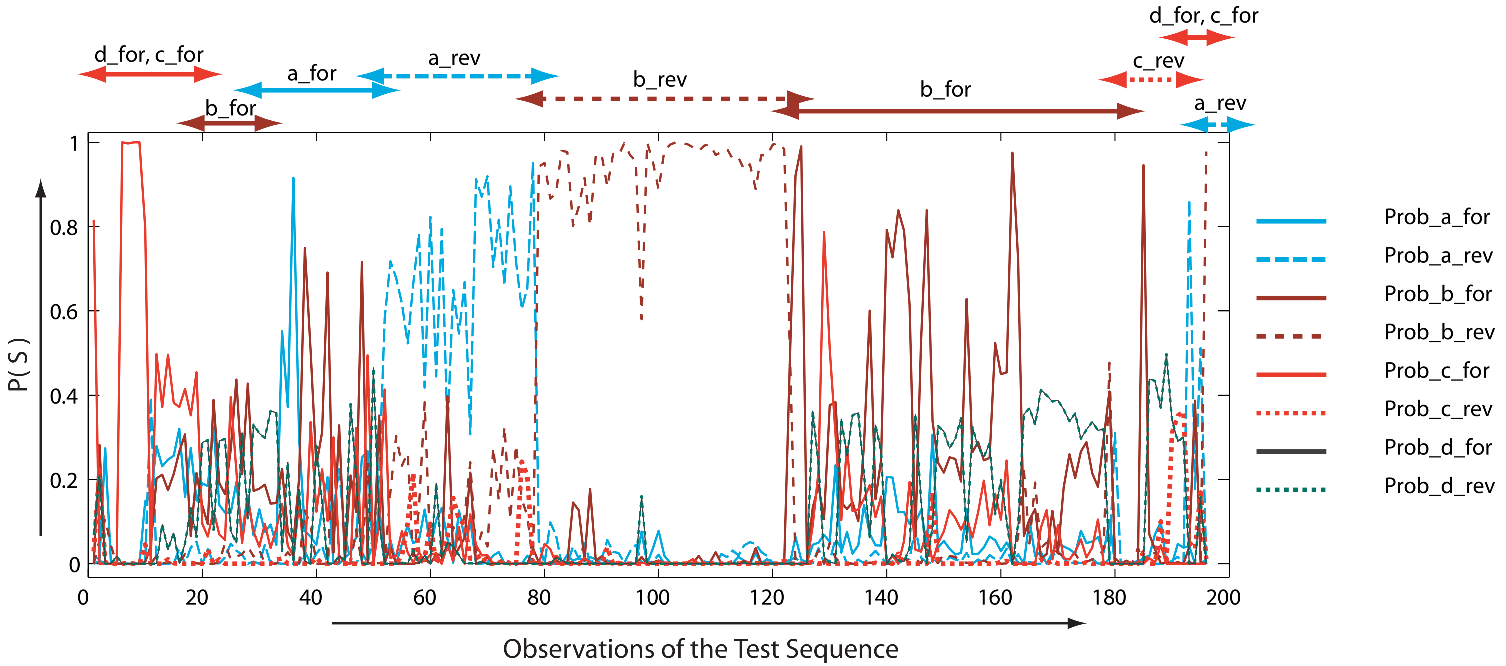


Figure 4.14: The figure includes a plot of the variation in $P(s)$ for each of the 8 Reference Sequences that make up the merged map over the entire path taken by the robot during the localisation test.

4.6 Further Discussion

It is important to emphasise what the ’merging’ of maps does do. Since, in the general case, the view sequences are completely devoid of spatial or geometric information, it is impossible to build up a global map of the environment where the environment paths intersect quite precisely and reliably at the junctions in the environment. Therefore, it will not be normally possible to layout individual sequences on a sheet of paper, based solely on the method to evaluate intersections or cross-overs that is described in this chapter.

In the previous section 4.5, for the first experiment that is described, we drew the merged sequence so as to illustrate the resulting graph structure. In the case of the second example, with the 8 Reference Sequences, the graph was not drawn to avoid the false notion that it is always possible to draw the graph that represents the Topological Map.

Instead, the method that is proposed in this chapter seeks to create a consistent means of keeping the robot localised within one Reference Sequence or another. When a view sequence typically leads into another sequence, it is important that this be reflected in terms of the prior probabilities applied to the place recognition along multiple sequences according to the method described in section 4.3.

The development of algorithms for localisation within geometric maps has progressed from simple, single-hypothesis tracking algorithms to multi-hypothesis approaches that employ particle filters and other methods to boot-strap the localisation process and to recover from possible environment changes and from the robot kidnapping problem. In a similar way, the procedure described in section 4.3 might be modified to maintain multiple hypothesis. Such methods could employ List Viterbi algorithms [Seshadri 94], so as to provide a finite number of hypothesis for topological localisation.

A final point of discussion focuses on the type of sensor and the arrangement of sensors on the robot. In the case of our platforms, since the observations do not cover the entire environment around the robot, it is not possible to merge Reference Sequences that were taken when the robot reversed its direction along a path. Thus, in the second experiment described in section 4.5, it was not possible to merge Reference Sequences A and A Reverse, B and B reverse and so on.

The possibility to do this depends on the sensory capability of the robot.

4.7 Summary

This chapter highlights the representation of the environment using the topological representation of multiple sequences of views. The results are still preliminary but the method offers a powerful and, in our opinion, new approach to the creation of topological maps. A targeted area of application is in the mapping of large environments in which the robot can be taken on selective tours through the environment, potentially speeding up the mapping process.

The global position of the robot is obtained by combining two separate distributions, i.e. the probability of the Robot being along a Reference Sequence and the probability distribution over the views in that Reference Sequence. The specification of the global position of the robot in terms of two separate distributions is consistent with our view of topological maps, as espoused in chapter 1.

The competing possibilities for merging sections of new view sequences will increase with an increase in the size of the map. Exact and heuristic optimisation methods will be required to speed up the procedure. Further work is under way to allow the intervention of a user to improve/correct the process of creation of the topological map. User intervention could be directed towards correcting tentative adjacency relationships on different sequences and to reduce the search space for overlapping segments during the creation of the topological map.

Chapter 5

System Integration and User Interfaces

5.1 Introduction

The methods discussed in the chapters 2, 3 and 4 were developed and tested on Laser and Image sequences captured using two, very different robotic, platforms. These platforms contain very different sensory and motion capabilities. A common robot architecture within which the two platforms could be supported was developed. Over the period in which this research was conducted various modules were developed to allow the acquisition of data and the testing and validation of the various algorithms.

This chapter discusses issues that are relevant to the overall implementation of the system. The original architecture that was proposed for the Place Recognition system and later modifications are presented. This architecture is presented in section 5.2.

Section 5.3 describes the graphical user interfaces that are utilised to create the topological representation of the path and to monitor the results of localisation.

5.2 The Hierarchical System Architecture

The earlier chapters dealt with the algorithms and procedures that were developed to integrate the features and improve place recognition capability. A substantial amount of supporting software

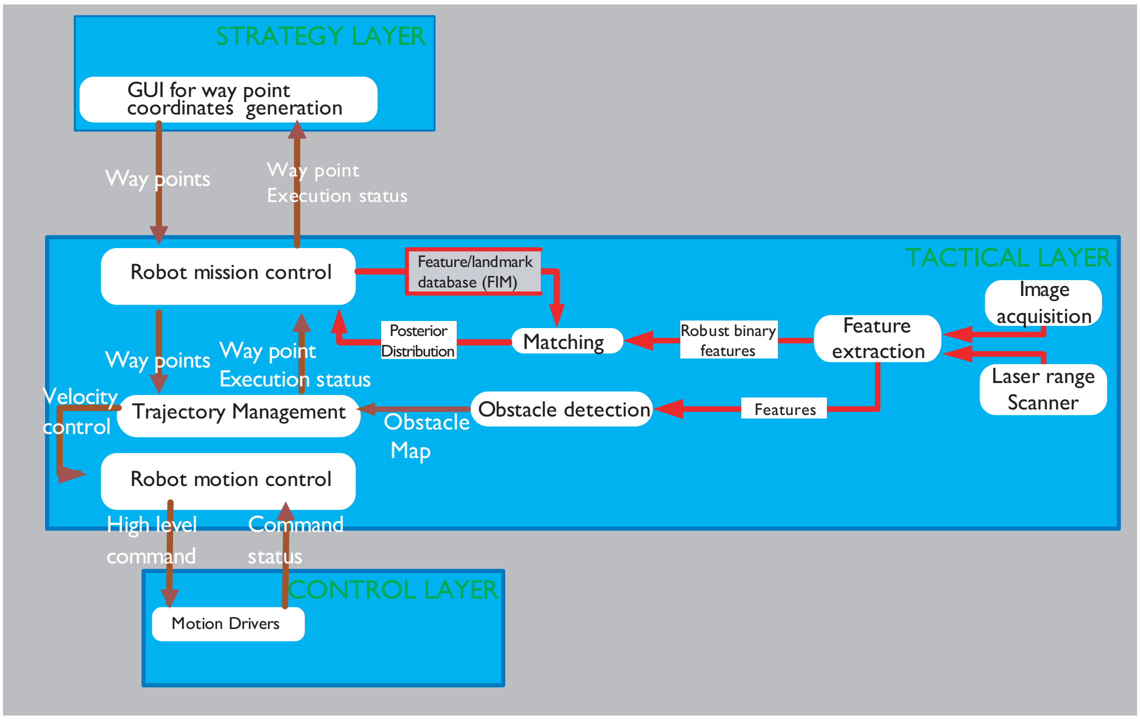


Figure 5.1: An overview of the general architecture of the robot system

that enabled the application of these algorithms and the execution of the experiments demonstrating the Place-recognition capabilities of the algorithms was developed. Function libraries that enabled the extraction of local features from images and laser range scans were developed. Communication protocols were also specified to enable the different processes to run on multiple machines. During the last stage of the work, the proprietary communication architecture that had been developed in the initial stages of this work was substituted for an open-source mobile robotics 'toolkit', CARMEN.

Since the start of this work, the system architecture has been modelled on hierarchical systems [Arkin 89], [Mourioux 04] with some of the layers performing low-level control of the robot hardware and sensors and other layers performing higher level tasks involving navigation and interaction with users and with other objects in the environment.

The architecture is composed, from bottom-up, of three principal layers: the *Control* layer, the *Tactical* layer and the *Strategy* layer as seen in Fig. 5.1. The Control layer is a low-level machine-dependent collection of modules that implement behaviours. These behaviours are functions of the expected data from the robot sensors and the dynamics of the robot platform. The Tactical layer is the intermediate layer within which the entire navigation scheme is implemented. This layer, as shown in the center of Fig 5.1, is composed of modules that control the sensors and process the allothetic data and of modules that allow the robot to move in its environment in the continuous pursuit of the mission goals. Most of the work in place recognition that is presented in this thesis was developed within the extents of the Tactical layer. Finally, the top-level Strategy layer provides the greatest abstraction from the robot motion. User and agent interaction, planning and other interaction with the environment is provided by modules situated within this layer.

The specification of an architecture in modular form requires the exchange of information to occur smoothly and transparently, independently of the way individual modules are hosted. A number of supporting technologies were developed to build up the robot navigation system and run it safely.

In order to achieve the stated aim of portability of the navigation approach to different robot platforms the Robot Mission Control module issues commands from a set of standardized commands that can achieve forward and reverse motion.

Communication is carried out using TCP/IP sockets that ensure that machines with different operating systems can still communicate. This simple communication protocol was extended into a communication library for ease in directing the behavior of the robot from another computer using TCP/IP sockets. ASCII messages are exchanged between the sockets which have the capability to reconnect if required and to synchronise clocks. File transfer capability via the reliable ftp protocol was added in order to reliably transfer images and large structures between different machines. The communication protocol implements a "keep-alive" check for the downstream components. Sensor modules such as the Laser Range Scan feature extractor and the local-image feature extractors are built around these communication modules.

The units of motion utilized by the Robot Motion Control module are standardized to the SI system of units. The required translation to and from the standard of units utilized by each robot platform is implemented at the level of the Robot Control layer. By standardizing the units of the motion, displacement and time parameters and decoupling the polling of information and posting

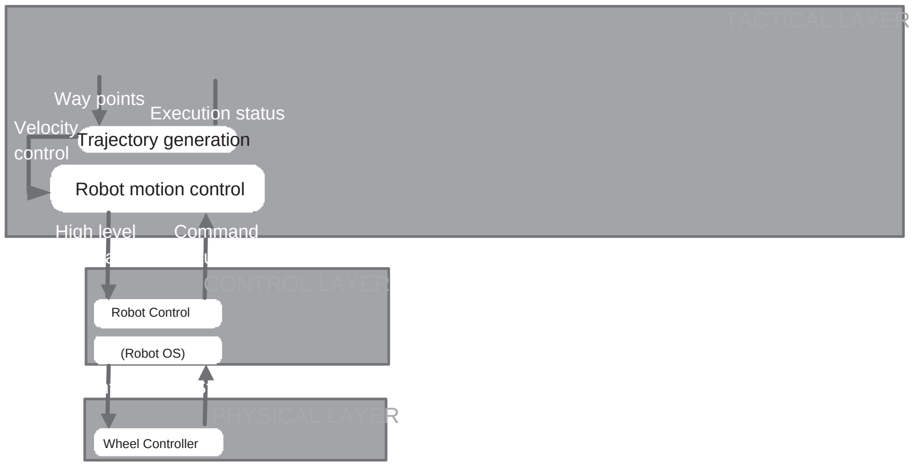


Figure 5.2: Transfer of Information Between Robot Motion Control and the Robot

of commands from the robot sensors and to the motors, the creation and control of a robot object is much simplified.

The Robot Motion Control layer communicates with the Robot Control Module layer asynchronously. A robot-dependent application then acts as an interface between the robot hardware and the rest of the system in the Robot Motion Control layer. The capabilities of this interface vary greatly since it has to translate a common command protocol developed for holonomic and non-holonomic robot platforms. The individual Robot Control layers developed for each type of robot (and their sensor configuration) possess the ability to transmit the commands to the robot hardware via other communication devices such as serial ports.

5.2.1 The Control layer and the Robot Motion Control layer

The Control layer has been defined for two, very different robots: the Robuter robot manufactured by Robosoft, and the RMP 200 robot that is sold by Segway Inc (`http://www.segway.com`). Initial development of the algorithms was performed on the Robuter followed by intensive testing on the Segway RMP robot.

The Control layer defines the behaviours that a particular robot is capable off. In an autonomous mode, a robot mission will be composed of the behaviours that are defined in this layer.

By collecting the platform-specific modules in the Control layer, it was possible to utilize the other layers unaltered for all the robot platforms. The Robot Control Motion layer is primarily responsible for allowing the creation of a standardized protocol of commands that the Control layer can use to communicate with the different platforms. It is responsible for translating the move commands to the robot and to make available to the other modules the state of the robot such as the status after the execution of motion, any idiothetic information that might be available to the robot platform, and other information such as imminent or already occurred contact with some object in the environment.

To ensure safe operation of the robots, the implementation of the modules in the Robot Control Motion layer includes certain checks that verify whether the robot is alive and responsive and that the modules in the rest of the architecture are also functioning and in communication with the robot.

RobotSoft Robuter

The Robuter robot platform, seen in Fig. 5.3, is the main mobile robotics research platform that is available at the Robotics and Automation Laboratory of the Department of Mechanical Engineering, University of Aveiro. It is based on an embedded Motorola 68040 processor running at 25 MHz. The operating system running on this processor is responsible for running the various I/O boards and the communication ports. Through the I/O ports, the processor can control the 24 ultrasound sensors and the robot motors.

The current set-up reutilizes previously developed drivers that integrate motion commands with idiothetic and ultrasound data. These drivers endow the Robuter with collision-avoidance, wall-following, setting robot orientation with respect to objects, etc. Crossing of narrow-openings is achieved by running an application on the robot processor with the robot motion Control loop being closed with filtered data from the sonar sensors.

A high degree of safety is also provided as a result of the use of this driver and of other emergency routines implemented as part of work that is described in [Santos 01]. The same drivers also provide accessible ways of querying the robot's state. Communication with the

Figure 5.3: Arrangement of sensors on the Robuter mobile robot platform. The sensors used in this work include the forward-facing Camera #1, the lateral-facing Camera #2 and the LRF mounted on the front of the robot.

Table 5.1: Translation of high-level commands into robot commands

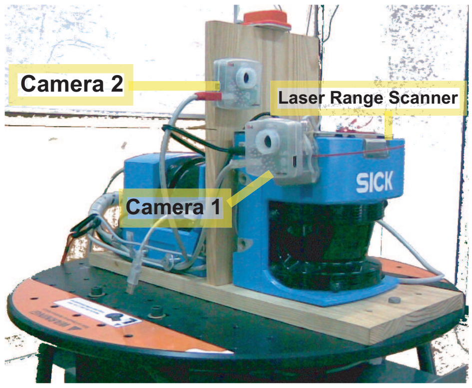
High-level command	Low-Level command	Notes
mva x	AM a a	.
mvb y z	AM v a	.
mvr -t	AM -v -v	Blind backing up; dangerous
stp	AM 0 0	.
Emergency Stop	AM 0 0 and SERV OFF	Kill motion and lock motors

robot is maintained through the use of text messages relayed via two serial lines.

The main messages to the Robuter consist of an adaptation of the command from the Robot Motion Control module of the Tactical layer. Velocity-limited messages are emitted at a frequency of 5 Hz. The robot replies only with an "ack" that ordinarily indicates syntactical correctness but is also taken to mean that the robot is alive and receptive to commands. The command set that is utilized and the translation from the standard communication protocol used by the Robot Control Motion module to the Robuter specific commands are stated in the table 5.1.

A second type of message corresponds to a request for the state of the robot. These commu-

(a) The Segway RMP 200.



(b) Sensor Platform consisting of Camera #1 and Camera #2 and Laser range Finders mounted atop the Segway RMP200.

Figure 5.4: The Segway Robotic Mobile Platform (RMP) and the sensor platform.

nications are maintained at twice the rate at which commands are emitted by the Robot Motion Control module, i.e. at 10 Hz.

Bumper control and the front sonar readings are utilized to stop the robot in case of actual collisions or upon the detection of obstacles. Information about the emergency provisions that are activated is passed up from the robot with the rest of the status information.

Segway Robotic Mobility Platform 200

The Mobile Robot Lab at the Institute of Systems and Robotics, Coimbra is equipped with independent sensor and Robot platforms that are supported by modules developed by this lab and by other mobile robotics laboratories within CARMEN.

The modules required to run the Segway RMP have been developed within the CARMEN software platform, with IPC providing the communication support. Improved sensor logging formats, allowing faster parsing and support for additional sensors was carried out.

Through the use of a library developed at the Mobile Robotics Laboratory at the Institute of Systems and Robotics - Coimbra, the robot processor is able to directly access the motors and the inertial sensors that provide the odometry information. The system under CARMEN is well integrated although lacking a comprehensive low-level safety and basic motion routines such as those available on the Robuter. The RMP is run with a collision avoidance system that uses data

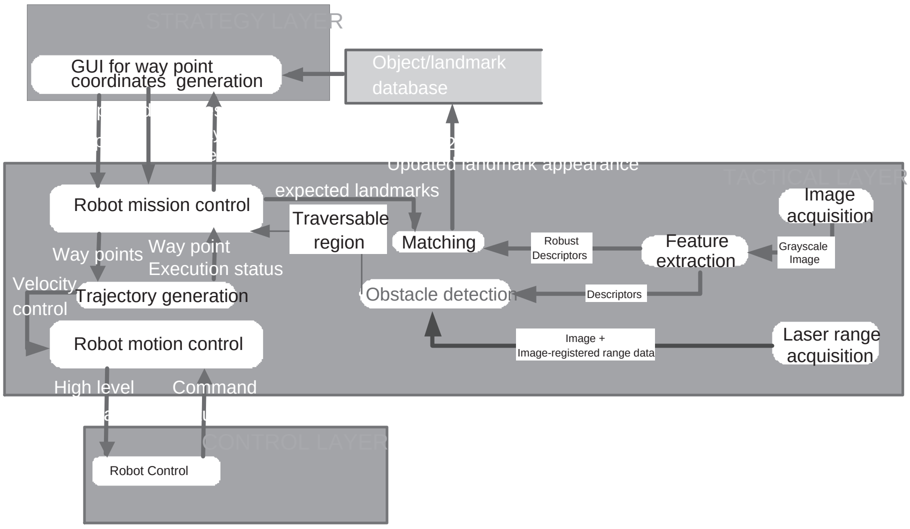


Figure 5.5: The Tactical layer in the Robot Architecture

from the front-facing LRF.

5.2.2 The Tactical layer

The modules in the middle layer, the Tactical layer, are accorded the responsibility of monitoring and controlling the navigation tasks. The Tactical layer has three modules that control the robot motion. These are the Robot Mission Control module, the Trajectory Management module and the Robot Motion Control module. These modules, denoted in Fig. 5.5 serve to:

- Collect features processed from the sensor data: The Tactical layer provides the structures to handle combinations of features that define the individual places in the environment.

- Recognize previously trained locations: The Tactical layer provides methods to match the current views with the previously collected views contained within the topological representation of the environment.

- Move the robot in a way that allows the completion of the robot mission: Associated with each place in the environment representation is a motion or behaviour that will take the robot to the next represented place in the environment.

The Trajectory Management module is associated with keeping track of the appearance (disappearance) of objects around the robot in order to allow the robot to trace a viable and consistently similar path through the environment. Finally the Robot Motion Control executes the motion required by the Trajectory Management module. Currently, a reduced capability is provided in the form of a collision avoidance.

The *Robot Mission Control* module is responsible for keeping the robot on the topological map. The Robot Mission Control module, provides the *Matching* module with prior probabilities and probability of transitions between the places, information that will be used at the time of matching fresh data.

By continuously matching fresh sensor information with the previously collected parts of the environment, the *Matching* module provides am estimate of the position of the robot in the environment. As described in chapter 3, the consistency of the matching algorithm is improved by using the BMM within a Hidden Markov Model, HMM.

The actual features that are visible all through the environment are stored in the Landmark/Feature database. In the context of work developed in this thesis, the topological representation and the features (environment signatures) that are associated with the environment are gathered during the course of an initial environment-familiarisation phase in which the robot is moved around the environment.

The perception process consists of self-contained ’sensor classes’ that are entrusted with the task of extracting features from the current sensor data. These features are matched against a previously created database of features and provide the Localisation modules with a list of feature IDs that correspond to the current features being observed. The database of features can be explicitly specified as in the case of the LRF that detects doors, walls and corners, or it might be built automatically as in the case of vision, where the database of conspicuous local image features are collected during a previous Environment-Familiarization phase.

5.2.3 Strategic or Interactive layer

The Strategy layer provides the highest level of abstraction to the motion of a robot along a topological path. Path planning, interaction with other agents in the environment and communication/monitoring are tasks that fall within the scope of the Strategy layer.

In the course of work that is reported in this thesis, the high-level abstraction role that is contemplated for the Strategy layer was not developed much. In a real-world scenario, the Strategy layer provides the framework for interfacing the Place-Recognition and Autonomous Navigation capabilities of the Robot with a more complete Global Map of the environment with a user interface that is used for communication and cooperation capabilities with other robots.

At the time of initiating a mission, the topological structure of the environment and the required features are transfered to the Robot Mission Control module in the Tactical layer, which subsequently assumes control of the robot.

In the section 5.3, the provision of some user-interaction capability is demonstrated using which the user can specify the execution of certain tasks at specific places in the environment.

5.2.4 Integration with CARMEN

Recently, a major modification was made to the whole concept of communication between the various modules in the above architecture. The open source project, CARMEN (`http://carmen.sourceforge.net`), and its IPC application was adopted as the basis for interprocess communication. All the sensor and processing modules register with the IPC central process as soon as they come alive. At the time of registering, the module subscribe to a particular message (processing modules) or offer to publish it (sensor modules). Such a proven communication architecture greatly simplifies the development and deployment of modules on different machines connected via a TCP/IP network, Fig. 5.6.

As can be seen from the Fig. 5.6, the architecture supports a number of additional sensors besides the LRF and cameras used in this work.

Figure 5.6: Inter-process communication is now performed using CARMEN programming style and the IPC-central application. Shown above is the Laser range Finders, camera and pose sensor mounted upon the Segway RMP robot. All communication is performed using CARMEN.

5.3 Program Interface and User Interaction

User interaction capability has been provided to monitor the Environment Familiarisation and the place localisation phases. The design and implementation of different forms was undertaken. These forms and dialogs allow the control of the data acquisition procedures and the visualisation of the localisation results.

5.3.1 Environment Familiarisation Interface

The Environment Familiarisation phase involves leading the robot once through the environment. The order of the places in the environment is recorded. In the case of those features that are matched against a 'bag of features' such as SIFT and the HU moment vectors, the Environment Familiarsation phase also serves to collect these features that will be subsequently used during localisation.

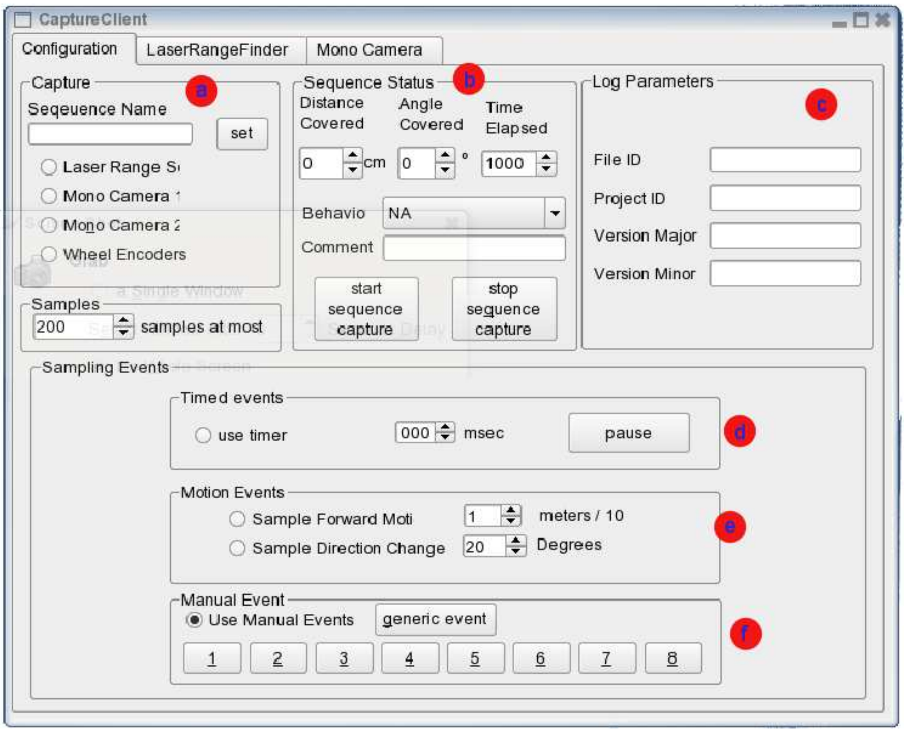


Figure 5.7: Program Interface for Environment Familiarisation

The dialog for the Environment Familiarisation phase, shown in Fig. 5.7, allows some degree of control of the sampling procedure. The forms in the dialog allow the set up of the Environment Familiarisation phase and of events that control the sampling during the Environment Familiarisation phase. We briefly describe below some of the fields that appear in the form pictured in Fig. 5.7.

a) The Capture form sets up the name of the Mission and the sensors that will be used.

b) The sequence status dialog allows the user to monitor the Environment Familiariation phase and start and stop the phase.

c) The Log Parameters form provides additional information on the sensor data sequence that is being collected. The information is maintained in the CARMEN log files and in the XML files.

d) The Timer control form allows the sampling of the environment using a timer event.

e) The Motion control form allows the sampling of the environment using events triggered by the motion of the robot.

f) The Manual control form allows the triggering of custom events.

The interface also allows the user to visualise the robot scans and the images that are currently being captured by the robot.

5.3.2 Localisation Interface

The interface of the localisation algorithm needs a dialog to introduce the place that the robot is expected to reach and any task that the root is supposed to perform. To test the improvements that were put forth in this thesis to improve the place recognition along the path, the user is allowed to monitor the procedures and view the results via the interfaces shown in Fig 5.8. These are provided in the form of a tabbed interfaces, each of which address one following phases a) loading of the Reference Sequence, b) Monitoring and Visualisation of localisation statistics, c) compacting the Sequences and d) Localisation along the Topological Sequence.

The first tab, seen in Fig. 5.8a allows the user to load a particular Reference Sequence. The Mission Monitoring interface, Fig. 5.8b, interface allows the user to monitor the successful execution of the mission by verifying various plots and position probability distribution plots. The Mission Execution interface, Fig. 5.8b, allows the user to visualise the arrival of the robot at the final goal and at intermediate places. At each place that is specified in the Robot Mission, the robot executes the tasks specified in Robot Mission. The motion behaviours that take the robot along its path are included in the Reference Sequence.

The tab shown in Fig. 5.8c, allows the user to run an algorithm that seeks to create a more compact Reference Sequence. The algorithms that perform this procedure constitute work in progress and have not been described in this thesis.

The Robot Mission interface, Fig. 5.8d, allows the user to specify the goal of the mission in terms of the intermediate and final position to be occupied by the robot. The interface also includes the tasks that must be performed at different places as the robot proceeds toward its goal.

(a) Program Interface to Load the Reference Sequence

(b) Program Interface to monitor the Mission Execution

(c) Program Interface to create Compact Reference Sequence.

(d) Program Interface to Specify the Robot Mission

Figure 5.8: Program Interfaces for place recognition

5.4 Summary

The techniques developed in the previous chapters and the algorithms described were initially tested on simulated data followed by an implementation in C++ to achieve the performance required for localisation on a real world system.

During a later stage of the work, the Carnegie Mellon robotic toolkit, CARMEN, was incorporated leading to enhanced sensor support and an access to a reliable communication protocol between the software modules.

Working applications were required to demonstrate the usability of the place recognition method in the real world. Graphical user interfaces were written so that the end user can control and monitor every step, starting from the creation of the topological path to the identification of the current position of the robot within the original path.

Chapter 6

Conclusions and Perspectives for Future Work

6.1 Introduction

This chapter lists some of the contributions of this thesis and points to a few topics as directions for future research. Most of these contributions lie in the application of algorithms and methods that have been borrowed from other areas and applied to the problem of mobile robot localisation.

In section 6.2, the principal contributions of this thesis to the state of the art in place recognition have been listed.

Section 6.3 lists some of the publications that involved work that was developed within the scope of the doctoral program.

In section 6.4 a few incremental modifications that could simplify the models that were applied and improve the results of the Place recognition are presented.

Finally, in section 6.5, a few closing comments have been presented.

6.2 Contributions of this Thesis

In the search for an appropriate representation for the sequence of views gathered whilst leading the robot along a path, an interesting insight was obtained into the nature of geometric and topological maps. A comparison of the Metric and Topological approaches to the representation of an environment highlights the fact that the Metric map allows a single joint probability to represent the entire environment whereas a topological map uses two or more conditional distributions to represent the same. This perspective, from a point of view of probabilistic events, appears to be a very consistent way of looking at these two commonly used representations, each of which has a set of associated advantages.

This insight, led us to develop a topological representation of the view sequence in which the transitions between views lying on the same path are constrained using a novel graph network. This resulting representation forms the basis for our method of localisation along a topological path. The views that are obtained at places that lie on the topological path are stored in terms of features that are extracted from the images and from the laser range scans.

The resultant data structures and the algorithms that are required to represent the topological path and to perform localisation have been found to work effectively for paths of around 200 meters in length.

The localisation procedure was tested with success in indoor environments, as seen in the earlier chapters and with a lower degree of success in outdoor environment, see Fig. 6.1 for a FIM from an localisation experiment that was performed outdoors.

In order to move from single paths to real environments with multiple intersecting paths, new procedures and data structures were developed. To enable the use of the procedure for larger environments more compact data structures must be employed.

6.2.1 Conversion into Binary Features

The need to utilize views that are composed of data from multiple sensors led to the conversion of information into binary form. This conversion and the subsequent integration of multiple Binary Features allows the inference of the position of the robot along the sequence of views.

Each place in the environment is represented in the form of a vertical column of binary features, all of which, when put together, result in the Feature Incidence Matrix, FIM. Each line in the FIM represents a particular and distinct feature. The FIM is created by first leading the robot along the desired path during an Environment Familiarisation phase.

The problem of inference is characterised by high dimensionality because of the large number of features from the multiple sensors. Since the views are represented in discrete form, as binary vectors, more conventional dimensionality reduction methods such as Principal Component analysis or Gaussian Mixture models could not be employed.

The reduction in the dimensionality reduction was achieved by modelling the data as a mixture of Bernoulli distributions, a method first used in Data Mining applications [Nadif 98], [Gonzalez 01]. The Bernoulli Mixture method was shown to be effective in the integration of many hundred binary features. This work is detailed in chapter 2.

6.2.2 Use of Contextual Information of the Topological Path

The current position of the robot within the graph network that represents the topological map must be inferred in the face of noisy sensor data and uncertainty in the distance covered by the robot. The adoption of Hidden Markov Models to enable the use of *contextual information* present in the topological path proved to be relatively straight forward. An original contribution was the addition of the $Lost\_Places$ of the Robot that allowed the incorporation of the concept of a sampled Reference Sequence. This work is detailed in chapter 3.

6.2.3 Scalability and Usability Improvements

The line-graph model of the topological path was shown to be effective while performing place recognition. Real-time localisation using a conventional personal computer was obtained with a position estimate obtained up to five times per second for paths around 200 meters in length.

In order to have an effective place recognition system for large environments and for environments within which a robot can take many distinct paths, modifications were introduced to the above method.

Figure 6.1: The path taken by the robot is represented in the form of a sequence of views. Each of the 153 views taken over the 300 meter-long path within an open-air car park is represented as a vertical vector of binary features in the Feature Incidence Matrix, FIM. The FIM, depicted graphically at right, counts around 8000 different features.

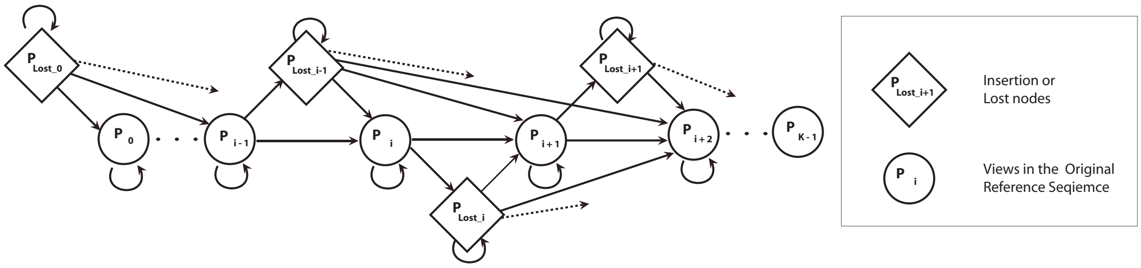


Figure 6.2: During localisation, the inserted '$Lost\_Places$' accommodates the robot wherever it has not localized in the Reference Sequence. The robot is now assumed to start out at the Place '$P_{Lost\_Place\_0}$', The dotted lines indicate the transitions to each of the Places in the original Reference Sequence which have not been drawn to avoid cluttering the figure.

The Feature Incidence Matrix, FIM created after the Environment Familiarisation phase describes the environment along a single path. In order to map any general environment, multiple runs of the robot, along different routes, are proposed. By ensuring a minimum amount of overlap between the different routes taken by the robot, the individual topological paths can be stitched together to create the global topological map over the regions covered by the robot.

The method, depicted in Fig. 6.3, compares sections of each topological path pairwise with each other path. By making an assumption that the same region is not covered twice within the same path during the Environment Familiarisation phase, we avoid having to compare sections of a topological path with itself. An Information theory approach was utilised to test the similarity of a path segment with some previously captured path. This work is detailed in chapter 4.

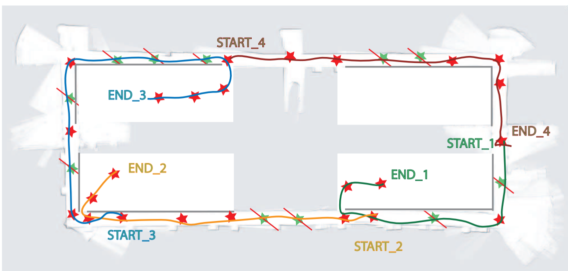


Figure 6.3: A depiction of the merging of multiple paths to create a topological map of the environment.

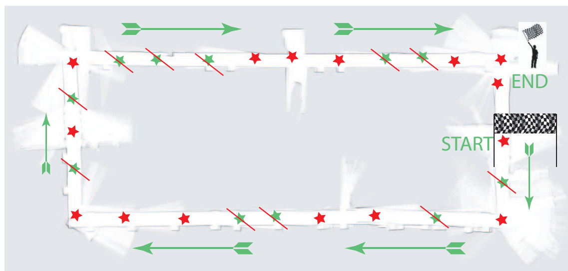


Figure 6.4: A depiction of the removal of views that are non-distinctive resulting in more compact view sequences.

As the size of the environment to be mapped increases, the views that are maintained also increase, typically resulting in an increase in the number of features. This large number of features worsens the dimensionality problem and results in the need for increased computation.

Some of the views, the distinctive views, in the sequence must be maintained since it is important that the robot recognise the corresponding place in order to be able to navigate the map. In certain cases it might be possible to eliminate *some of the views* without worsening the results of place recognition. The procedure is depicted in Fig. 6.4. An exact solution could not be found for the problem of selecting the non-distinctive views, i.e. the views that can be safely dis-regarded.

The above improvements were implemented and the viability of using this method for larger environment was tested with partial success. Additional development and testing is required in order to create a complete topological map-building method for large environments.

6.2.4 Improvements in Feature Detection

Contributions were made to the application of SIFT features to image sequence by allowing the addition of SIFT keys to the KDTree as new images are added to a sequence.

An assortment of features extracted from the laser range scanner were utilised, some of which, to the best of our knowledge are original. The Hu moments that are used to match laser range scans are easy to construct, have been well studied in the context of intensity images and have been shown to be robust and repeatable features when applied to laser range scans.

6.3 Publications

The list of publications that were prepared and presented/published over the course of the last 4 years include:

- In [Ferreira 04a], it was shown how various researchers were converging to the use of multiple scale methods together with histogram representation methods to store and process features used to characterize image data. Using software developed by us albeit still in early stages of development, good image matching results demonstrated the possible inclusion of such a scheme in robot navigation.

- In a presentation in the same year [Ferreira 04b], preliminary results from the inclusion of depth data in the process of extraction of interesting points in the image was also presented. This work was not developed further in the course of this thesis and has not been presented in this thesis.

- At some point at the end of 2004 the requisites for representing the environment in the form of a topological graph were identified. In [Ferreira 05] a mathematical basis to allow the integration of features obtained for perception for topological navigation using the concept of entropy was developed. This article described the state of the work in a very preliminary stage and represents the first first steps in arranging a common representation for all types of features (from all sensors), which eventually culminated in the usage of a set of binary features.

- The common framework that was found to represent features from all sensors was published in [Ferreira 06]. This article reported on the development of a Bernoulli Mixture model to integrate features extracted independently from two or more distinct sensors. Local image features (SIFT) and multiple types of features from a 2D laser range scan were all converted into Binary form and integrated into a single binary Feature Incidence Matrix (FIM). This is the method that is described in detail in the next chapter.

- The process of application of Hidden Markov Models to the Localisation task and the results showing the resultant improvement and increased robustness of the results have been published in [Ferreira 07a] and in [Ferreira 07b].

6.4 Perspectives for Future Work

This work has described techniques that improve the place recognition capabilities of the system using a method for un-biased View matching and parameters for the HMM that are appropriate for the problem. Some modifications have been planned and are in varying stages of executions.

While these modifications are expected to improve the place recognition in certain environments and to better adapt the HMM models to the problem of Place Recognition, they do not significantly alter the approach that has been presented in the course of the thesis. They represent an evolution of the work presented in the earlier sections and have been ear-marked as future work since they could not be completed within the time frame allocated for the completion of this thesis.

6.4.1 The promise of Improved Feature detection

The Topological map, when modelled in the form of the Reference Sequence is defined in terms of two conditional distributions. The transition probability matrix addresses the change of the hidden state and acts as a sort of consistency check for the underlying sequence of states that explain the sequence of observations. The emission probability matrix serves to handle the uncertainty of the observations so that the underlying state can be identified with greater or lesser certainty. The Viterbi algorithm uses these two distinct distributions to find the most plausible underlying sequence of states/places.

Thus, there are two ways to improve the results of localisation. Improvements in the results of sequence matching /alignment procedure using the Viterbi algorithm depend on the appropriateness of the state transition probability matrix and a better observation probability matrix.

Among the two approaches it is known that the cost of discrepancies in the 'log-likelihood score of a match is usually dominated by the output probability' [Mitchell 95b]. This observation implies that improvements in single view matching techniques would make important contributions to the sequence matching procedure.

In the case of Place recognition in outdoor environments, improved results are conditional on the design and implementation of better features. While the currently used camera and laser features for appearance-based place recognition provide convincing results for the indoor environments, the same features were found to offer relatively weaker results in the case of outdoor environments.

Future work must include the development of better features for outdoor environments. These features will exploit the relief and 3D structure of the environment through the use of robust features from 3D point clouds.

6.4.2 Further Development for the Integration of Binary Features

In future work, Communication Theory can provide some insight into the problem of the simultaneous handling of multiple binary features. The acquisition of a sequence of raw views can be modelled as the communication of a sequence of bits over a noisy channel. The state of the environment S is coded within a 'word', V_{map}, and is communicated, via the sensors, to the localisation system. This word might get corrupted and is actually received as V_{obs} from which the original state must be estimated in the form of $\hat{S}$. The process has been depicted in (6.1).

$$S \rightarrow V_{map} \rightarrow V_{obs} \rightarrow \hat{S} \tag{6.1}$$

A solution to this problem must include the creation of a model for the individual feature uncertainty and the use of techniques from the error code correction literature to recover the original word.

The detection (or lack of) can be modelled in various ways depending on the characteristics of the sensing procedure. Fig. 6.5, at left, shows a feature modelled as a 'Z' - binary stochastic variable (a probable model when the features are quite unique from each other as in the case of SIFT features), while Fig. 6.5 at right, depicts the model of the feature as a non-symmetric binary stochastic variable (which could be used to model features that are similar or impossible to tell apart).

Figure 6.5: Viewing perception as a transmission of a single bit across a noisy, non-symmetric binary channel.

Communication theory might also help us evaluate the choice of feature combinations that are used to represent the individual places. As described in the earlier sections, each of the views is composed of multiple features. Feature combinations that are used to represent different states of the environment should be as unique as possible. The more similar the composition of two views is, the more difficult it is to tell these two views apart.

The theory of noisy channel coding helps us find answers to some questions such as :

a) What is the maximum M number of codewords that can be transmitted given n bits can be transmitted (or how many bits are necessary to transmit a set of M codewords)? This is actually the rate R of the channel. R is defined as being equal to $\frac{\log_2 M}{n}$.

b) If errors in the transmission (and subsequent decoding) up to a maximum number of bits are allowed, which codewords can be recovered given a particular Block code (FIM)?

Claude Elwood Shannon provided the answer to the first two questions in his seminal paper [Shannon 48]. He showed that the maximum information that can be transmitted across a noisy channel, its channel capacity, is equal to the mutual information rate between the input and output of the channel. This mutual information is a property of both the expected noise and the initial or marginal entropy of the input to the channel.

A value that is analogous to the 'Channel Capacity' might be developed for the sensor system. This 'Perceptual Capacity' will specify the maximum rate of the perception channel in the presence of an optimal distribution of the features in the environment. In case the errors that affect each feature are correlated, the capacity of the channel automatically reduces since the entropy of the source itself is reduced.

The corollary of Shannon's work would be not to use the noisy channel to communicate beyond the channel capacity. In the analogy with our localization system, this signifies that when sensor data is noisy, the quality of localisation and of place recognition might be improved by increasing the 'Perceptual Capacity', i.e. employing more uncorrelated features or by using better filters to guess and correct the errors that appear in the features.

6.4.3 The specification of Robot Motion behaviours.

There are many works in current and past robotic research that deal with discovering new strategies to be applied to robot motion or alternatively the procedures that could be adopted so that robot motion can be specified using only a topological representation. This work has not addressed this problem although the title of the thesis indicates that it was an original intention. as it did not pretend to advance the state of art in these areas. The primary focus of this work remained the representation of the environment using data from multiple sensors.

During the execution of this work it has become clear that a complete separation of the mapping representation and the motion specification is not possible in the case of a complete navigation system.

Importantly, the organisation of the map and the kind of representations that are utilised might actually exclude certain types of motion behaviour. The motion that was possible with such a representation was taken as a consequence of this environmental modelling scheme.

6.4.4 Using Simpler Models for Path representation

The introduction of the Lost-Robot Place in between the original places from the Reference Sequence, allows for the enhancement of the applicability of the HMMs in many ways. One such usefulness of the inserted $Lost\_Places$ was introduced in chapter 3, where the $Lost\_Place$ is used to account for the observations that cannot be predicted and which lie between known observations.

There are other ways of modelling the above requirement, such as using another variant of the Markov chain that defines the transition between states in the HMM, such as the Meally Model. In such a model the observation is made during transitions between hidden states. Such HMMs are commonly used in speech recognition and a segment of an HMM using such a model with Explicit duration modelling is shown in Fig. 6.6. Given the large body of experience that there exists in using these models, their adoptions should result in greater adaptability of the HMM models applied to Localisation.

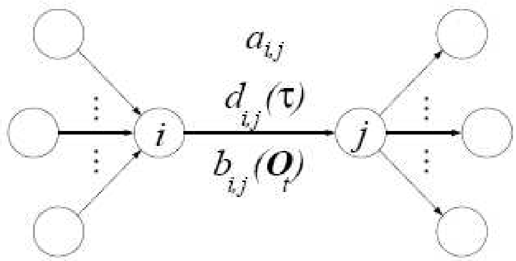


Figure 6.6: HMM model that follows the Mealy Model with Explicit state duration, from [Mitchell 95a]

6.4.5 Creating Compact and Efficient Representations

Up to this point in this document, the Reference Sequence that was used for localisation is the one created by sampling the environment according to some sampling plan. It is conceivable that in some cases improved results might be obtained by altering the Reference Sequence and removing certain views that do not provide enough information. Better localisation could result from the improved prior and place-transition probabilities over the fewer and more informative places. Additionally, by identifying these distinct places a more compact Reference Sequence could be created. Shorter, more compact Reference Sequences are desirable in applications involving communication between robots/persons having different capabilities and limited computing power or communication bandwidth. Compact representations of the Reference Sequence are also desired when long paths through the environment must be traversed, resulting in faster localisation.

The problem of Localisation within a Reference Sequence is akin, to the problem of supervised sequential learning [Dietterich 02], albeit, in our case, for a single presentation of the training data set. In [Dietterich 02], Dietrich identifies three important issues that must be addressed in the case of sequential machine learning namely, specification of a loss function, feature selection and computational efficiency. A schematic for a definition of Sequential Machine Learning is shown in Fig. 6.7. Localisation within the Reference Sequence as defined in chapter 1 can be viewed as a sequential learning problem, albeit with special characteristics where the *data samples* are actually the views that were first introduced in chapter 2.

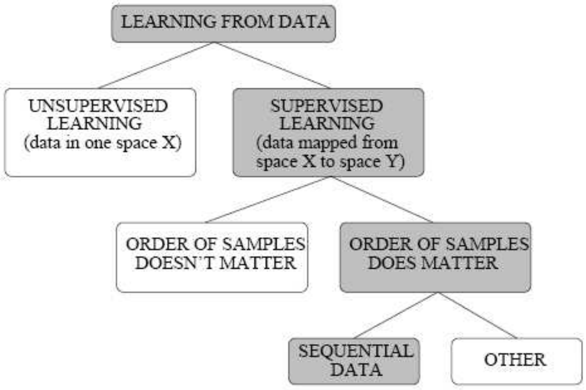


Figure 6.7: [Comparative] definition of the Sequential Machine Learning Problem [Schuster 99].

The *features* in the terminology applied by Dietterich [Dietterich 02] are analogous to a subset of *distinctive places* in the Reference Sequence. In order to improve the results of the localisation algorithm, distinctive places must be maintained in the topological map so that they improve the results of localisation and represent the environment in the form of a compact Reference Sequence.

The compact Reference Sequences no longer can be used in conventional HMM since the frequency interval of the observations is not regular. To continue to utilize HMM-like models, the 'duration' or the elapsed time between observations must be utilised.

Using Second order HMMS might be a good idea where a number of 'classes' of HMMs are applied [Aycard 04].

At the cost of greater computational cost and model complexity, the explicit modelling of the duration for which the system remains at a state can be included in the Viterbi algorithm. In order to create a robust system that is not tailored to any one particular environment, the choice of what is implied by distinctive place or view should be defined based on the views in that specific environment.

The transition probabilities of an HMM can also be expressed in terms of the 'duration' or the average value of a measure (say time or distance) for which the system remains at a state before changing. This approach is much more intuitive in the case of certain left-right HMM that are used to model speech, handwriting or music. The variable of interest in these problems is time and often the pace at which the system changes its states is a key aspect of the model. Various distributions have been employed to model these durations including some of the more well-known parametric distributions such as Exponential, Gaussian and Beta distributions [Levinson 86], [Pikrakis 06], [Kundu 98], [Chen 95].

After a comparison on literature on the various duration modelling methods, the post-processor duration model by [Juang 85] was implemented and tested. Although the results of localisation using this approach on the compact Reference Sequence have not improved terribly as compared to the results reported in chapter 3, this method represents the first steps in the localisation of non-uniform Reference Sequences. Further research into the use of duration-based HMMs is required.

6.5 Closing Comments

This thesis presented a technique to index a sequence or stream of sensory data to some variable of interest (position of the robot in this thesis). By converting all the sensory information into binary features and then reducing the dimensionality of the 'space' occupied by these binary features using a Bernoulli Mixture Model, a powerful index retrieval mechanism was developed which has shown good applicability to the problem of recovering a robot's position in indoor environments. The 'ordering' information that is implicit in the sequence or stream came to be utilised by integrating multiple and sequential observations using a Hidden Markov model.

While improvements in the Localisation capability of this algorithm are expected as environment-appropriate and reliable features are developed, a strong advantage of the combination of the underlying techniques is that they could be applied to other scenarios and problems.

Another original contribution was the modification of the Markov Chains used in Hidden Markov Models to enable the use of the sequential context in which the expected observations are specified in the navigation Mission.

Other contributions include developments that were made in the characterisation of images through the application of local features and of laser range scans through the creation of original features based on the scan contour and free-area properties.

Appendix A

Fast Extraction of Local Image Features

The use of cameras on mobile robots has become widespread over the last few years. Cameras can be viewed as high band-width sensors and images can have large redundancy. However, it becomes 'expensive', in terms of memory and computational costs, to store every raw image that is associated with a place or with an object.

The sensitivity of the cameras and the effect of changes in light, changes in viewpoint and multiple reflectance means that individual pixel values are not easily reproduced. A pixel-wise comparison between pair of images will work only if there are no major changes in the illumination and the viewpoint. This problem implies that images taken of the same scene, but at different times will appear different.

For this reason, images are often compared and matched through the intermediary of 'features' that are extracted from the images. Feature extraction for vision-based robots varies from local image descriptors to global image properties derived from the intensity distribution over the entire image. Local image descriptor methods seek to store a few pixels or measures derived from these pixels whereas the latter methods calculate certain interesting properties of the image as a whole. As implied from above, both methods seek to avoid having to store the entire image itself.

The main goal of the thesis proposal was to develop methodologies to localize a robot within a previously constructed Reference Sequence of Views.

In the study of image description and image correspondence, global image features, by their nature convey information about the entire scene in a single descriptor. Such approaches attempt to calculate properties (e.g. moments of some value) of interest within the image or a substantial portion of the image. A commonly used approach is to attempt to capture the distribution of intensities in the colour space using a histogram [Andreasson 04]. Along with area-based methods there are other contour based methods which seek to code the properties of contour of a regions and use novel ways to match these properties [Iivarinen 97], [de Trazegnies Otero 04].

As such, global features represent a compact way of presenting information from scenes if these are not expected to change significantly. Local-image features on the other hand work on image patches and, when taken together, provide information about the scene as a whole. In the case of images of scenes taken by a mobile robot, illumination changes, occlusion as a result of dynamic object, scale and view point changes all suggest the use of local image features. Only local image features have been utilised in this work.

A.1 Local Image Features

Certain regions in an image are known to be [more] stable with viewpoint and lighting changes. The use of local image descriptors based on these stable regions is characterized by two steps 1)the selection of points of interest and 2)their characterisation. The selection must be repeatable (even with changes in the conditions in which the images are taken) and the characterisation must employ properties that must, again, be tolerant to changes in the viewpoints, lighting and other conditions.

Using combinations of heuristics and parametric methods, features that correspond to the architectural properties of certain environments can be extracted. Such methods have used planes [Corso 03], kerbs [Se 97], the ground plane [Se 97] and roads. Other works such as [Torralba 03] have used the output of the application of wavelet image decomposition.

A competing and more popular approach is to create descriptors for regions of the image which are then matched against a bag-of-feature. While this method might be blunt in the case where there is strong prior knowledge of the nature of features that the robot will encounter, the features are generic and can be appropriate for application to the general scene correspondence problem. In seminal work, Murase and Nayar [Murase 97] attempted to represent objects in terms of a 'parametric Eigen-space representation'. Their work has influenced the use of Eigen-spaces in topological mapping to represent places in the environment. Local-image features based on local image gradients are an important class of vision features. Baker, in [Baker 98], attempts to create a generalised descriptor for local image features and the introduction to his thesis provides a perspective on the development of gradient based methods. The stability and repeatability of points extracted at local Maxima (or Minima) in gradient images that have been repeatedly smoothed using operators, has been known for some time [Koenderink 84] [A.P.Witkin 83], and research in the field finally culminated in the Scale-Space theory proposed by Lindeberg [Lindeberg 94].

In certain cases and the fact that they need not convey the same amount of information, special methods might be designed to integrate local features with global ones to increase their distinctiveness [Lisin 05].

The uniqueness of the extracted features is ensured by building vector descriptors to represent each such feature. In work that combined the lessons of Scale-Space with the reliable characterisation of features, Lowe [Lowe 99] describes the use of gradient histograms taken at various points close to some point of interest. These features were called Scale Invariant Feature Transforms, SIFT. The work described in this thesis is based on local image features that are based on the SIFT features. The procedure for creation of the feature database has been modified to simplify the creation of features for image sequences.

A.2 Scale Invariant Feature Transforms, SIFT

Since their introduction, SIFT features have been widely applied, among others, to object recognition [Pope 00] [Lowe 01], in the panoramic assembly of images [Brown 03] and in image retrieval [Ke 04]. Various researchers have used this descriptor in new applications and modifications on the original procedure have appeared (see Weighted Gradient Orientation Histograms [Bradley 05], Modified SIFT [Andreasson 04], PCA SIFT and Global SIFT).

Two factors affect the efficacy of the SIFT descriptors (for that matter, of any other descriptor). These are 1)the repeatability of the point extractor and 2)the robustness of the descriptor itself to changes in the viewpoint, orientation and changes in lighting, scale, etc. Once the points are extracted from the images using a good corner extractor (see [Schmid 98] for an evaluation of different extractors), SIFT keys are employed to create a long and robust descriptor of the local point.

While the strength of the SIFT features stems from the long descriptors, this same property presents new challenges since each image easily throws up hundreds of good features and calculating the distance between the vectors of each of these features with the vectors of the features obtained from any other image is computationally very demanding. For this reason Lowe [Lowe 04], suggested the utilisation of a data structure, the KDTree, whose principal advantage lies in its ability to quickly retrieve points represented in very large dimensions. KDTrees can be constructed quickly (see [Kennel 04] for an fast open-source implementation) and make the matching of SIFT features a feasible task.

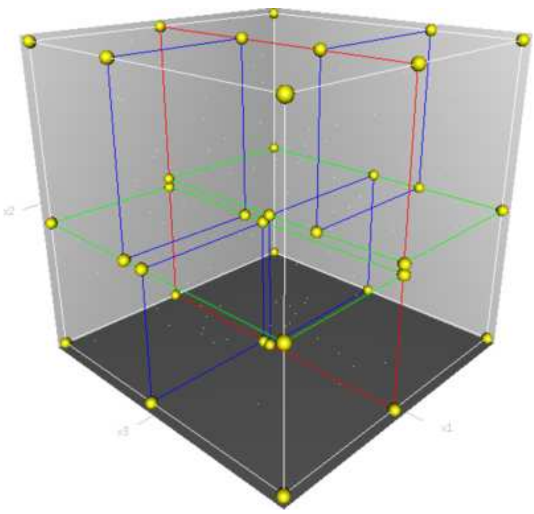


Figure A.1: A KD-Tree created from a set of 3-Dimensional point. Image appears at http://en.wikipedia.org/wiki/Kd-tree

A.3 Modifications to improve SIFT in Image Sequences

SIFT features have been used to characterize images in this work. In experiments presented in this thesis, typically between fifty and two hundred features are extracted per image from sequences containing up to three hundred and fifty VGA-size images.

The bag-of-features against which an image is compared is extracted by presenting a sequence of images sequentially to the feature extractor. If the sequence of images is presented without previous selection it results in the creation of many features with similar SIFT keys which, in certain situations can prevent us from correctly building a KDTree. Since the KDTree must be built all at once (no accepted method exists for incrementally adding data to a KDTree) the following simple procedure was adopted to aid the construction of the KDTree. This procedure requires no major modification of the original procedure for the creation of a KDTree and is described in the next subsection.

A sequence of sixty seven images of a laboratory wall was taken while the camera was displaced along a trajectory parallel to the wall, see Figs 2.5a through 2.5d. As expected, there are features that simultaneously appear over a large number of views. The resulting Feature Incidence matrix in Fig. 2.5e contains approximately three thousand points of which around one thousand five hundred are unique and are represented in the KDTree. Note that the arrangement of the feature across the diagonals of this and other FIMs is simply an artifact that appears because of the way the features are ordered (the features are added all at once followed by the elimination of repeated features).

KDTrees are extremely effective structures that aid the recovery of features that are characterized as vectors [Bentley 75]. In the context of their use in place recognition they have a serious limitation in that they must be constructed all at once. Separate KDTrees to recover (and match) HU features extracted from the laser range scan and SIFT features from the images were employed. To enable the creation of the KDTree for sequences of laser ranges scans and for sequence of images the original procedure has been modified by adding a ’noise-addition’ step before creating the tree and a ’Querying’ step which removes searches for similar vectors after the creation of the tree and marks copies of vectors them for deletion.

The novel procedure outlined in Algorithm 3 allows the trouble free construction of the tree and does not alter the way in which the SIFT features are retrieved and used. The added noise was small (less than 0.5%) and no significant degradation in the performance of the tree at the time of retrieval of the points was verified.

Algorithm 3 Create KDTree

$N =$ total number of points to insert
$n =$ number of points already inserted
$F =$ number of points to insert at a time (per image possibly)
Require: $N \geq 2$
while $(N - n > 0)$ **do**
Add tiny amount (less than 0.5%) of random noise to F points
Query tree for F points to be inserted (without noise)
Mark matched points in total of $n + F$ points for removal
Add marked points as aliases of points still in tree.
Destroy KDTree
Create new KDTree only with unique points
end while
Destroy KDTree
Create new KDTree only with unique points and without noise

A.4 Other Properties of the SIFT implementation

To account for the loss in the resolution as larger filters are applied to the images during the corner extraction, the original image is usually decimated repeatedly to reduce the computational requirements as the pixel-redundancy increases. When image correspondence has been applied to object recognition, the decimations, which result in the image pyramids, have been known to still yield useful results when performed many times (deep pyramids). SIFT keys are extracted for the whole image and for an image with a resolution reduced by half. This reduced number of decimations results in a capability to process images at a higher frequency without a discernible reduction in performance.

One of the most costliest steps in the creation of the SIFT descriptors is the identification of maxima/minima in the scale-space. Points that satisfy such a maxima/minima criteria, must be local maxima-minima and also be [scale-adjusted] maxima-minima along the dimension of scale dimension. In order to optimise the selection of local maxima-minima at the same scale, optimized functions to detect corners that available in the computer vision library OpenCV [Intel 7], were employed.

Appendix B

Extraction of 2D Laser Features

The reduction in the price of 2-D laser ranging devices, their superior accuracy, longer range and high measurement rates vis a vis other range finding devices, has resulted in their ubiquitous presence on mobile robots. In their contribution to various tasks ranging from mapping to path planning, they have all but replaced ultrasound sensors. Ultrasound range sensors have retained their place on robot platforms for the purpose of obstacle detection.

Range based methods have been frequently used to index place in indoor environments, first with ultra-sound sensors and later with laser range scanners. Methods that depend on range sensors have used maps with landmark and free-space boundary depictions to represent places. Range data is matched with one or more places in the map to perform place recognition.

In many applications, range finders are used either as the principal sensor or in conjunction either with vision, ultrasound, odometry or a combination thereof. From an application point of view, Laser range sensor data might be employed for characterisation the open-area at a particular place, for detection of the presence of absence of some (any) object at a particular position, for an estimation of motion by calculation of the 2D transformation that explains the difference between two consecutive scans. In this chapter the laser scan features used in this work are compared with use of laser range sensors in related literature.

Feature extraction from scan data attempts to segment points that reflect properties, of a portion of the scan or the scan as a whole, that are [relatively] invariant to changes in the point at which the scan is taken, the presence of view-obstructing objects and changes that occur in dynamic, real-world environments. Segmentation of the data into primitives or clusters based on their proximity or some parameters is, by far, the most commonly used approach to launch this process.

Multiple types of features from the laser range scan have been employed, namely 1)wall-like (line) features, 2)scan region properties and 3)scan contour properties in the form of a vector that characterises 2D discontinuities in the plane of the scan using Hu moments [Gonzalez 02].

B.1 Extraction of Wall-like features

Long lines from the points in the laser scan have been extracted using a two-stage method. The extraction of short line segments (containing between 2-6 points) was performed using the incremental method [Nguyen 05] followed by fusion into longer lines (segments of at least 2 meters). Binary features are created by classifying the number of extracted lines and their distance from the Laser range scanner.

In a first stage the complete laser range scan is modelled in terms of small line segments (of not more than say 5 points). These segments are constructed using the iterative-end-point-fit algorithm, see [Nguyen 05] for a comparison of results with other methods, that seeks to 'grow' a line by adding the next point (its theta neighbour in radial coordinates) and checking whether this addition is appropriate. If the new point satisfies the equation of the exiting line the line segment is extended, else a new line is initiated. To allow the lines to better describe the scan contour, the growth of the lines has been arrested at this stage.

In the second stage, the short line segments previously extracted are fused together into longer line segments. Fusion takes place as long as the line's inner points are not separated by more than a certain threshold distance and as long as the slopes of the lines are similar. As iterative procedure compares every pair of lines and checks for the possibility of fusing them. Once more, to create long line segments that best reflect the original laser scan, each line is allowed to fuse with at most one other in each iteration and the smaller lines are given preference over the bigger ones for fusion.

Algorithm 4 Create Small Segments using the Incremental End Point Algorithm

$N =$ total number of un segmented points in laser scan
$d =$ maximum separation distance between points (defined by distance between inner points)
$D =$ maximum allowable angle between lines.
$\#P =$ maximum number of points satisfying equation of a line segment
Require: $N \geq 1$
while $(N > 0)$ **do**
check if next point Add tiny amount (less than 0.5%) of random noise to F points
if next point lies at distance greater than d **then**
create new line
Current point is first point in new line
end if
if difference between angle between next point and last point in line and the line is greater than D **then**
create new line
Current point is first point in new line
end if
if number of points in line greater than $\#P$ **then**
create new line
Current point plus one is first point in new line
end if
Include point in existing line, re-calculate line parameters
end while
Destroy KDTree
Create new KDTree only with unique points and without noise

Figure B.1: Two failed attempts at fusing pair of lines are shown. The middle segment cannot be paired with the segment at right because the slopes are quite different (lines within the grey region are eligible for fusion). The middle segment cannot be paired with the segment at left because the two are separated by a large distance.

Table B.1: Categorisation of long lines into binary features

Feature Number	Description
0	>= 1 lines at 4+ meters
1	>= 2 lines at 4+ meters
2	>= 3 lines at 4+ meters
3	>= 1 lines at 2+ & 4- meters
4	>= 2 lines at 2+ & 4- meters
5	>= 3 lines at 2+ & 4- meters
6	>= 1 lines at 2- meters
7	>= 2 lines at 2- meters
8	>= 3 lines at 2- meters

Walls have been characterized in the image using long (of length greater that 2m) line segments. Various schemes exist to segment laser range scan data into line segments. Laser range scans already have some structure in the points within the scan - They are best described using radial coordinates. This structure has resulted in certain schemes that are more applicable to such data than schemes that seek to detect lines in say noisy images. These schemes vary on the criteria of what makes one line better than another line that competes for a point on the first line. Such criteria include but are not necessarily limited by

a) Average Point-to-Line distance

b) Distance between segments

c) Minimum number of Segments

d) Set of segments with a Minimum/maximum total length of Segment

Wall-like segments are important because of their robustness to changes in the point of view. They are also instrumental in building up other types of features as will be shown in a later section.

The (long)line segments themselves are then converted into binary features utilising a classification scheme describes in Table B.1.

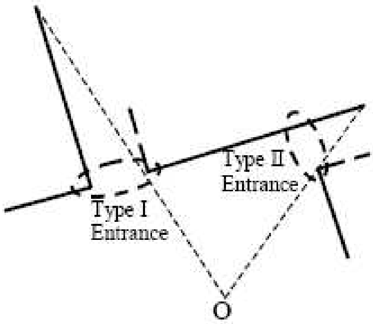
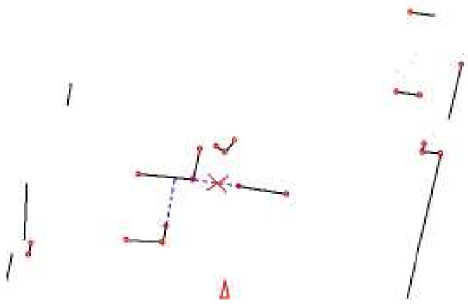


Figure B.2: Two types of doors that are detected in data from a LRF using the method developed in [Xiang 04].

(a) Simulated environment with one 'navigable' passage on the left and another 'blocked' passage straight ahead.



(b) Plan view of extracted segments representing walls (dark, continuous lines) and doors (lighter, broken lines).

Figure B.3: An example of detection of passable and non-passable doors in a typical laboratory scene.

B.2 Extraction of Doors or Door-like Features

These features were chosen primarily because of the availability of previously developed work that detected doors in laser range scans [Xiang 04]. Although (permanently open) doors are not very common in the environments in which the experiments were conducted in, the algorithm has a low probability of false detection and the presence of detected doors becomes important to characterise certain places.

In that work the authors apply empirical rules to extract open doors based on the position of adjacent walls and the geometric layout of these walls and the opening, Fig. B.2.

To obtain binary features the detected features have been converted by classifying them based

Table B.2: Classification of Open doors into binary features

Feature Number	Description
0	>= 1 doors at 4+ meters
1	>= 2 doors at 4+ meters
2	>= 1 doors at 2+ & 4- meters
3	>= 2 doors at 2+ & 4- meters
4	>= 2 doors at 2- meters
5	>= 2 doors at 2- meters

Table B.3: Classification Scan Boundary Features into binary features.

Feature Num	Description
1	area >= 2 m^2
2	area >= 4 m^2
3	area >= 8 m^2
4	MaxDim >= 4 m
5	MaxDim >= 8 m
6	MaxDim >= 16 m
7	MaxDim/MinDim >= 1
8	MaxDim/MinDim >= 4
9	MaxDim/MinDim >= 8

on the distance of the sensor to the center of the door or passage according to the Table B.2. A door is said to be detected if the probability of its existence is superior to 75%.

The results of the application of the door detection are shown in Fig. B.3.

B.3 Scan Boundary Features

Features describing the regional properties of the laser scan have been applied. The values of the area covered by the scan and the lengths of the principal dimensions are then classified as in Table B.3.

One type of feature is created with the aim of characterizing the free or open space covered by the laser range scan. The free area is classified using the thresholds in Table B.3 and the appropriate binary features are created.

Another type of feature provides a basic measure of the distribution of the free-space measured by the range scan. The layout of this free space is measured in terms of the longest and shortest dimension, $MaxDim$ and $MinDim$ respectively. These are then classified using the thresholds shown in rows Table B.3.

B.4 Hu Moment Features

Hu moments [Hu 62] are commonly employed in the matching of images and their use is based on the observation that combinations of the centralised moments of an image are (quite) invariant to rotation, scale and reflection [Gonzalez 02]. In this work, Hu image moments are used in a similar way to SIFT features, by collecting the 7-element features into a KDTree and matching the features extracted from new images (see next subsection).

The laser range scan for four scans from a sequence of 118 scans is shown in Fig. 2.4. The Feature Incidence Matrix for the sequence is also shown.

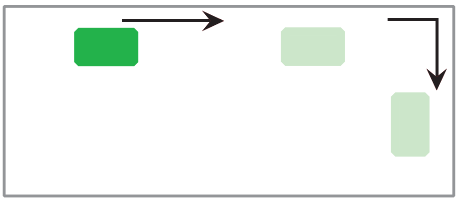
Appendix C

Description of a Robot Mission

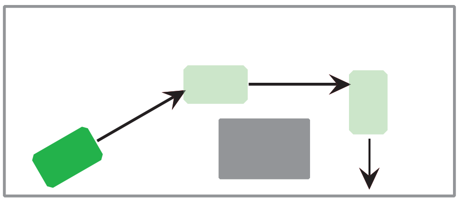
The robot is endowed with capabilities of moving in the environment according to certain laws that depend of the immediate environment as sensed by its complement of sensors.

The type of behaviours that can be implemented depend on the configuration of the robot, the type of sensors that the robot possesses and the algorithms that are available. Thus, a robot with steering capability and a range finder should be able to perform wall following or corridor following as suggested in Fig. C.1a.. The simplest behaviour that the robot can perform is way point following without closing the loop Fig. C.1b.

Upon reaching a ’Place’ the current motion behavior is stopped and a new behavior must be initiated. The Places in the environment are defined as locations at which the behavior of the robot must change or be modified so that the robot can take a new path or due to restrictions imposed by the sensory system of the robot.



(a) Illustration of Wall-Following behaviour.



(b) Way-point-Following behaviour.

Figure C.1: The two behaviours that have been used in the course of this work.

C.1 The Set of Robot Motion behaviors

The Way-Point-Generation module of the Tactical layer, described in section 5.2.2, provides a set of motion behaviors that propel the robot along. These consist of algorithms that are closely coupled with raw sensor data and properties of the local geometric map to discover trajectories with desirable properties through hill-climbing behavior and maintenance of these trajectories. The topological map that is generated during the Environment-Familiarization phase provides the topological and local metric information together with motion information that allow the robot to move from one node to another using a hill-climbing behavior on the function of sensor information.

From the set of possible behaviors stated above, the type of Place and the layout of the environment do not allow the execution of certain behaviors. The aim of the Robot Mission Control module is to recover the same behaviours used to get to the next place in the Reference Sequence during the final Navigation stage. To generate this Reference Sequence, the Views are created as the robot is led around the environment.

The motion behaviors that the robot is equipped to implement are

a) To move to some way point according to some control law (with or without feedback): The Move-Ahead behavior utilizes this trajectory generating method to negotiate a path in the environment.

b) To turn about itself $90°$, $180°$ or $270°$: The Turn-About behavior makes the robot turn about itself while keeping it at the same position. Such a behavior is possible wherever the local metric map allows it.

c) To move parallel to a wall.

d) To cross an open door or narrow opening: This behaviour is as abstraction of other behaviours in which the robot identifies the neared doorway, positions itself so as to be able to safely cross the doorway and finally negotiates the narrow crossing.

In certain situations specifying a particular type of abstracted motion behavior might be more intuitive and useful. For example while sending the robot through an unexplored part of the environment such as getting it to cross a door that was not previously crossed. Therefore a second type of symbol might be included in the string, attached to the symbol of the Key point, specifying the motion behavior of the robot. These Motion behaviors are always associated with the Key-points that they are attached to. The complete set of of motion behaviours is shown below.

$$\{M \in MA, MT, MC\} \begin{cases} MA \text{ specifies a trajectory set-point following behavior} \\ MT \text{ specifies a rotation of the robot about itself} \\ MW \text{ specifies wall-following behaviour} \\ MC \text{ specifies crossing of a door} \end{cases} \tag{C.1}$$

C.2 Mission Specification using Strings

Mission string: is a sequence of Important *Views* taken along a path, each of which is associated with an action. The Mission string is created as a 'string' of observations that the robot is expected to encounter as the Mission progresses.

In general, a human operator who guides a robot could be influenced by 1)the presence of sensory cues and 2)from clues about the affordability at the particular place.

Examples of the former include navigation in a corridor-like environment indoor, following a street or highway to a particular destination. Examples of the latter might include behaviours at Dead-ends, exiting open spaces and halls, crossing a door into another space etc. The key difference between the two types is that it would be difficult to describe such locations using a single image and hence some semantic property is attached to the Key-point. In this work the influence of environment affordability to control the execution of a Mission has not been attempted.

The labels that a human operator can assign to a place in an environment that are offered for use in Mission specification by the robot will belong to the set specified below.

$$\{L \in LD, LI_0 \dots LI_N, LO_0 \dots LO_M\} \begin{cases} LD \text{ represents a door in the environment} \\ LI_j \text{ represents an Image } j \text{ in the environment} \\ LO_k \text{ represents an Object } k \text{ in the environment} \\ LA_t \text{ represents a location } t \text{ with an abstract property} \end{cases} \quad \text{(C.2)}$$

From the definition, visual cues can be both ambiguous as in the case of doors and robots might not necessarily possess the capability to identify these same cues and the affordability that the environment presents at any place. The robot will , therefore, maintain its own representation of these places.

A simple mission could be specified that could set the robot off from a starting point in a corridor toward the end of the corridor. The map of the environment is shown in the image and the possible mission that gets the robot to the end of the corridor could be written as in Fig. C.2.

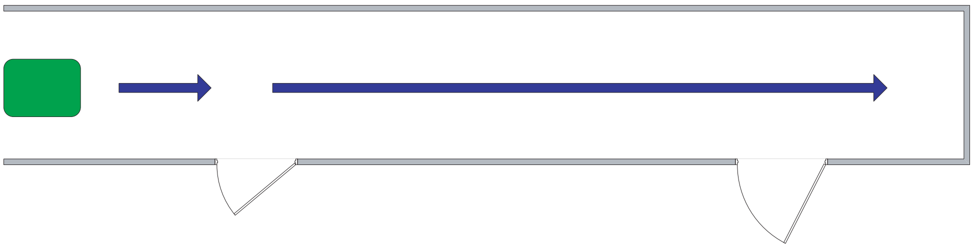


Figure C.2: A plan view of a simple environment. The Mission string could be written thus $Mission1 = [LD\_LA3]$

Thus, if the robot must cross the second door in the corridor, a door it has not crossed before, a mission could be written as $Mission2 = [LD\_LDMC]$, see Fig. C.3. The robot is expected to go past the first door and, when it reaches the second door, to go through the door.

Each of the Features that the robot can utilize in the construction of the Mission string can be identified using one or more of the robot sensors.

Two types of Landmarks are utilized, regular architectural features that represent landmarks such as doors that are plentiful but impossible to discriminate between, and unambiguous or unique Landmarks such an interesting scene or object that are (sufficiently) unique over the en-

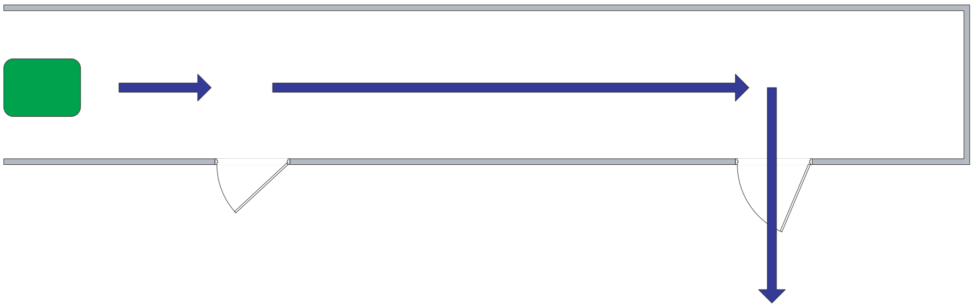


Figure C.3: A plan view of a simple environment. The to robot to continue past the first door and enter the second door.

vironment of operation of the robot. From a point of view of constructing a mission, choosing certain types of key-points over others should not be performed by the human operator, but the choice of key-points chosen might greatly affect the capability of the robot to localize itself in the topological map and deal with modified environments, occlusion of landmarks and varying light conditions. In the case of ambiguous landmarks the robot might not always represent environment correctly, resulting in an 'aliasing' error. Thus some landmarks might be missed while other might be detected. The consistency of a mission string must be maintained inspite of these difficulties. A solution to this problem has been attempted through the establishment of a one-to-one correspondence between the features in the mission string and the topological map at the time of mission generation. Also, ordering of the features that are presented for construction of the Mission string reduces the aliasing error.

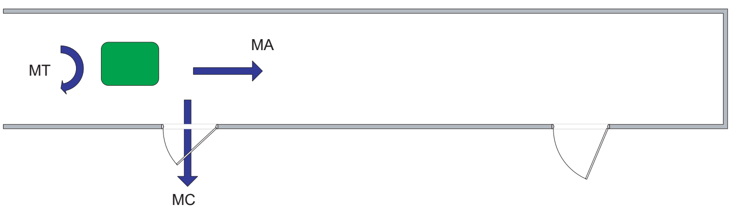


Figure C.4: Motions that can be performed at a Place defined by a door.

During the Environment-Familiarization phase, doors along the environment have been isolated using one or more sensors. The definition of doors includes open, closed and partially open doors. Thus an open door can be matched to a closed or a partially open door. The robot, however can obviously cross only open doors and the Way-Point Generation module arrests any attempt to cross closed or partially closed doors. Typically, during the Environment-Familiarization phase, the robot will approach the door and either go past or cross it. Unless the robot somehow returns to the same door during the Environment-Familiarization phase the motion behavior adopted the first time will the one used during the execution of the mission. If a Motion behavior is included in the Mission string for the door then it will be performed instead. The Motion behaviors that can be associated with Key-points of this type are $\{MN, MA, MT, MC\}$.

One of the capabilities of the robot is to extract local features from images and store this information for subsequent retrieval. Through the extraction and matching of local points of interest in an image, images are retrieved from a previously created database of images. The intensity image is also stored so as to allow the use of the position from which that scene was observed in the construction of the Mission string. Interesting scenes can be recognized using a camera or a set of stereo cameras. The Motion behaviors that can be associated with Key-points of this type are $\{MN, MA, MT\}$.

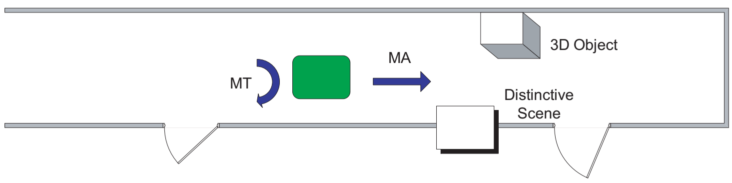


Figure C.5: Motions that can be performed upon “seeing” a distinctive scene or upon identifying a 3D object.

References

[Andreasson 04] Henrik Andreasson & Tom Duckett. *Topological Localization For Mobile Robots Using Omni-Directional Vision And Local Features*. In 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Instituto Superior Tenico, Lisboa, Portugal, July 2004. IFAC/EURON.

[A.P.Witkin 83] A.P.Witkin. *Scale-space Filtering*. In Proc. 8th Joint conference on Artificial Intelligence, pages 1019–1023, Karlsruhe, W. Germany, 1983. 8th Joint conference on Artificial Intelligence.

[Arkin 89] R. C. Arkin. *Motor schema-based robot navigation*. International Journal of Robotics Research, vol. 8, no. 4, pages 92–112, August 1989.

[Arras 99] K. Arras & N. Tomatis. *Improving Robustness and Precision in Mobile Robot Localization by Using Laser Range Finding and Monocular Vision*. In 3rd European Workshop on Advanced Mobile Robots, 1999.

[Arsénio 98] Artur Arsénio. Active laser range sensing for natural landmark based localisation of mobile robots. Master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa, July 1998.

[Aycard 04] Olivier Aycard, Jean-FranÃğois Mari & Richard Washington. *Learning to automatically detect Features for Mobile Robots using Second Order Markov Models*. International Hournal of Advanced Robotic Systems, vol. 1, no. 4, pages 231–244, November 2004.

[Baker 98] Simon Baker. *Design and Evaluation of Feature Detectors*. PhD thesis, Columbia University, 1998.

[Bellman 61] Richard Bellman. Adaptive control processes: A guided tour. Princeton University Press, Princeton, NJ, 1961.

[Bengtsson 98] Ola Bengtsson & Magnus Jonasson. Robot Localization in Changing Environments Using a Laser Range Scanner. Master's thesis, Halmstad University, Sweden, February 1998.

[Bentley 75] Jon Louis Bentley. *Multidimensional Binary Search Trees Used For Associative Searching*. Commun. ACM, vol. 18, no. 9, pages 509–517, 1975.

[Borenstein 94] J. Borenstein, B. Everett & L. Feng. *Where am I? Sensors and Methods for Mobile Robot Positioning*. Rapport technique, University of Michigan, 1994.

[Bradley 05] David M. Bradley, Rashmi Patel, Nicolas Vandapel & Scott M. Thayer. *Real-Time Image-Based Topological Localization in Large Outdoor Environments*. In Proceedings of the 2005 IEEE Conf. on Intelligent Robotics and Systems, August 2005.

[Brown 03] Matthew Brown & David G. Lowe. *Recognising Panoramas*. In Tenth International Conference on Computer Vision (ICCV 2003), October 2003.

[Carpin 07] Stefano Carpin, Andreas Birl & Viktoras Jucikas. *On Map Merging*. Robotics and Autonomous Systems, vol. 53, no. 1, pages 1–14, 2007.

[Castellanos 01] J. A. Castellanos, Juan Tardos & José Neira. *Multisensor fusion for simultaneous localization and map building*. IEEE Transactions on Robotics and Automation, vol. 17, no. 6, pages 908–914, December 2001.

[Chen 95] M.-Y. Chen, A. Kundu & S. N. Srihari. *Variable Duration Hidden Markov Model and Morphological Segmentation for Handwrittten Word Recognition*. IEEE Transactions on Image Processing, vol. 4, no. 12, pages 1675 – 1688, 1995.

[Choset 01] Howie Choset & Keiji Nagatani. *Topological Simultaneous Localization and Mapping (SLAM): Toward Exact Localization Without Explicit Localization*. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, vol. 17, no. 2, pages 125–137, 2001.

[Conant 04] Gavin C. Conant & Andreas Wagner. *A Fast Algorithm For Determining The Best Combination Of Local Alignments To A Query Sequence*. BMC Bioinformatics, vol. 5, pages 62–70, May 2004.

[Corso 03] Jason Corso, Darius Burschka & Gregory Hager. *Direct Plane Tracking in Stereo Images for Mobile Navigation*. In Proceedings of International Conference on Robotics and Automation, pages 875–880, 2003.

[Cover 91] Thomas Cover & Joy Thomas. Elements of Information Theory. Wiley Series in Telecommunications. Wiley InterScience, 1991.

[Cox 91] Ingemar Cox. *Blanche- An Experiment In Guidance And Navigation Of An Autonomous Robot Vehicle*. IEEE Transactions on Robotics and Automation,, vol. 7, no. 2, pages 193–204, April 1991.

[de Trazegnies Otero 04] Carmen de Trazegnies Otero. *3D Object Learning and Recognition System Based on Planar Views*. PhD thesis, Universidad de Málaga, Málaga, Spain, 2004. ISBN:84-689-5286-9.

[Dedeoglu 99] Goksel Dedeoglu, Maja Mataric & Gaurav Sukhatme. *Incremental on-line Topological Map Building with a mobile robot*. In Mobile Robots, volume 14, pages 129–139, 1999.

[Dietterich 02] Thomas G. Dietterich. *Machine Learning for Sequential Data: A Review*. Rapport technique, Oregon State University, 2002.

[Drumheller 87] Michael Drumheller. *Mobile Robot Localisation using Sonar*. IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 9, no. 2, pages 325–332, March 1987.

[Dudek 93] Gregory Dudek & Paul MacKenzie. *Model Based Map Construction for Mobile Robot Localization'*. In Proceedings of Vision Interface, pages 97–102, Toronto, Ontario, July 1993.

[Durbin 98] R. Durbin, S.R. Eddy, A. Krogh & G.J. Mitchison. Computational methods in molecular biology, chapitre 4, pages 45–63. Elsevier, 1998.

[Eddy 98] Sean R. Eddy. *Profile Hidden Markov Models*. BioInformatics Review, vol. 14, no. 9, pages 755–763, 1998.

[Elfes 87] A. Elfes. *Sonar-based real-world mapping and navigation*. IEEE Journal of Robotics and Automation, vol. 3, no. 3, pages 249– 265, June 1987.

[Ferreira 04a] Filipe Ferreira, Jorge Dias & Vitor Santos. *Convergence of methods for multi-scale Image analysis for Object and Image Recollection.* In 8TH Conference On Intelligent Autonomous Systems (IAS-8), 2004.

[Ferreira 04b] Filipe Ferreira, Vitor Santos & Jorge Dias. *Robot Navigation based on Landmarks detected in Multi-Scale range Images*. In ISORA 2004 10th International Symposium on Robotics and Applications, 2004.

[Ferreira 05] F. Ferreira, V. Santos & J. Dias. *A Low-level Framework for a Probabilistic Treatment of the Topological Description of a Robot Mission*. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems, IROS, Edmonton, Canada, August 2005. IEEE/RSJ.

[Ferreira 06] Filipe Ferreira, Vitor Santos & Jorge Dias. *Integration of Multiple Sensors Using Binary Features and a Bernoulli Mixture Model*. In IEEE Conference on Multisensor Fusion and Integration, Heidelberg, Germany, 3-6 September 2006.

[Ferreira 07a] F. Ferreira, V. Santos & J. Dias. *A Topological Path Layout for Autonomous Navigation of Multi-sensor Robots.* International Journal of Factory Automation, Robotics and Soft Computing, vol. 1, pages 203–215, 2007.

[Ferreira 07b] Filipe Ferreira, Vitor Santos & Jorge Dias. *Robust Place Recognition Within Multi-Sensor View Sequences Using Bernoulli Mixture Models*. In 6th IFAC Symposium on Intelligent Autonomous Vehicles, Toulouse, France, 2007.

[Filliat 03] David Filliat & Jean-Arcady Meyer. *Map-based navigation in mobile robots:I. A review of localization strategies*. Cognitive Systems Research, vol. 4, no. 4, pages 243–282, December 2003.

[Fleuret 04] Francois Fleuret. *Fast Binary Feature Selection with Conditional Mutual Information*. Journal of Machine Learning Research, vol. 5, page 15311555, 2004.

[Folkesson 04] John Folkesson & Henrik Christensen. *Graphical SLAM - A Self-Correcting Map*. In IEEE International Conference on Robotics and Automation, 2004, 2004.

[Forney 73] G. D. Forney. *The Viterbi Algorithm*. In Proceedings of the IEEE, volume 61, pages 268– 278, March 1973.

[Fox 99] Dieter Fox, Wolfram Burgard & Sebastian Thrun. *Markov Localization for Mobile Robots in Dynamic Environments*. Journal of Artificial Intelligence Research, vol. 11, pages 391–427, 1999.

[Fox 03] D. Fox, J. Hightower, H. Kauz, L. Liao & D. Patterson. *Bayesian Techniques for Location Estimation*. In Workshop on Location-aware Computing, part of UBICOMP Conf., Seattle, Washington, October 2003.

[García-Hernández 04] José García-Hernández, Vicent Alabau, Alfons Juan & Enrique Vidal. *Bernoulli mixture-based classification*. In A. R. Figueiras-Vidal et al., editeur, Proc. of the LEARNING04, ISBN 84-688-8453-7, Elche (Spain), October 2004.

[Gerstmayr 04] Lorenz Gerstmayr, Alexandre Bernardino & José Santos-Victor. *Appearance based Landmark Selection and Reliability Evaluation for*

Topological Navigation of an Aerial Vehicle. Rapport technique, Laboratorio de Visão, Instituto de Sistemas e Robótica, Instituto Superior Técnico, 2004.

[Gonzalez 01] J. Gonzalez, A. Juan, P. Dupont, E. Vidal & F. Casacuberta. *A Bernoulli Mixture Model for Word Categorisation*. In Symposium Nacional de Reconocimiento de Formas y Analises de Imagenes, 2001.

[Gonzalez 02] R. C. Gonzalez & R. E. Woods. Digital image processing. Addison-Wesley Pub. Co, 2002.

[Hernando 05] Diego Hernando, Valentino Crespi & George Cybenko. *Efficient Computation of the Hidden Markov Model Entropy for a Given Observation Sequence*. IEEE Transactions on Information Theory, vol. 51, page 7, 2005.

[Hines 90] William Hines & Douglas Montgomery. Probability and statistics in engineering and management science. John Wiley and Sons, 1990.

[Ho 05] Kin Leong Ho & Paul Newman. *Multiple Map Intersection Detection using Visual Appearance*. In Ciras 05, 2005.

[Howard 06] Andrew Howard, Gaurav S. Sukhatme & Maja J. Mataric. *Multirobot Simultaneous Localization and Mapping Using Manifold Representations*. Proceedings of the IEEE, vol. 94, no. 7, pages 1360 –1369, 2006.

[Hu 62] Ming-Kuei Hu. *Visual pattern recognition by moment invariants*. IEEE Transactions on Information Theory, vol. 8, no. 2, pages 179– 187, February 1962.

[Huang 05] W. Huang & K. Beevers. *Topological Map Merging*. International Journal of Robotics Research, vol. 24, no. 8, pages 601–613, 2005.

[Iivarinen 97] Jukka Iivarinen, Markus Peura, Jaakko Särelä & Ari Visay. *Comparison of Combined Shape Descriptors for Irregular Objects*. In British Machine Vision Conference, 1997.

[Intel 7] Intel. *Open Computer Vision Library*. `http://opencvlibrary.sourceforge.net`, 2007 (accessed March 3, 2007).

[Ishiguro 96] H. Ishiguro & S. Tsuji. *Image-based memory of environment*. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems, IROS, volume 2, Osaka, Japan, November 1996.

[Jensfelt 99] Patric Jensfelt & Henrik Christensen. *Laser Based Pose Tracking*. In IEEE International Conference on Robotics and Automation, Detroit, Michigan, USA, May 1999.

[Juan 04] Alfons Juan & Enrique Vidal. *Bernoulli Mixture Models for Binary Images*. In International Conference on Pattern Recognition (ICPR'04), volume 3, pages 367–370, 2004.

[Juang 85] B. Juang, L. Rabiner & S. Levinson M. Sondhi. *Recent developments in the application of hidden Markov models to speaker-independent isolated word recognition*. In IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '85, volume 10, pages 9– 12, 1985.

[Kaban 00] Ata Kaban & Mark Girolami. *Initialized and Guided EM-Clustering of Sparse Binary Data with Application to Text Based Documents*. In Proceedings of the 15th International Conference on Pattern Recognition (ICPR'00), volume 2, 2000.

[Ke 04] Yan Ke, Rahul Sukthankar & Larry Huston. *An Efficient Parts-Based Near-Duplicate And Sub-Image Retrieval System*. In MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia, pages 869–876, New York, NY, USA, 2004. ACM Press.

[Kennel 04] Matthew B Kennel. *KDTREE 2: Fortran 95 and C++ software to efficiently search for near neighbors in a multi-dimensional Euclidean space*, 2004.

[Koenderink 84] Jan J Koenderink. *The Structure Of Images*. Biological Cybernetics, vol. 50, no. 5, pages 363–370, 1984.

[Kohavi 97] Ron Kohavi & George H. John. *Wrappers for Feature Subset Selection*. Artificial Intelligence, vol. 97, no. 1-2, pages 273–324, August 1997.

[Konolige 04] Kurt Konolige, Dieter Fox, Charlie Ortiz, Andrew Agno, Michael Eriksen, Benson Limketkai, Jonathan Ko, Benoit Morisset, Dirk Schulz, Benjamin Stewart & Regis Vincent. *Centibots: Very Large Scale Distributed Robotic Teams*. In 9th International Symposium on Experimental Robotics, ISER 2004, 2004.

[Kortenkamp 94] David Kortenkamp & Terry Weymouth. *Topological mapping for mobile robots using a combination of sonar and vision sensing*. In Twelfth national conference on Artificial intelligence (vol. 2), pages 979 – 984, Seattle, Washington, United States, 1994.

[Kosecka 04] Jana Kosecka & Fayin Li. *Vision Based Topological Markov Localization*. In Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2004, 2004. SIFT Usage: As long as 4could be matched successfully between the consecutive views they were assigned as belonging to the same location.

[Kuipers 77] Benjamin Kuipers. *Representing Knowledge of Large-Scale Space*. PhD thesis, Mathematics Department, Massachusetts Institute of Technology, Cambridge, Massachusetts, June 1977. Also MIT-AI TR-418, 1977, Available via ftp://ftp.cs.utexas.edu/pub/qsim/papers/Kuipers-PhD-77.ps.gz.

[Kuipers 91] Benjamin Kuipers & Yung-Tai Byun. *A Robot Exploration And Mapping Strategy Based On A Semantic Hierarchy Of Spatial Representations*. Journal of Robotics and Automated Systems, vol. 8, pages 47–63, 1991.

[Kuipers 00] B. J. Kuipers. *The Spatial Semantic Hierarchy*. Artificial Intelligence, Elsevier, vol. 119, pages 191–233, 2000. http://www.cs.utexas.edu/users/qr/papers/Kuipers-aij-00.html.

[Kuipers 02] Benjamin Kuipers & Patrick Beeson. *Bootstrap Learning For Place Recognition*. In Eighteenth National Conference on Artificial Intelligence(AAAI-02), 2002.

[Kundu 98] Amlan Kundu, Yang He & Mou-Yen Chen. *Alternatives to Variable Duration HMM in Handwriting Recognition*. IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 20, no. 11, pages 1275–1280, November 1998.

[Lakaemper 05] Rolf Lakaemper, Longin Jan Latecki, Xinyu Sun & Diedrich Wolter. *Geometric Robot Mapping*. In International Conference on Discrete Geometry for Computer Imagery (DGCI), April 2005.

[Lamon 03] Pierre Lamon, Adriana Tapus, Etienne Glauser, Nicola Tomatis & Roland Siegwart. *Environmental Modeling with Fingerprint Sequences for Topological Global Localization*. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems, IROS, Las Vegas, October 2003.

[Lamon 06] Pierre Lamon, Sascha Kolski, Rudolph Triebel, Roland Siegwart & Wolfram Burgard. *The SmartTer for ELROB 2006 âĂŞ a Vehicle for Fully Autonomous Navigation and Mapping in Outdoor Environments*. Rapport technique, Autonomous Systems Lab, Ecole Polytechnique Federale de Lausanne, Switzerland, 2006.

[Lee 00] Sooyong Lee, Nancy M. Amato & James Fellers. *Localization based on Visibility Sectors using Range Sensors*. In Procceedings of the IEEE Int. Conference Robot. Autom. (ICRA)„ pages 3505–3511, Jan 2000.

[Levinson 86] S. Levinson. *Continuously variable duration hidden Markov models for speech analysis*. In IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP ’86, volume 11, pages 1241–1244, 1986.

[Limketkai 05] Benson Limketkai, Lin Liao & Dieter Fox. *Relational Object Maps for Mobile Robots*. In International Joint Conference on Artificial Intelligence (IJCAI), 2005.

[Lindeberg 94] Tony Lindeberg. Scale-Space Theory in Computer Vision. Kluwer Academic Press, 1994.

[Lisin 05] Dimitri A. Lisin, Marwan A. Mattar & Matthew B. Blaschko. *Combining Local and Global Image Features for Object Class Recognition*. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 3, pages 47–55, 2005.

[Lowe 99] David G. Lowe. *Object Recognition from Local Scale-Invariant Features*. In Proc. of the International Conference on Computer Vision, Corfu, pages 1150–1157, 1999.

[Lowe 01] David G. Lowe. *Local Feature View Clustering for 3D Object Recognition*. In Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pages 682–688, Kauai, Hawaii, 2001.

[Lowe 04] David G. Lowe. *Distinctive Image Features-From Scale-Invariant Keypoints*. IJCV, vol. 60, Issue 2, no. Issue 2, pages 91–119, November 2004.

[Manandhar 01] Dinesh Manandhar & Ryosuke Shibasaki. *Feature Extraction From Range Data*. In ACRS 2001 - 22nd Asian Conference on Remote Sensing, volume 2, pages 1113–1118, Singapore, November 2001.

[Marsland 01] Stephen Marsland, Ulrich Nehmzow & Tom Duckett. *Learning to select distinctive landmarks for mobile robot navigation*. Robotics and Autonomous Systems, vol. 37, pages 241–260, 2001.

[McLachlan 00] Geoffrey McLachlan & David Peel. Finite Mixture Models. John Wiley and Sons, 2000.

[Mitchell 95a] Carl Mitchell, Mary Harper & Leah Jamieson. *On The Complexity Of Explicit Duration Hmms*. IEEE Transactions on Speech and Audio Processing, vol. 3, no. 3, pages 213–217, 1995.

[Mitchell 95b] C.D. Mitchell, M.P. Harper & L.H. Jamieson. *Using Explicit Segmentation To Improve Hmm Phone Recognition*. In International Conference on Acoustics, Speech, and Signal Processing, ICASSP-95, volume 1, pages 229–232, 1995.

[Montemerlo 02] M. Montemerlo, S. Thrun, D. Koller & B. Wegbreit. *FastSLAM: A factored solution to the simultaneous localization and mapping problem*. In Proceedings of the AAAI National Conference on Artificial Intelligence, Edmonton, Canada, 2002. AAAI.

[Mourioux 04] G. Mourioux, C. Novales & G. Poisson. *A hierarchical architecture to control autonomous robots evolving in an unknown environment*. In IEEE International Conference on Industrial Technology, 2004. IEEE ICIT ’04. 2004, volume 1, pages 72– 77, 2004.

[Murase 97] Hiroshi Murase & Shree K. Nayar. *Detection of 3D objects in cluttered scenes using hierarchical eigenspace*. Pattern Recognition Letters, vol. 18, pages 375–384, 1997.

[Nadif 98] M. Nadif & G. Govaert. *Clustering For Binary Data And Mixture Models-Choice Of The Model*. Applied Stochastic Models And Data Analysis, vol. 13, pages 269–278, 1998.

[Newman 06] P. Newman, D. Cole & K. Ho. *Outdoor SLAM using Visual Appearance and Laser Ranging*. In IEEE International Conference on Robotics and Automation, 2006, 2006.

[Nguyen 05] Viet Nguyen, Agostino Martinelli, Nicola Tomatis & Roland Siegwart. *A Comparison of Line Extraction Algorithms using 2D Laser Rangefinder for Indoor Mobile Robotics*. In International Conference on Intelligent Robots and Systems, 2005.

[Nikovski 02] Daniel Nikovski & Illah Nourbakhsh. *Learning Probabilistic Models for State Tracking of Mobile Robots*. In IEEE/RSJ International Conference on Intelligent Robots and System, volume 1, pages 1026– 1031, 2002.

[Nock 03] Gabriel Nock. Segmentation algorithms for 2d-laserscans in an indoor environment. Master’s thesis, Universidad de Zaragoza, Dept. de Informática e Ingenieria de Sistemas, Grupo de Robótica y Tiempo Real, 2003.

[Ohba 97] Kohtaro Ohba & Katsushi Ikeuchi. *Detectability, Uniqueness, and Reliability of Eigen Windows for Stable Verification of Partially Occluded Objects*. IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 19, no. 9, pages 1043–1048, September 1997.

[Pearl 00] Judea Pearl & Stuart Russel. *Bayesian Networks*. Rapport technique, Computer Science Department, University of California, Los Angeles, 2000.

[Peng 05] Hanchuan Peng, Fuhui Long & Chris Ding. *Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy*. IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 27, no. 8, pages 1226 – 1238, August 2005.

[Pikrakis 06] A. Pikrakis, S. Theodoridis & Kamarotos. *Classification of musical patterns using variable duration hidden Markov models*. IEEE Transactions on Audio, Speech and Language Processing,, vol. 14, no. 5, pages 1795–1807, Sep 2006.

[Pope 00] Arthur Pope & David G. Lowe. *Probabilistic Models of Appearance for 3-D Object Recognition*. IJCV, vol. 40, no. 2, pages 149 – 167, 2000.

[Posner 06] I. Posner, D. Schroeter & P. Newman. *Using Scene Similarity for Place Labelling*. In 10th International Symposium on Experimental Robotics, ISER 2006, 2006.

[Rabiner 89] L. R. Rabiner. *A Tutorial on Hidden Markov Models and Selected applications in Speech Recognition*. Proceedings of the IEEE, vol. 77, pages 257–286, 1989.

[Rachlin 05] Yaron Rachlin, John M. Dolan & Pradeep Khosla. *Efficient Mapping Through Exploitation of Spatial Dependencies*. In 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.

[Reporter 00] Staff Reporter. *In praise of Bayes*. The Economist, vol. 2000, no. September 28, page 1, September 2000.

[Ribeiro 96] Maria Isabel Ribeiro & J Goncalves. *Natural Landmark Based Localisation of Mobile Robots Using Laser Range Data*. In IEEE EUROBOT 96, 1996.

[Sack 04] Daniel Sack & Wolfram Burgard. *A Comparison Of Methods For Line Extraction From Range Data*. In IAV 2004, 2004.

[Sajama 05] Sajama & Alon Orlitsky. *Supervised dimensionality reduction using mixture models*. In Proceedings of the 22 nd International Conference on Machine Learning, Bonn, Germany, 2005.

[Santos 01] Vítor M. F. Santos & Emanuel A. Oliveira. *Missões de Navegação para um Robot Móvel baseadas em Tarefas e Relações Topológicas do Ambiente*. Robótica, 2001.

[Savelli 05] Francesco Savelli. *Topological Mapping of Ambiguous Space: Combining Qualitative Biases and Metrical Information*. PhD thesis, Department of Computer and Systems Science, University of Rome, 2005.

[Scarlatos 93] Lori L. Scarlatos. *Spatial Data Representations for Rapid Visualization and Analysis*. PhD thesis, State University of New York at Stony Brook, 1993.

[Schmid 98] Cordelia Schmid, Roger Mohr & Christian Bauckhage. *Comparing and Evaluating Interest Points*. In ICCV, pages 230–235, Bombay India, 1998. Narosa Publishing House.

[Schmidt 06] Jochen Schmidt, Chee K. Wong & Wai K. Yeap. *A Split & Merge Approach to Metric-Topological Map-Building*. ICPR, vol. 3, pages 1069–1072, 2006.

[Schuster 99] Michael Schuster. *On Supervised Learning from Sequential Data with Applications for Speech Recognition*. PhD thesis, Nara Institute of Science and Technology, 1999.

[Se 97] Stephen Se & J Michael Brady. *Vision-based Detection of Kerbs and Steps*. In British Machine Vision Conference, BMVC, 1997.

[Sequeira 93] Vítor Sequeira & J. G. M. Gonçalves. *Position Estimation of a Mobile Robot Using Laser Range Data*. Rapport technique, JRC Technical Note No. I.93.132,, October 1993.

[Seshadri 94] N. Seshadri & C. Sundberg. *List Viterbi decoding Algorithms with Applications*. IEEE Transactions on Communications, vol. 42, no. 2/3/4, pages 313–323, 1994.

[Shannon 48] Claude Elwood Shannon. *A Mathematical Theory of Communication*. Bell System Technical Journal, vol. 27, pages 623–656, October 1948.

[Silver 04] David Silver, Dave Ferguson, Aaron Morris & Scott Thayer. *Feature Extraction for Topological Mine Maps*. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems, IROS, September 2004.

[Stewart 03] Benjamin Stewart, Jonathan Ko, Dieter Fox & Kurt Konolige. *The Revisiting Problem in Mobile Robot Map Building: A Hierarchical Bayesian Approach*. In Conference on Uncertainty in Artificial Intelligence, 2003.

[Tang 04] Fan Tang, Martin Adams, Javier Iba nez Guzmán & Wijerupage Sardha Wijesoma. *Pose Invariant, Robust Feature Extraction From Range Data With a Modified Scale Space Approach*. In IEEE International Conference on Robotics and Automation, 2004, 2004.

[Tapus 04a] Adriana Tapus, Guy Ramel, Luc Dobler & Roland Siegwart. *Topology Learning and Recognition using Bayesian Programming for Mobile Robot Navigation*. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems, IROS, 2004.

[Tapus 04b] Adriana Tapus, Nicola Tomatis & Roland Siegwart. *Topological Global Localization and Mapping with Fingerprints and Uncertainty*. In 9th International Symposium on Experimental Robotics, ISER, 2004.

[Tapus 05] Adriana Tapus & Roland Siegwart. *Incremental Robot Mapping with Fingerprints of Places*. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems, IROS, 2005.

[Tebo 97] Albert Tebo. *Sensor Fusion*. www.spie.org/web/oer/august/aug97/sensor.html, August 1997.

[Theocharous 04] Georgios Theocharous, Kevin Murphy & Leslie Pack Kaelbling. *Representing hierarchical POMDPs as DBNs for multi-scale robot localization*. In IEEE International Conference on Robotics and Automation, 2004, 2004.

[Thomas 00] Romain Thomas & Stéphane Donikian. *A Model Of Hierarchical Cognitive Map And Human Memory Designed For Reactive And Planned Navigation*. Rapport technique, l'Institut National de Recherche en Informatique et en Automatique, INRIA, Project siames, 2000.

[Thrun 98] Sebastian Thrun. *Learning Metric-Topological Maps for Indoor Mobile Robot Localization*. Artificial Intelligence, vol. 99, no. 1, pages 21–71, 1998.

[Thrun 02a] S. Thrun. *Robotic Mapping: A Survey*. In G. Lakemeyer & B. Nebel, editeurs, Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, 2002.

[Thrun 02b] Sebastian Thrun. *Particle Filters in Robotics*. In Proceedings of Uncertainty in AI (UAI), 2002.

[Thrun 05] Sebastian Thrun, Wolfram Burgard & Dieter Fox. Probabilistic robotics. MIT Press, 2005.

[Torralba 03] Antonio Torralba, Kevin P. Murphy, William T. Freeman & Mark A. Rubin. *Context-based vision system for place and object recognition*. In International Conference on Computer Vision, ICCV03, 2003.

[Tovar 03] Benjamin Tovar, Steven M. LaValle & Rafael Murrieta. *Locally-Optimal Navigation in Multiply-Connected Environments without Geometric Maps*. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems, IROS, 2003.

[Ulrich 00] Iwan Ulrich & Illah Nourbakhsh. *Appearance-Based Place Recognition for Topological Localization*. In International Conference on Robotics and Automation, pages 1023–1029, San Francisco, April 2000.

[Vale 05] Alberto Vale. *Mobile Robot Navigation in Outdoor Environments: A Topological Approach*. PhD thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2005.

[Vlassis 00] N. Vlassis, Y. Motomura & B. Kröse. *Supervised Linear Feature Extraction for Mobile Robot Localization*. In Proc. IEEE Int. Conf. on Robotics and Automation, pages 2979–2984, 2000.

[Vlontzos 92] J.A Vlontzos & S.Y. Kung. *Hidden Markov models for character recognition*. IEEE Transactions on Image Processing, vol. 1, no. 4, pages 539–543, Oct 1992.

[Wang 05] Xin Wang & Ata Kaban. *Finding Uninformative Features in Binary Data*. In Sixth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL ’05), 2005.

[Winters 02] Niall Winters & José Santos-Victor. *Information Sampling for vision-based robot navigation*. Robotics and Autonomous Systems, vol. 41, pages 145–159, 2002.

[Xiang 04] Zhiyu Xiang, Vitor M. F. Santos & Jilin Liu. *Locating And Crossing Doors And Narrow Passages For A Mobile Robot*. In First Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO), 2004.

[Zivkovic 07] Zoran Zivkovic, Olaf Booij & Ben Króse. *From images to rooms*. Robotics and Autonomous Systems, vol. 55, no. 5, pages 411–418, 2007.