Loop Closing Detection in SLAM using Scene Appearance

Kin Leong Ho
St Catherine's College

Robotics Research Group
Department of Engineering Science
University of Oxford

Michaelmas Term, 2007

This thesis is submitted to the Department of Engineering Science, University of Oxford, for the degree of Doctor of Philosophy. This thesis is entirely my own work, and, except where otherwise indicated, describes my own research.

Kin Leong Ho
St Catherines' College

Doctor of Philosophy
Michaelmas Term, 2007

Loop Closing Detection in SLAM Using Scene Appearance

Abstract

This thesis is concerned with the detection of loop closing in a Simultaneous Localisation and Mapping (SLAM) application. The loop closing detection problem asks how a robot can 'recognise' it has returned to a previously visited location after completing a long circuitous path. Many SLAM implementations look to internal map and pose estimates to make decisions about whether a robot has closed a loop. Approaches that rely on these estimates are generally unreliable in detecting loop closure when true and estimated robot poses diverge greatly.

The aim of this thesis is to produce a loop closing detection algorithm that is independent of pose estimates, sensor modality and estimation techniques, and works across a spectrum of workspaces. A key competency required is appearance-based place recognition. In order to achieve this goal, some significant issues pertinent to place recognition, namely perceptual variability and aliasing, have to be resolved. Viewpoint invariant descriptors are derived from observations used to represent local scenes. An efficient retrieval system coupled with indexing techniques allows for rapid comparison between observations based on a similarity function. Similarity relationships between local scenes are then encoded within a similarity matrix.

The loop closing problem is then addressed as a sequence detection problem within a similarity matrix. Exploiting the phenomenon that loop closing events occur as off-diagonals within a similarity matrix, a sequence detection algorithm is developed to extract such sequences. Instead of finding matching pair of observations, matching sequences are detected so as to exploit the topological relationships between scenes to reduce false positives. To further tackle the perceptual aliasing problem, spectral decomposition of a similarity matrix is carried out. The effects of repetitive and ambiguous artefacts found within an environment are removed through rank reduction based on an entropy maximisation criterion. Sequence detection is achieved in these rank reduced matrices. A principled manner to determine the probability of sequence occurring randomly allows the evaluation of the significance of such sequences before loop closing is triggered. The practical implementation of the loop closing technique is demonstrated in a variety of challenging scenarios and experimental settings.

Acknowledgements

First of all, I will like to express my thanks to my supervisor, Paul Newman, for his patience, guidance, encouragement and friendship. This thesis will not be possible if not for Paul's willingness to go the extra mile for his students.

Next, I will like to thank Andew Davison, Ian Reid and Andrew Zisserman for spending time to go through my work and guiding me along. Also, my thanks to Timor Kadir, Fred Schaffaltizky and Krystian Mikolajczk for getting my feet wet in the computer vision world.

My time in Oxford would have been much less interesting without the excellent company of Manjari Chandran (for her yummy vegetarian Indian cusine), Dave Cole (for his willingness to engage in hours of discussions) and Alastair Harrison (for his quirky puns). Much thanks to Ingmar Posner, Derik Schroether, Steve Reece and Iead Rezek for helping me to proof-read this thesis. Thanks as well to all other members of the Robotics Research Group, Charles, Teo, Matt, Neil, Phil, Chuan, Nick, David, Aeron, Jamie, Lyndsey, Ollie, Mark, Rebecca, Maria, Mukta, James, Josef and Rob, for all the pub trips, dinners and parties.

I will like to thank Mervyan Konjore, for putting up with me for two years when we were housemates. Honorable mention should also be given to Rashid, Sale (previously my naval academy roommate as well) and Nick Wu for staying with me at one point or another in my time in Oxford. Thanks to Kirk and Melissa for providing me with a London getaway when I need a breather from Oxford. Not to forget Ben and Renee for organizing those barbecues and get-togethers. Also thanks to Vijay Kowtha for his mentorship and the Office of Naval Research Global for its generous sponsorship.

I am grateful to the Rhodes Trust for providing me with an opportunity of a lifetime. In particular, I am grateful for the friendship and guidance of Mary Eaton, Catherine King, Sarah Freeman and Sheila Patridge. Thanks for all the balls and parties in the Rhodes House. For the exceptional latitude he has shown me, I thank the Warden of the Rhodes Trust, Sir Colin Lucas, for his trust and confidence.

Last but not least, I like to thank my family for always being there for me.

Contents

List of Notations

Mathematical Notations

Covariance Matrix	P
Control Vector	u
Descriptor	d
Descriptor Set	$D = \{d_1, \cdots, d_k\}$
Extractor Operator	$E(S) \rightarrow D$
H-matrix	H
Image	I
Image Sequence	$I^k = \{I_1, \cdots, I_k\}$
Landmark	x_f
Laser scan	L
Measurement	z
Observation	S
Segment	Seg
Sequence of observations	$S^k = \{S_1, \cdots, S_k\}$
Similarity Matrix	M
Similarity Function	$Sim(S_i, S_j)$
Rank Reduced Similarity Matrix	$\tilde{M}$
State Vector	$x(i\|j)$
Visual word	$\hat{\mathbf{d}}$
x-coordinate of robot	x_v
y-coordinate of robot	y_v
robot orientation	θ_v

Abbreviations

Consistent Pose Estimate	CPE
Cumulative Angular Function	CAF
Eigenvalue Decomposition	EVD
Extended Kalman Filter	EKF
Inverse Document Frequency	IDF
Lower Triangular Matrix	LTM
Maximally Stable Extremal Region	MSER
Scale Invariant Feature Transform	SIFT
Simultaneous Localisation and Mapping	SLAM
Spatial Similarity Matrix	SSM
Visual Similarity Matrix	VSM
Visually Ambiguous Region	VAR

Chapter 1

Introduction

1.1 Motivation

This thesis is concerned with "loop closure" detection in SLAM. Loop closing detection is the problem of ascertaining that a robot has returned to a previously visited location. SLAM is a technique used by a robot to perform online mapping of an *a-priori* unknown environment while using the same map to localise within its environment. As a core information engineering problem in mobile robotics, SLAM has received much attention in past years, especially with regard to its estimation theoretical aspect. However, lack of robustness continues to plague SLAM systems especially when it comes to loop closing. This is a particularly important component of the SLAM problem because accurate loop closing ensures previously visited locations are not remapped in incorrect global locations or topological order and enables accumulated errors in mapping and pose estimates to be corrected. Many SLAM implementations look to internal map and pose estimates to make decisions about whether a robot is revisiting a previously mapped area or exploring a new area. Loop closing approaches that rely on these estimates generally cannot detect loop closure when there is a great divergence between true and estimated robot poses.
An obvious case of poor loop closing is illustrated in Figure 1.1. Here, a robot has traversed a small loop of approximately 100 metres around a building in an anti-clockwise fashion. Robot poses are represented by triangles. Position uncertainties in robot poses are represented by grey

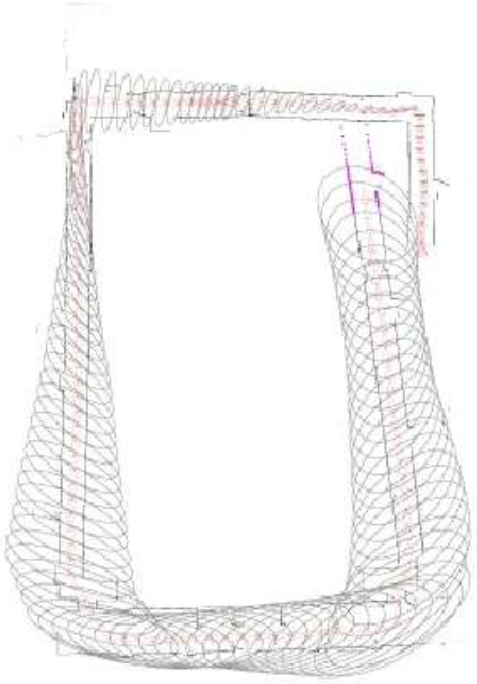


Figure 1.1: A snapshot of a SLAM algorithm just before loop closing takes place. The robot poses are depicted as triangles. Position uncertainty (gray ellipse) increases as length of the excursion from the start location increases. When the robot returns to a previously visited location, the true location (top right) lies outside the three-sigma boundary of pose uncertainty.

ellipses. As the robot traverses along the loop, uncertainty in robot pose increases as the robot is unable to re-observe any previously mapped environment. Linearisation and perception errors have led to a gross error in robot pose estimate and distortions in the mapping representation. When the robot returns to a previously visited location, the true location lies outside the three-sigma boundary of pose uncertainty.

The problem here is that the likelihood used to detect loop closing is not independent of pose estimates. Searching only in the neighbourhood of the robot is not robust in the face of gross pose error, which may be caused by small local errors. A gross pose error does not necessarily imply that a gross error of judgement was made. A small heading error over long linear traverses quickly leads to substantial position errors. The inability to detect loop closing has proved to be a stumbling

block in real-time SLAM systems from succeeding in mapping autonomously a large environment.

1.2 Summary of Approach

One of the aims of this thesis is to produce a loop closing detection algorithm that is independent of internal pose estimates. After all, the challenge of loop closing detection stems from the fact these estimates are often in error. A key competency of the loop closing technique is appearance-based place recognition. A robot associates each location of the environment with a local perception or observation (eg. geometric shapes or photometric information). A robot can generally recognise it has returned to a previously visited local scene by matching the current observation with previously stored observations without knowledge of internal pose. There are various challenges associated with using the approach of recognition. The first challenge is to derive a characteristic set of descriptors that can describe a local scene aptly. In many contemporary landmark-based SLAM algorithms (which will be discussed in Section 2.2), simple geometric primitives such as corners or lines are used as landmarks. These landmarks, by themselves, are not particularly discriminative. They are generally distinguishable only by their global or relative locations.
In contrast, this work employs a rich, discriminative set of descriptors to describe each local scene. The assumption is that each local scene is inherently unique and can be distinguished from one another based on its appearance. Nevertheless, the appearance of a local scene may be altered due to changes in perspective; this is known as the perceptual variability problem. This is a significant problem since a robot is unlikely to return to the same exact pose when closing a loop. A robot may re-observe a local scene from a different perspective and may not recognise it is the same location. This problem can be mitigated by the careful use of viewpoint invariant descriptors to describe a local scene. The goal is to extract sets of viewpoint invariant descriptors that can collectively be used to describe the appearance of a scene.
The environment explored by a robot is represented by a sequence of observations. Every observation is compared against one another in terms of similarity. Efficient scene similarity comparison is achieved through a scene retrieval system. The general concept of how a retrieval

system works is as follows: Every local scene visited by a robot is represented in a database as a set of descriptors, without any reference to pose estimates. Each newly captured observation will be compared against every other observations stored in the database with regard to their similarity. Indexing techniques are used to speed up the similarity comparison between local scenes. The pairwise similarity relationships between all local scenes are encoded into a similarity matrix. When a robot returns to a previously visited location after a circuitous route, newly captured observations will begin to match previously stored observations. As such, either a single matching pair of local scenes or a matching pair of *sequences* of local scenes will occur as a robot retraces its previous route.

However, loop closing detection is still complicated by the challenge of perceptual aliasing, where many different locations may appear similar. This challenges the assumption that each local scene has a unique appearance or that the unique characteristics of a local scene can be extracted within an observation to enable the scene to be distinguished. This is especially true in man-made environments where repetitive patterns and structures appear frequently. A pair of local scenes that are similar to each other may not necessarily indicate an actual loop closure. This thesis tackles this problem on two fronts: One approach is to exploit topological links between local scenes within a similarity matrix to reduce false positives. Sequences of matching pairs of local scenes, instead of a single matching pair of local scenes, are detected within a similarity matrix via a sequence detection algorithm.

Another approach for dealing with the challenge of perceptual aliasing is to remove the effects of ambiguous artefacts found within an explored environment. Certain artefacts may be found ubiquitously within an environment. This results in similarities between multiple local scenes containing such ambiguous artefacts. The problem is further compounded when these mutually similar local scenes are topologically close together. Spectral decomposition is employed to extract these common similarity modes from a similarity matrix. General themes of an environment are useful for summarising an environment but are not as useful for the purpose of loop closing. Consequently, the effects of such ambiguous artefacts are discarded from consideration during the

loop closing detection process without the removal of any observations. This is achieved in a principled manner by rank reduction of a similarity matrix based on an entropy maximisation criterion. Sequence detection is applied onto these rank reduced matrices. A method for evaluating the statistical significance of such sequences detected is introduced.

1.3 Achievements

A summary of the contributions of this thesis is as follows:

- Overall, a loop closing algorithm that is independent of pose estimates, sensor modality and estimation techniques has been formulated. The loop closing decision is based on scene appearance comparison. The challenges of perceptual variability and aliasing associated with an appearance-based approach are tackled. The notion of a similarity matrix that encodes pairwise similarity relationships between local scenes is introduced and the inherent properties of a similarity matrix are exploited to detect loop closing events.

- A sequence detection algorithm is introduced to exploits topological links between local scenes. Loop closing detection, in this context, is presented as the detection of a matching pair of local scenes or a matching pair of sequences of local scenes within a similarity matrix. A method for evaluating the statistical significance of such sequences is introduced.

- The role of multimodal sensing in tackling the perceptual aliasing problem in loop closing is briefly investigated. Loop closure detection was achieved in a visual similarity matrix and a spatial similarity matrix. Robust loop closure detection within a combined similarity matrix created from comparison of pairs of images and laser scans is achieved as well.

- The role of visual appearance in multi-robot mapping without recourse to pose estimates is demonstrated. A solution to a map joining application is achieved through the utilisation of the sequence detection algorithm in a similarity matrix created from comparison of sequences of observations collected from different robots. Overlaps or intersections between local maps

built by a team of robots are detected. From the alignment of these intersections, the local maps are combined into a single global map.

- An algorithm to explicitly handle and remove the effects of common mode similarity between multiple scenes throughout the workspace is developed. This approach is complementary to and work in tandem with the sequence detection algorithm.
- Extensive results are obtained by implementing the loop closing technique in a variety of environmental settings to demonstrate its applicability in real-life, challenging scenarios. Critical analysis of the performance of the loop closing algorithm is provided.

1.4 Thesis Outline

This thesis describes the progress towards solving the loop closing problem in SLAM associated with a scene appearance approach. The rest of the thesis is composed as follows:

Chapter 2 begins by describing the loop closing problem in a SLAM context. A literature review of several SLAM estimation-theoretical frameworks is given, emphasising the challenge that loop closing poses. The loop closing problem is considered as a two-part problem consisting of loop detection and map adjustment. The data association problem, which forms the basis of loop closing technique for many SLAM implementations, is reviewed. A review of appearance-based localisation is given. The loop closing problem is differentiated from the data association problem and appearance-based localisation. Other related works on existing loop closing approaches are reviewed.

Chapter 3 introduces the notion of how an environment can be represented as a set of local scenes. Each local scene is associated with an observation or a set of observations. The descriptor extraction process for both images and laser scan are reviewed. The similarity functions used to compare viewpoint invariant descriptors are also explained. Basic scene matching performance based on images and laser scans for varying perspective is demonstrated.

Chapter 4 describes the general concept of how a scene retrieval system can detect loop closing and enforce the constraints using an Extended Kalman Filter SLAM implementation. The latter part of the chapter investigates the problem of retrieval ambiguity; different descriptors have different discriminative power. A technique on how to assign weights to different descriptors based on their rarity within the database of descriptors is described. An improved retrieval system coupled with indexing technique and k-d tree structure is introduced.

Chapter 5 introduces the notion of a similarity matrix, which encodes pairwise similarity relationships between all local scenes. It also describes a method of extracting matching sequences of observations with highest similarity within a similarity matrix. The sequence detection algorithm is applied to visual, spatial and combined similarity matrices to investigate loop closing performance using multiple modal imaging sources. The sequence detection algorithm is also applied in a map joining application by detecting intersections between local maps built by different robots.

Chapter 6 is concerned with the management of ambiguity. It describes how spectral decomposition of a similarity matrix can help to remove the effects of ambiguous artefacts and, as such, improve the sequence detection performance. The effects of eigenvalue decomposition on synthetic similarity matrices are investigated. Experimental results from eigenvalue decomposition of different similarity matrices are demonstrated. A principal manner of rank reduction of a similarity matrix based on entropy maximisation is described. A method for assessing the significance of the sequence detected is described.

Chapter 7 presents results of implementing the loop closing technique. The performance of the loop closing algorithm has been tested in indoor, outdoor terrain and urban environments and an analysis of the performance for each scenario is given.

Chapter 8 concludes with a summary of the thesis. A summary of the contributions of this work is provided. Finally, a general discussion of ideas for future research is given.

Chapter 2

The Loop Closing Problem - Background and Literature

2.1 Introduction

This chapter introduces the loop closing problem in SLAM. The purpose of developing a loop closing technique is to improve the robustness of current SLAM systems. As such, the chapter begins by introducing the SLAM problem in Section 2.2. Various estimation-theoretical frameworks are reviewed and in particular, their inherent problems with detecting loop closing events are highlighted. Section 4.3 describes the loop closing problem as a two-part problem; loop detection and map adjustment. Data association techniques, which are commonly used to associate measurements with elements of a map, are discussed in Section 2.4 and compared with the loop closing problem. Section 2.5 describes appearance-based techniques used in topological localisation and relates it to the loop closing problem. Section 2.6 discusses other related work and current trends in loop closing. Section 2.7 concludes the chapter.

2.2 SLAM problem

The loop closing problem has generally been studied in conjunction with SLAM. To motivate the work in this thesis, the SLAM problem is briefly reviewed. SLAM is a technique used by a robot to

perform online mapping of a-priori unknown environment while using the same map to localise within its environment. Let $\mathbf{x}$ denote robot pose in $x-y-\theta$ space. A discrete time model of evolution of robot pose and measurements is adopted, $k = 1, 2, \cdots$. A robot pose at time t_k is expressed as $\mathbf{x}_k$. Without loss of generality, pose $\mathbf{x}_0$ is defined to be the origin of the coordinate system with a heading direction of 0 degrees such that $\mathbf{x}_0 = [0, 0, 0]^T$. Let $\mathbf{u}_k$ denote a control input applied at time t_{k-1} to drive a robot from $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$. The control input consists of a combination of rotational and translational motion. Let $\mathbf{z}_k$ denote a measurement at time t_k. Let $\mathbf{m}$ denote a map which is time invariant. Let the history of poses be $\mathbf{x}^k = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k\} = \{\mathbf{x}^{k-1}, \mathbf{x}_k\}$. Let the history of control inputs be $\mathbf{u}^k = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\} = \{\mathbf{u}^{k-1}, \mathbf{u}_k\}$ and the history of measurements be $\mathbf{z}^k = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\} = \{\mathbf{z}^{k-1}, \mathbf{z}_k\}$.

The SLAM problem can be abstracted as follows. As a robot explores an environment, it moves through a series of discrete poses $\mathbf{x}_1, \cdots, \mathbf{x}_k$. The movement of the robot from one pose to the other is modeled by a control input, $\mathbf{u}_k$. At each pose, a measurement of the environment is captured, resulting in a series of measurements $\mathbf{z}_1, \cdots, \mathbf{z}_k$. The end goal is to estimate the map, $\mathbf{m}$, and robot pose $\mathbf{x}_k$. In probabilistic form, the SLAM problem requires the the probability distribution $p(\mathbf{x}_k, \mathbf{m}|\mathbf{z}^k, \mathbf{u}^k, \mathbf{x}_0)$ to be calculated. A recursive estimator which is is central to virtually all SLAM algorithms is as follows:

$$p(\mathbf{x}_k, \mathbf{m}|\mathbf{z}^k, \mathbf{u}^k, \mathbf{x}_0) = \eta. \underbrace{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m})}_{measurement\ model} \int \underbrace{p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)}_{motion\ model} \underbrace{p(\mathbf{x}_{k-1}, \mathbf{m}|\mathbf{z}^{k-1}, \mathbf{u}^{k-1}, \mathbf{x}_0)}_{previous\ estimate} d\mathbf{x}_{k-1} \quad (2.1)$$

where the recursive estimator is a function of a motion model $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)$ and a measurement model $p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m})$. The probabilistic formulation of SLAM can be found in Appendix A.

2.2.1 Estimation Frameworks

Some common approaches to calculating the full conditional posterior on the L.H.S of Equation 2.2 shall be discussed now.

Landmark-based Extended Kalman Filer

An important formulation of the SLAM problem was introduced in the seminal paper by Smith *et al.* [115], which proposed using the Kalman filter to provide a recursive Bayesian estimator to the navigation problem. It concurrently provides an estimate of uncertainty of the robot pose and the landmark positions based on predictive model of the robot's motion and the relative measurement of landmarks. The state vector, $\mathbf{X}$, consisting of robot pose, $\mathbf{x}_v$, and n landmarks, $\mathbf{x}_{f_1}, \cdots, \mathbf{x}_{f_n}$, can be expressed as $\mathbf{X} = [\mathbf{x}_v; \mathbf{x}_{f_1}; \mathbf{x}_{f_2}; \cdots; \mathbf{x}_{f_n}]$. Associated with the state vector is a map covariance matrix, $\mathbf{P}_{(\mathbf{x})}$, where the off-diagonal sub-matrices encode the correlations between landmark location estimates and provide the mechanism for updating all the relational estimates. One limitation of the EKF approach is computational in nature. Computational complexity increases quadratically with n number of landmarks stored in the state vector. This limits the number of landmarks this approach can handle. This limitation has been recognised and several solutions have been proposed. One approach to deal with the complexity is through postponement [40]. Computational complexity is limited to only updating a local region of the global map while maintaining globally referenced coordinates. The overall map is updated in one full iteration at full computational cost at a later stage. Another approach is to decompose the problem of building a single large map into a collection of smaller maps [9, 74]. The smaller maps can be updated more efficiently. A second limitation of the Kalman formulation is related to its inherent unimodal character. The EKF can only maintain one hypothesis with its unimodal Gaussian distribution model. To resolve this, multiple Kalman filters were used to maintain multiple hypotheses [54] at the expense of computational complexity. Landmark-based EKF SLAM systems generally employ data association techniques (which will be discussed in Section 2.4) to detect loop closing events. The weaknesses of current techniques in handling loop closing when faced with cluttered landmarks and high uncertainties in pose and map estimates are described.

Delayed-State Extended Kalman Filter

In contrast to landmark-based EKF, the state vector in a delayed-state EKF includes previous robot poses, $\mathbf{X} = [\mathbf{x}_{v_k}; \cdots; \mathbf{x}_{v_{k-m}}; \mathbf{x}_{f_1}; \cdots; \mathbf{x}_{f_n}]$. Augmenting the state vector with previous poses enables delayed initialisation of landmark and fusion of measurements for each stored pose. Once sufficient information is available, a landmark can be initialised by a batch update. Deferred decision making also facilitate batch gate validation (described in Section 2.4) and make data association more reliable.

In certain applications, no landmarks are stored within the state vector and instead, one or more observations (a laser scan or an image) is associated with each pose [29, 97]. The state vector, $\mathbf{X}$, consisting of present robot pose, $\mathbf{x}_{v_k}$, and previous poses, $\mathbf{x}_{v_1}, \cdots, \mathbf{x}_{v_{k-1}}$, can be expressed as $\mathbf{X} = [\mathbf{x}_{v_k}; \mathbf{x}_{v_{k-1}}; \cdots; \mathbf{x}_{v_1}]$. This is also known as trajectory-oriented SLAM. At a suitable interval, the state vector is augmented with a new robot pose. The delayed-state concept has been adopted in a visual augmented navigation application [28, 29]. The state vector contains only a vehicle trajectory that is represented by a history of poses in an augmented state Kalman Filter and defines the 'map'. The EKF concurrently estimates online present vehicle position and its past trajectory. This formulation is particularly suited for environment where discrete identifiable landmarks are not easily discerned. However, the state space grows unbounded with time or distance, as well as the quantity of stored measurements, even if the robot is re-exploring a mapped environment.

FastSLAM

FastSLAM [92] decomposes the SLAM problem into a robot localisation problem and a collection of landmark estimation problems conditioned on robot pose estimate. Taking advantage of an insight that the posterior can be factored [93], the problem of determining N landmark locations is decoupled into N independent estimation problems when robot path is known. The path estimator is implemented using a Rao-Blackwellised particle filter while the landmark location estimator is implemented using EKFs. Each particle consists of an estimate of the present robot pose, s_k, all

the previous poses of the robot, s^{k-1}, and a set of N independent EKFs that estimate the n landmark locations conditioned on the path estimate. The particle set is resampled based on consideration of control input, $\mathbf{u}_k$ and measurement, $\mathbf{z}_k$. A tree-based data structure was developed to reduce the running time of FastSLAM to $O(mlog(n))$, where m is the number of particles and n is the number of landmark.

FastSLAM is capable of mapping larger environments than EKF SLAM due to less computational complexity. FastSLAM does not face the limitation of unimodal Gaussian distribution as EKF-SLAM. FastSLAM is capable of maintaining multiple data association hypotheses for landmarks simultaneously because each landmark location estimators can make different data association decisions and are independent of each other [91]. This is similar to multiple hypothesis tracking where multiple Kalman filters are maintained [54]. However, the data association technique employed by FastSLAM is nevertheless similar to other estimation techniques [99] and is equally susceptible to making false data association. Moreover, a substantial number of particles is required in order to model robot position estimate uncertainty adequately. FastSLAM may suffer from a depletion of particles when closing large loops [118] even though this problem may be mitigated with active path planning [119].

Consistent Pose Estimate Method

Consistent pose estimate (CPE) is a SLAM method that uses constraints between robot poses. For each pose, a set of constraints connects it to other poses. The constraints can be in the form of odometry between poses or matching scan between poses. Scan matching is the process of rotating and translating a range scan such that a maximal overlap with the priori map is achieved [83]. No explicit representation (landmarks) of the environment is required. The poses, $\mathbf{x}_1, \cdots, \mathbf{x}_k$, are the free variables in the system. CPE tries to globally optimise the set of poses based on how well neighbouring range scan matches.

Sequential poses and (and hence range scans) are registered to each other resulting in a trajectory of robot poses and a map, $\mathbf{m}$, made implicitly from the now aligned pure raw laser scans. The method is incremental and runs in constant time independent of the size of the map [62]. This

approach has so far been demonstrated to work with planar environments only. This method used the *Local Registration Global Correlation* technique to resolve the loop closing problem. Details on this technique are discussed in Subsection 2.3.2.

2.2.2 Map Types

Different types of mapping an environment are available; in metric form [73, 125], topological form [17, 68] or hybrid mapping form [127, 71]. Metric mapping represents an environment in terms of metric relationships with respect to one or more common frames of reference. Metric maps can be further divided into landmark-based maps [73] or maps that does not require an explicit representation of landmarks [43, 11], for example matching a set of scans associated with a series of poses. A topological approach represents a map as a graph where the 'nodes' correspond to "distinctive places" in an environment and 'edges' correspond to paths between places. Hybrid mapping combines both metric and topological mapping by representing local regions with metric maps while representing overall structure with a topological map [127, 71].

2.3 Loop Closing problem in SLAM

Loop closing is now deemed one of the main challenges in developing, a real-time, large-scale SLAM system [43, 82] and there has been much interest in tackling the loop closing problem [104, 85, 4]. A successful loop closing prevents re-mapping of the same location in a wrong topological order or metric location, and allows errors in a map to be corrected. In SLAM, loop closing can be considered as a two-part problem; the first step is correctly asserting that a robot has returned to a previously visited location and the next step is how to correct the errors of a map. However, it is not uncommon for each part of the problem to be dealt with separately. As such, the review of existing loop closing technique will be conducted in two parts. Firstly, a brief review of techniques that improve the efficiency and or accuracy of map correction upon detection of loop closure is given. Next, different approaches to detecting loop closing events are reviewed.

2.3.1 Map Adjustment

When the point of loop closure or a loop constraint is assumed known, the loop closing problem is addressed as the challenge of using all available information from the observation gathered so as to optimally adjust the whole map.

Extended Kalman Filter

The Extended Kalman Filter maintains the correlations between all landmarks which contain all the information about the whole loop and is used to propagate the correction throughout the map. However during a large loop, linearised methods such as the Extended Kalman Filter fails to obtain accurate map estimation due to large uncertainties. A method is proposed to approximate the correlations between all landmarks and avoid linearisation [85]. This method still enable the same corrections to be made but with a computational complexity that is independent of the number of landmarks. However, only simulated experimental results are available to demonstrate the approximated solution is comparable to standard EKF.

Optimisation

In order to tackle the non-linearities problem faced by the EKF, a different approach is proposed where the map adjustment problem is formulated as the problem of obtaining the maximum *a-posteriori* likelihood estimation of relative locations at the global level, given a loop constraint [27]. A non-linear constrained least-squares optimisation method based on the *Sequential Quadratic Progamming* method [131] is developed. Only the portion of the map that is associated with the loop is involved with the optimisation process, thus making this technique independent of total map size.

A Monte Carlo EM-based algorithm is used to tackle the loop closing problem in a 2D laser range-based SLAM [60]. The EM algorithm finds the most likely robot trajectory and implicitly closes the loop. The E-step is used to create a probability distribution over partitions of feature tracks while the M-step maximises the robot trajectory based on the virtual structure obtained in

the E-step. This approach has so far only been demonstrated to work for a planar environment using range based scan. EM approaches overcome the data association problem by performing hill-climbing search in the space of all maps in a way that constantly refine the estimated data association. However, it requires a good initial guess and does not guarantee a global minimum. Moreover, it is inherently a batch algorithm, requiring multiple passes through the entire data set. As such, it is not applicable to online mapping problems.

2.3.2 Loop Closing Detection

The focus of this work is on the detection of loop closing events. For the rest of this chapter, the focus will be on the different approaches on how the loop detection problem can be addressed.

Local Registration and Global Correlation

Previously in subsection 2.2.1, it is mentioned that an approach called "Local Registration and Global Correlation" is used in the consistent pose estimate method to determine topologically correct relations between new and old poses after long cycles [42]. As a whole, the local registration and global correlation method consists of three techniques, namely scan matching, consistent pose estimate and map correlation. Given a map is incrementally built from scan matching, it is important to establish correct topological relationships between scans as a mistake will cause the map to be misaligned. Once a topological connection is made, the effects are irreversible. Loop detection is activated with every new scan. To identify loop closure, a large "map patch" is correlated [63] (over motion in the plane) with a partition of the "old" map. A scan patch is a set of neighbouring range scans registered together. The intuition being that the larger scan patch will be more reliable than a single laser scan in rejecting false positives. However, the location of the search space is still restricted to an area around the robot pose estimate. The search space grows linearly with uncertainty in robot pose.

Global Localisation

The loop closing problem has been addressed as a global localisation or "kidnapped robot" problem, whereby the robot tries to determine its location within a map when it has actually closed a loop. The *linear time vehicle relocation* technique [95] uses the principle of global localisation to detect loop closure by checking if there is considerable overlap between present local map with other local maps [27]. If there is significant overlap with multiple local maps, loop closing decision is delayed until enough information can be gathered to resolve this ambiguity. Making use of a combination of geometric constraints that consider feature correlation, joint compatibility test (which will discussed in the next section), random sampling and locality, this algorithm is linear with both the size of the stochastic map and the number of measurements. The notion of locality limits search in the map to a subset of covisible features. The issue with treating the loop closing problem as a localisation problem will be discussed in greater detail in Section 2.5.

2.4 Data Association

The loop closing problem is essentially a correspondence problem just as the data association problem is. In a similar fashion, loop closing usually involves the process of associating current measurements with elements stored in a map albeit after a robot has traversed a long, circuitous route. Indeed, many practical SLAM applications use data association techniques as the default means of detecting loop closure. As such, a review of data association techniques is given in this section. Data association is the process of determining whether the existing map representation, $\mathbf{m}$, explains the current sensor measurement, $\mathbf{z}_k$. In the context of landmark-based SLAM, decisions must be made to determine if the measurement (1) originates from one of the landmarks in the map, (2) originates from a new landmark, or (3) is spurious. If spurious, the measurement is rejected. Otherwise, the map representation will have to be augmented using the measurement.

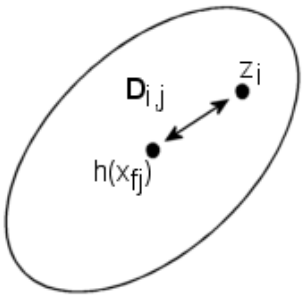


Figure 2.1: Validation gate: The points represent the predicted location and measured location. The ellipse represents the position uncertainty of the predicted observed location. The *Mahalanobis distance*, $D_{i,j}$, between the measurement $\mathbf{z}_i$ and the predicted observation $h(\mathbf{x}_{fj})$ for landmark $\mathbf{x}_{fj}$ must be less than $\gamma^2_{n,\alpha}$ to be considered as a candidate assignment.

2.4.1 Nearest Neighbour gating technique

This subsection describes the probabilistic data association method which deals with a single landmark in clutter [3]. A predicted observation, $h(x_{f_j})$, (eg. range and bearing) for landmark x_{f_j} from a robot is generated for each landmark. The predicted observation is compared with the actual measurement, $\mathbf{z}_i$, obtained, using a normalised squared innovation test as shown in Equation 2.2. The difference between predicted observation and actual measurement is known as innovation ν. The landmark with the closest predicted measurements within the gate defined by the Mahalanobis distance, $D_{i,j}$, is selected as the origin of the sensor measurement as illustrated by Figure 2.1. If the sensor measurement does not gate with any of the mapped landmarks, a new landmark is initialised. This is a popular data association approach adopted in many SLAM implementations [14, 30].

$$\nu^T S^{-1} \nu \leq \gamma_{n,\alpha} \tag{2.2}$$

where ν is the innovation, S is the innovation covariance, n is the dimensionality of sensor observation and α is desired confidence level.

However, the maximum likelihood approach is reliable only where clutter is low, sensor precision is

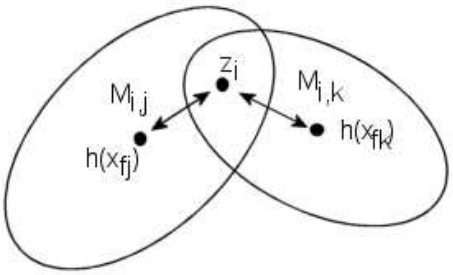


Figure 2.2: If a single observation falls within the validations gate of multiple targets, there exists uncertainty as to the correct assignment. The ellipses represent the position uncertainties of the predicted observed locations. For example, $\mathbf{z}_i$ can be either assigned to $\mathbf{x}_{fj}$ or $\mathbf{x}_{fk}$.

high and pose error is moderate. An example case of poor data association due to clutter and high pose uncertainty is shown in Figure 2.2. In the context of loop closing, the reliability of gated nearest neighbour approach drops drastically when the uncertainties of robot pose and landmark locations are high after a robot has traversed a large loop.

2.4.2 Batch Gating

A natural way to enhance robustness is to consider multiple measurements and multiple landmarks together. This ensures the resulting hypothesis contains jointly compatible pairings. The *joint compatibility test* [94] tackles this problem by measuring the joint compatibility of multiple matchings so as to remove spurious matching. It takes advantage of the fact that innovations in matchings between different observations obtained from the same robot pose are correlated. Consider a set of observations $Z = [z_1, \cdots, z_n]^T$ with covariance model R and a set of landmark estimates $\hat{X}_f = [\hat{x}_{f_1}, \cdots, \hat{x}_{f_m}]$ with covariance P. A tentative set of associations $E_k = [e_1, \cdots, e_j]$ is chosen subject to individual validation such that associated pair for e_i be denoted as z_{ei} and x_{ei}. The joint observation is given by $Z_{E_k} = [z_{e_1}, \cdots, z_{e_j}]$ with covariance R_{E_k}. The joint predicted observation is as follows:

$$\hat{Z}_{E_k} = h_{E_k}(\hat{x}) = \begin{bmatrix} h_{e_1}(\hat{x}) \\ \vdots \\ h_{e_j}(\hat{x}) \end{bmatrix} \tag{2.3}$$

The joint innovation and innovation covariance are then calculated as follows:

$$\nu_{E_k} = Z_{E_k} - \hat{Z}_{E_k} \tag{2.4}$$

$$S_{E_k} = \nabla h_x P \nabla h_x^T + R_{E_k} \tag{2.5}$$

where Jacobian $\nabla h_x = \dfrac{\partial h_{e_k}}{\partial x}$.
The joint validation gate is found to be:

$$\nu_{E_k}^T S_{E_k}^{-1} \nu_{E_k} \leq \gamma_{n,\alpha} \tag{2.6}$$

The probability that a spurious pairing is jointly compatible with all the number of pairings in the hypothesis is small. The joint compatibility test involves a search algorithm that traverses the tree in search of the hypothesis that has the largest number of jointly compatible pairings.
Similarly, *combined constraint data association*, which is a graph search technique, permits robust association in cluttered environments by constructing and searching a correspondence graph between landmarks [2]. It obtains a hypothesis with mutually compatible associations and is able to perform reliable association without knowledge of robot pose. As pointed out in [2], most landmark based SLAM implementations model landmarks as simple geometric primitives such as points and lines. This means the map is represented by virtually identical landmarks that are only distinguishable by their global or relative locations. This results in their dependence on pose estimate. As such, the data association techniques mentioned so far are unlikely to translate well from geometric landmarks to deal with unstructured environments.

2.4.3 Multiple Hypotheses

Multiple hypothesis data association handles association ambiguities by keeping a separate track estimate for each association hypothesis [3]. Over time, an ever-branching tree is created and this causes a substantial burden on computational resources. As such, low-likelihood tracks are pruned from the hypothesis tree. Multiple hypotheses data association can be considered a significant attribute of FastSLAM since each particle can be associated with different landmarks. A novel multiple hypothesis tracking implementation uses FastSLAM to create a new particle for each hypothesis [99]. Each particle is split into $n + 2$ particles, one for each of the n hypothesis, one for the non-association hypothesis and one for new landmark hypothesis. During the resampling phase, the wrong hypothesis will be eliminated when observations are incorporated. However, such mechanism only help to improve robustness of SLAM systems but does not improve data association decisions. The idea of maintaining multiple hypothesis extends to global localisation [54] and loop closing [127] as well.

2.4.4 Comparison of data association with loop closing

To conclude this section on data association, it is important to point out the nature of the loop detection problem as compared to data association. The loop closing problem can be looked upon as a special case of the data association problem in which the underlying challenges of pose uncertainty and uncertainties in map estimates are exacerbated. These conditions generally result in a reduction in performance of common place data association techniques that rely heavily on pose estimates. As such, data association techniques are not likely to be able to close large loops. Even though the loop closing detection problem is conceptually similar to the data association problem, a different approach that is independent of pose estimates is required in order to develop practical SLAM systems that can handle large loops.

2.5 Appearance based localisation

A closely related field to our approach for loop closing detection, which is based on appearance-based techniques, is reviewed. The underlying principle of appearance based localisation is that if the current scene appearance matches one of the observations stored in a database (built from a previous exploration run), a robot has returned to the previous location in which the matching observation was captured. The question is how this can be translated into a loop closing context; if the current scene appearance is similar to one of the observations stored in a growing database (during the exploration phase), it is likely a robot has returned to a previously visited location and thus closed a loop. A key advantage of appearance-based technique is its independence of pose estimate. As such, place recognition plays a central role in topological localisation and loop closing. Place recognition has been utilised in topological localisation applications [16, 23, 122].

2.5.1 Geometric Appearance

Here, geometric appearance localisation is to be distinguished from metric localisation using geometric landmarks. For example, the *Anchor Point Relationships method* (APR) derives from a laser scan a description of the geometric structure of the local scene that can be used to localize the robot without any initial hypothesis of the location of the robot [136]. This higher level description is in the form of spatial relationships between anchor points – reproducible object feature positions that correspond to sharp edges in the angle histogram [137]. This scene comparison process is based on a content-based retrieval system where the APR attempts to find matches to a query laser scan by graphs constructed of anchor points [135].

2.5.2 Visual Appearance

Several topological localisation systems have been developed to take advantage of the discriminative power of vision. These vision-based applications mainly differ in the type of features used to describe an image. The various properties of these features offer different performance in

tackling the perceptual variability problem and occlusions due to dynamic objects in the environment [140]. Ulrich and Nourbaksh [132] developed a system that uses colour histogram obtained from panoramic images. Krose and Bunschoten [67] used principal component analysis of images to represent local scenes. Wang *et al.* [133] uses vectors of *Scale Invariant Feature Transform* (SIFT) image features (which will be discussed in subsection 3.4.2) to describe locations within an environment.

During the training phase, representative images are captured and associated with corresponding locations. The training phase is generally done off-line. In contrast, an on-line appearance-based loop closing system will have to capture a representative signature for each local scene on-the-fly. During operational phase, input image is compared with map's reference images. The location whose reference image best matches the input image is considered the current visible location. This is similar to the problem of image retrieval except that, in some localisation approaches, the knowledge of pose estimate is taken into consideration and the current image is only compared with images captured around its immediate neighbourhood. For instance, an image retrieval system is integrated with Monte-Carlo localisation to localise a robot within an environment [139]. Image retrieval, on the other hand, generally compares the current image with all images stored in a database. In another example, vision-based global localization and mapping was achieved by matching SIFT features detected in a local submap to a pre-built SIFT database map [109], after which 3D submaps are built by aligning multiple frames while the global map is built by aligning and merging multiple 3D submaps.

2.5.3 Combined Appearance

To enhance the robustness of appearance-based localisation techniques, multi-modal perception can be adopted to increase reliability of topological localisation. A fingerprint concept in describing the unique characteristics of a local scene with a circular list of features around a robot is adopted in [72]. The underlying premise is each local scene is unique from one another and local scenes can be discriminated against each other if a rich form of description is used to describe each

local scene. The list of features is obtained from visual feature extractor and laser scan feature extractor from data collected from omni-directional sensors. The fingerprint sequence is denoted as a list of characters, where each character represents an instance of a specific feature.

2.5.4 Comparison of appearance-based localisation with loop closing

Despite the similarities shared, it is important to differentiate between the problem of localisation and loop closing. In localisation, it is implicitly assumed that a robot is operating in a closed world. This assumes that the robot is always inside the area represented by the map. Any observation captured by the robot should therefore match one of the observations in a database built from previous exploration runs. In contrast, loop closing occurs during the exploration phase. The database is growing with each new observation captured. There is no guarantee that the current observation captured by a robot must match one of the observations stored in the database. Conversely, a match does not necessary mean that a robot has returned to a previous visited location either. It could also mean that there is another local scene with similar appearance, albeit in a different location (this is known as perceptual aliasing [70] or scene ambiguity [26] which will be discussed in greater detail in Chapter 5). As such, the appearance based approach alone is insufficient to solve the loop closing problem but serves as a good foundation to build upon. Additionally, the goal of a localisation system is to locate a robot accurately at every pose. In contrast, it is not as crucial that a loop closure decision is made at the first opportunity for loop closing. If and when a robot returns to a previously mapped region, a single, reliable detection of loop closing at any pose is all that is required for a map to be corrected.

2.6 Related Work

In this section, more loop closing cases are briefly looked into. These examples illustrate the trend of using some forms of decision theoretic framework to make loop closing decisions instead of just establishing correspondence. It also points out the need to extend loop closing capabilities for denser and more complex representations of the environment.

2.6.1 Three-dimensional Maps

As a natural extension to building 2D map representation of the environment, the goal of having 3D map representation of unstructured environment is now being pursued [19, 124]. As such, a need for loop closing capability for the three-dimensional case is required. It might seem like a straightforward extension of the two-dimensional loop closing approach [43]. However, this is generally not true due to more general vehicle motion, increased complexity in feature modeling and increased amount of sensing information. A basic method to detect loop closing in a laser-based 3D map is by registering the last captured 3D scan with earlier acquired scans [121]. However, this method is generally computationally intensive given an exhaustive search is required. Without a good initial seed transformation between two 3D scans, it is likely that global minimum will not be achieved. In another approach, loop closing is detected in a vision-based 3D map [107] by checking if there is a significant amount of overlap of landmarks between a current submap and the initial submap. 3D visual landmarks described as SIFT features are compared against each other for correspondence.

2.6.2 Correspondence Map

Similarity between omnidirectional images can be expressed in a distance matrix or a "correspondence map" [76]. The correspondence between image frames within a distance matrix is expressed in a binary relationship. The main diagonal of a distance matrix shows the sequential temporal relationship of the neighbouring frames. The quality of the distance matrix is improved when quality of correspondence between image features are improved through imposing epipolar constraints. An off-diagonal spot represents a correspondence between two frames that are temporally far apart. When the same path is travelled again, it was observed that correspondence appear as a connected off-diagonal in the matrix and reverse-diagonals appear in the matrix when the reverse path is taken [76, 112]. However, this observation was not further exploited in either paper. This phenomenon will be investigated in greater detail in Chapter 5 to assist in making loop closing decisions.

2.6.3 Rehearsal Procedure

An evidential approach was adopted to tackle the loop closing problem in topological map formulation, based on the Dempster-Shafer theory of evidence [4]. In the work, a robot used a wall-following behaviour to travel around the environment. The nodes of the map consisted only of interior and exterior corners of walls. The robot measured the distance between nodes using odometry. The confidence bounds of this estimate was computed with a model of odometry error. When these bounds encompassed a previously visited location, potential loop closing is triggered and the robot made an hypothesis that it has closed a loop. Structural characteristics of a topological map were used to assist the loop closing decision [17]. Using the "rehearsal procedure" [70], a robot attempts to verify its hypothesis by traversing the environment, gathering evidence to support or refute the hypothesis. This is an important point. Loop closing decisions need not necessarily be made immediately but may be delayed until enough evidence accumulates to support the hypothesis.

2.7 Summary

This chapter has described the loop closing problem in the context of SLAM. Several SLAM estimation techniques have been briefly reviewed. This is not an exhaustive list and there are other variants of estimation techniques [25, 79], which tackle the SLAM problem. Our loop closing technique is intended to be independent of any estimation technique. For the sake of clarity, only the delayed-state EKF-SLAM [75, 87], will be employed for the rest of this thesis to demonstrate how our loop closing approach can close large loops in complex environments. The loop closing is divided into two parts; loop detection and map adjustment. A brief review on map adjustment techniques was given before concentrating on the focus of the research by looking into different approaches of detecting loop closures. The loop closing problem is further distinguished from the data association and appearance-based localisation problem. Finally, a review of related work in loop closing is given. The literature review motivates the desire to build a general loop closing

detection approach that is independent of pose estimates, sensor modality and estimation techniques that work across a spectrum of workspaces.

Chapter 3

Scene Analysis and Comparison

3.1 Introduction

The problem of loop closure detection is addressed as a scene recognition problem in this work, in which a robot recognises it has revisited a previous location by scene matching. Section 3.2 motivates the use of scene appearance matching to detect loop closure. The notation and measurement scheme is introduced. Section 3.3 discusses about the properties required of descriptors in order for a scene appearance approach to work in a SLAM context. Section 3.4 describes how visual scene appearance is encoded into a set of invariant descriptors. Several feature detection algorithms are described and a feature description algorithm is briefly reviewed. Section 3.5 describes how spatial scene appearance can be encoded into a set of invariant descriptors. Section 3.6 explains how similarity between two scenes can be measured using various similarity metrics. It is demonstrated that scene recognition can be achieved using images and laser scans through a few experiments. Section 3.7 concludes the chapter.

3.2 Role of Scene Appearance

An environment can be viewed as a set of local scenes. Each local scene is associated with a different location. The motivation for adopting a scene recognition approach stems from various inherent qualities of scene appearance. One of them is the power of discrimination of descriptors

derived from scene appearance. Generally, every location can be distinguished from one another based on its set of descriptors. There is of course the issue of perceptual aliasing or scene ambiguity; where scene appearances of two different locations are similar. This has more to do with the fact that repetitive patterns exist within some environments and not so much of a limitation in the discriminative power of descriptors. The challenge of perceptual aliasing has to be tackled through other means which will be described in subsequent chapters. At the same time, there is no need for higher reasoning such as classifying each individual location to a specific category (such as corridor, computer room, pantry etc.) and using these classifications later to help distinguish different locations. The underlying approach is to solely compare similarity between local scenes. Despite the amount of data used to describe each individual scene (that makes each description discriminative), it is possible to execute the description and comparison processes in an efficient and fast manner.

The notation and measurement scheme that the following sections will rely upon is introduced. As a robot moves around its environment, it senses its local workspace (local scene) at discrete intervals, $t_k; k \in 1, 2, \ldots$. The observation taken at time t_k is denoted as S_k. This notation is extended to the parameterisation of the vehicle state using x_k to represent the robot pose at the time observation S_k was acquired. The set of all observations up to until time t_k is denoted as $S^k = \{S_1, S_2, \cdots, S_k\}$.

Without loss of generality, the extraction of scene (commonly image) descriptors from a particular observation, S_k, is described by some extraction process $\mathcal{E}$ acting on S_k such that

$$\mathcal{E}(S_k) \rightarrow \{d_1, d_2, \cdots, d_n\} \tag{3.1}$$

where d is a parameterisation of a descriptor and n is the number of descriptors returned – which is a function of the observation data. For example, observations of complex scenes may yield a large number of descriptors whereas simple scenes only a few. The precise nature of of S and $\mathcal{E}$ are governed by the sensor modality. Visual image (I) and laser scan (L), two sensor modalities commonly used in robotics, are considered in this work.

3.3 Properties Required of Descriptors

One of the challenges with adopting a scene recognition approach is the issue of perceptual variability. Hence, there is a need to derive viewpoint invariant descriptors that can describe a local scene similarly despite varying perspective. Another challenge is the presence of dynamic objects within the environment. Dynamic objects will result in occlusions and cause variability in the appearance of a local scene. It is desired that the description techniques are robust to occlusions. These are generally the two main properties required of descriptors for the task of scene recognition in a loop closing context. There are other desirable properties which are sensor modality dependent. These properties will be discussed when the scene description process for each sensor modality is discussed.

3.4 Describing Scenes with images

When working with images, $\mathcal{E}_I$ must be specialised to complement one of multitude of schemes capable of extracting suitable wide baseline stable descriptors of image patches. Note that the operator $\mathcal{E}_I$ is actually performing two tasks. Firstly, it selects regions of interest in the scene measurement that are likely to exhibit the property of wide baseline stability (scenes have to be identified from a wide variety of vantage points). Secondly, it produces a suitable parameterisation of these regions. Thus, the operator $\mathcal{E}_I$ may be expressed in a compound form,

$$\mathcal{E}_I(S_k) = \mathcal{E}_D(\mathcal{E}_{ROI}(S_k)) \tag{3.2}$$

where $\mathcal{E}_{ROI}$ is a region of interest operator and $\mathcal{E}_D$ operates on regions of interest to produce parameterisation of scene descriptors.

Three useful forms of $\mathcal{E}_{ROI}$ specialised for use on images shall now be described before moving on to discuss on $\mathcal{E}_D$.

3.4.1 Image Feature Detection

The image feature detection algorithms, $\mathcal{E}_{ROI}$, namely scale saliency algorithm [58], maximally stable extremal regions detector [86] and Harris-affine interest points detector [88] are described here.

Scale Saliency

"Visual saliency is a broad term that refers to the idea that certain parts of a scene are pre-attentively distinctive" [58]. This is similar to the early stages of the human visual system where the human eye picks features that are different from the surroundings. This process is believed to enhance object recognition by disregarding irrelevant information and focusing on relevant information [96]. This formed the underlying motivation of the scale saliency algorithm developed by Kadir and Brady [58], which is based on earlier work by Gilles [36]. Gilles defined saliency in terms of local signal complexity or unpredictability. Specifically, the use of the Shannon entropy [110] of local attributes (intensity) to estimate saliency was suggested. Generally, image patches (a set of pixels) with a flatter intensity histogram tend to have higher signal complexity and, thus, higher entropy. This method was not limited to intensity histogram (the descriptor adopted in Gilles' implementation) and was equally applicable to a histogram of a different descriptor (eg. colour intensity or edge strength) [58].
One limitation of Gilles' method was that a fixed window size, R_x, had to be specified, over which the local probability density function (pdf) may be obtained. Kadir and Brady improved upon Gilles' algorithm to enable it to detect salient regions at multiple scales. In Gilles' definition of saliency, any region that exhibited complexity was considered as salient. However, this implied that regions of pure noise may also be deemed salient, which was not desired. Any feature that exists over large ranges of scale exhibits self-similarity and should be considered non-salient. The detector developed by Kadir and Brady concentrates on features that exist over a narrow range of scales. Salient regions within images are defined as a function of local complexity weighted by a measure of self-similarity across scale space [58].

In the discrete form, the saliency metric $\mathcal{Y}_{\mathcal{D}}$, a function of scale s and pixel location x, becomes:

$$\mathcal{Y}_{\mathcal{D}}(s,x) \triangleq H_D(s,x) \times \mathcal{W}_D(s,x) \tag{3.3}$$

Shannon entropy is defined in the discrete form as:

$$H_D(s,x) \triangleq -\sum_{I \in D} P_{(I,s,x)} \log_2 P_{(I,s,x)} \tag{3.4}$$

where $P_{(I,s,x)}$ is the probability density as a function of scale s, pixel location x and descriptor value I which takes on values in D.

The sum of the absolute differences between the grey-value histograms is used to measure the degree of self-similarity. The inter-scale saliency weighting function, $\mathcal{W}_D(s,x)$ is defined by:

$$\mathcal{W}_D(s,x) \triangleq \frac{s^2}{2s-1} \sum_{I \in D} |P_{(I,s,x)} - P_{(I,s-1,x)}| \tag{3.5}$$

where $|P_{(I,s,x)} - P_{(I,s-1,x)}|$ is the difference in the probability density between scale s and $s-1$. For each pixel location, the algorithm selects those scales at which the entropy, as expressed by Equation 3.4, for the region selected is a maximum. It then weights the entropy value for regions at such scales by a measure of self-dissimilarity, as expressed by Equation 3.5, in the scale-space of the feature [57]. Initially, there is the constraint that the scale is isotropic and, therefore, the method favors blob-like features. This was resolved in [59] to enable detection of elliptical regions. Visually salient descriptors have been used in visual matching, tracking and recognition applications [45]. The notion of saliency was similarly utilised in a localisation application where natural, salient image patches served as high discriminative landmarks for a mobile robot to navigate with [35].

Maximally Stable Extremal Regions

Wide baseline stable image features known as "maximally stable extremal regions" (MSERs) were introduced by Matas *et al.* [86]. "MSERs are defined solely by an extremal property of the intensity function in the region and on its outer boundary" [86]. Desirable characteristics of

MSERs include fast detection rate, affine-invariant stability and multi-scale detection. The process of detecting MSERs is as follows: Consider an image consisting of pixels taking on values in the range $I = \{I_0, \cdots, I_{max}\}$ (for example 8-bit intensity in the range [0:255]). Firstly, an ordering is placed on the pixels based on the values.

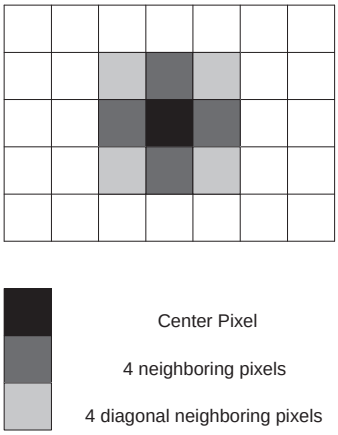


Figure 3.1: The search space of the centre pixel can be either 4 nearest neighbouring pixels or 8 neighbouring pixels.

The *union-find* [20] algorithm performs two operations; the "find" operation determines if two adjacent pixels belong to the same group. The union operation combines two groups into a single group. Starting from pixels with the minimum or maximum value (i.e., 0 or 255 for a greyscale image), the find portion of the algorithm works as follows: Set the index i to 0 and for simplicity, assume only one pixel, q, has value I_0 (I[0]). This pixel is placed in a set R. The method proceeds by incrementing i and examining all connected neighbours (as shown in Figure 3.1) of boundary pixels found in R (which only contain q at the starting point) and adding them to R if their values is within a certain threshold (for example, neighbouring pixels with a value of I[1] around pixel q which has a value of I[0] may be added into set R). The algorithm then iterates once more, incrementing i and this time testing all neighbours of the enlarged R.
In the union portion of the algorithm, two equivalent groups will be united together. A merge of two groups is viewed as termination of the existence of the smaller group and an insertion of all pixels of the smaller group into the larger one. The groups continue to grow until they reach the

boundary of the region of similar intensity values. Given a set of pixels, the union-find algorithm partitions them into a set of non-overlapping groups by maintaining a list of connected components and their areas. In the output, a MSER is represented a position of the local intensity minimum (or maximum) and a threshold. The set R is classified a MSER when its size remains constant with respect to i – the region has stopped growing and there is a discontinuity of pixel values all around its perimeter. For further details on implementation of this algorithm, refer to [20][86]. The reason for the wide baseline stability of features detected by this technique lies in the fact that connectivity is preserved under affine transformations. The application of the MSER algorithm on an image is shown in Figure 3.2.

Figure 3.2: Implementation of MSER algorithm: MSERs extracted from an image are demarcated by red boundaries.

Harris-Affine Detector

Harris-affine interest points [88], which are invariant to scale and affine transformation, is summarised here and the reader is referred to [88] for a complete description. The affine-invariant interest point detector is an affine-adapted version of the Harris detector. A multi-scale Harris detector [24] is used to detect initial interest points. The Harris detector [46] searches for points in which variations in two orthogonal directions are large. It does this by computing a local moment matrix, μ, from image gradients, and it then combines the eigenvalues of the moment matrix to

measure the corner 'strength', where $cornerness = det(\mu) - \alpha\ trace^2(\mu)$ (α is a parameter set to 0.04 as suggested by Harris). Harris interest points by themselves are not scale and affine invariant [106]. The affine adaptation is based on the second moment matrix and local extrema over the scale of normalised derivatives.

The scale-adapted second moment matrix is defined as follows:

$$\mu(\mathbf{x}, \sigma_D, \sigma_I) = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} L_x^2(\mathbf{x}, \sigma_D) & L_x L_y(\mathbf{x}, \sigma_D) \\ L_x L_y(\mathbf{x}, \sigma_D) & L_y^2(\mathbf{x}, \sigma_D) \end{bmatrix} \tag{3.6}$$

where g represents a non-uniform gaussian kernel, σ_I is the integration scale, σ_D is the differentiation scale and L_x and L_y are the derivatives computed in x and y direction respectively. An iterative algorithm modifies location, scale and neighbourhood of each point before converging to an affine-invariant point. Scale invariance is achieved by finding the characteristic scale region around the points of interest [89]. The characteristics scale of a local structure is selected by using the Laplacian-of-Gaussian to select points localised at maxima in scale-space, as in Equation 3.7.

$$|LoG(x, \sigma_n)| = \sigma_n^2 |L_{xx}(\mathbf{x}, \sigma_n) + L_{yy}(\mathbf{x}, \sigma_n)| \tag{3.7}$$

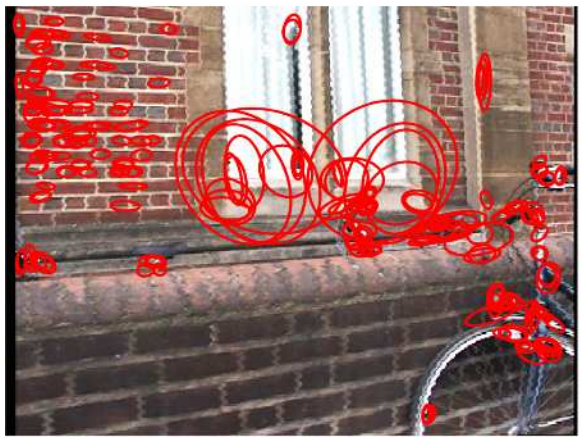


Figure 3.3: Elliptical Harris-affine regions of interest detected.

An anisotropic shape of a local image structure is estimated from the second moment matrix [77] via an iterative algorithm. An example of Harris-affine features is shown in Figure 3.3. A quantitative comparison of the Harris-affine detector with existing detectors in [89] showed a significant improvement in performance in the presence of large affine distortions.

3.4.2 Image Feature Description

SIFT Descriptor

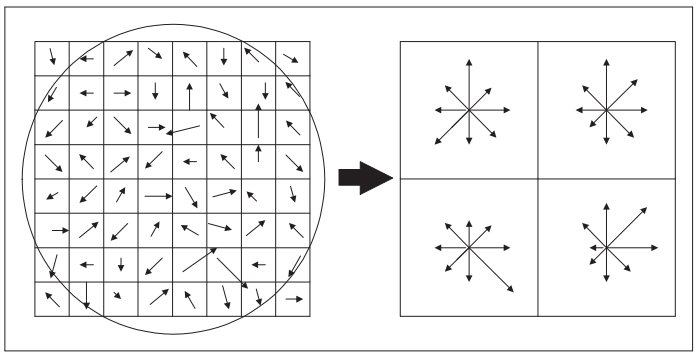


Figure 3.4: A 2×2 SIFT descriptor of gradient magnitudes and orientations is shown. Each arrow represents the orientation gradient and the length of the arrow represents the magnitude of the gradient. Given a 4×4 subregions and 8 orientation gradients in each subregion, a SIFT descriptor is made up of 128 dimensions.

After image patches have been selected by a detection algorithm, $\mathcal{E}_{ROI}$, these patches are encoded by a description algorithm, $\mathcal{E}_D$. A particularly effective description algorithm is the scale invariant feature transform (SIFT) algorithm [80]. This method is based on a model of the behaviour of complex cells in the cerebral cortex of mammalian vision [81]. A SIFT descriptor is created by first computing the gradient magnitude and orientation at each image interest point in a region around the descriptor location. In order to achieve orientation invariance, the coordinates of the descriptor and the gradient orientations are rotated relative to the descriptor orientation. Each sample point is then weighted by a Gaussian window as indicated by the overlaid circle in Figure 3.4. This is to avoid a small shift in the gradient positions from affecting the descriptor drastically and to give

less emphasis to gradients far away from the centre of the descriptor. These sample points are then accumulated into orientation histograms summarizing the contents over 4×4 subregions. The length of each arrow corresponds to the sum of the gradient magnitudes near that direction. The descriptor is a 4×4 array of histograms, each with 8 orientation bins. This results in a 128-dimensional feature vector. The high dimensionality of the SIFT descriptor plays a critical role in matching accuracy.

Desirable characteristics of SIFT features include invariance to a large extent to rotation, scaling, and translation and partial invariance to illumination. In an experiment conducted, the stability of detection for descriptor location, orientation and final matching to a database remains above 50% despite a 50-degree change in viewpoint [81]. There are several other local descriptors [6, 61, 105] that are known to give good performance in image matching. Nevertheless, the SIFT descriptor has been demonstrated to provide the best overall performance among several state of the art descriptors in a series of experiments [90]. It has also been used with good effect in visual SLAM [108].

3.5 Describing Scenes with Laser Scans

This section describes how the spatial appearance of a local scene is encoded by some specialised extraction processes $\mathcal{E}_L(L_k) \rightarrow \{d_1, d_2, \cdots, d_n\}$. A 2D planar laser scan or patch, L_k, (a set of registered scans as defined in [42]) captured at time t_k is used to represent a local scene. As a consequence, scene matching using this specialised extraction process $\mathcal{E}_L$ is limited to mostly planar indoor environments or outdoor environments with flat surfaces.

A laser scan or patch, L_k, is divided into smaller "segments", Seg. $L_k = \{Seg_1^k, Seg_2^k, \cdots, Seg_n^k\}$, where L_k is made of n number of segments. These segments are formed using a standard nearest neighbour clustering algorithm [111] to group neighbouring range points together. A new segment is formed whenever the distance between the points is greater than a distance threshold, τ . These breaks are due to both occlusions and the structure of the environment. Figure 3.5 shows a typical segmentation. The laser patch on the left hand side of Figure 3.5 is broken up into four segments.

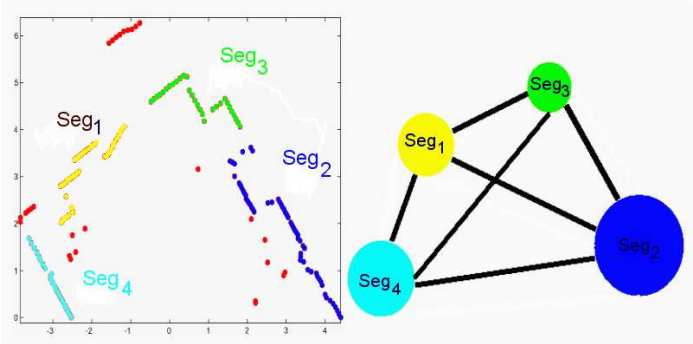


Figure 3.5: A typical geometry patch after segmentation and a graph depiction of how information is encapsulated. Each node is a segment and contains a cumulative angular function, its entropy measure and a list of critical points. The edges represent spatial relationships between segments.

Each segment is represented as a node in the graph on the right. The spatial relationships *between* the segments are encoded into inter-segment descriptors. The generation of these inter-segment descriptors is discussed after considering how the segments themselves are described.

3.5.1 Segment Description

The description processes of spatial properties for segment curvature, segment length ratio and relative entropy are described. Three scalar similarity metrics, η_1, η_2, η_3, are respectively derived.

Cumulative Angular Function

Each segment consists of a set of range points. The contour of the segment can be described by the curvature approximated by these points. The curvature is described by an "cumulative angular" function (CAF) or "turning" function [137, 18], as illustrated by Figure 3.6. A CAF is one-dimensional representation of 2D segments, which encodes the structure of the points within a segment by the change in tangential angles between consecutive points. The turning function is a plot of cumulative change in turning angle ϕ versus arc-length ζ of the segment. To illustrate, the turning function maps straight lines $ax + by + c = 0$ to $\phi = 0$, circles to $\phi = \alpha$ and squares to a "staircase" function in ϕ. Attractive properties of a CAF include translational and rotational

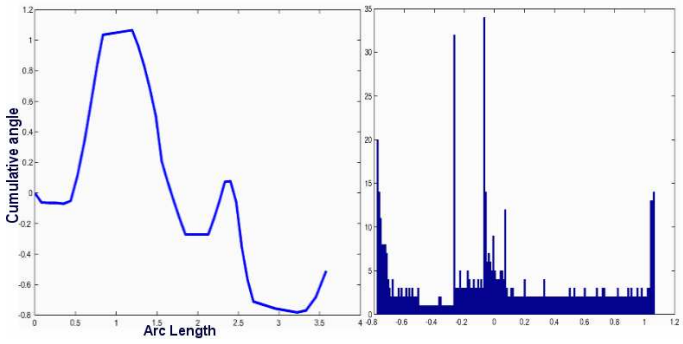


Figure 3.6: On the left is a typical cumulative angular function. The cumulative angular function is transformed into a histogram of angular values on the right. Each bin contains the number of points along the cumulative angular function that have angular values that fall within the bin value.

invariance.

By representing a 2D segment as a 1D shape descriptor, finding the best fit between two segments reduces from a 3D search space $[x, y, \theta]$ problem into a 2D problem [18] in the position-rotation space (β, γ) where the scale is fixed. The query curve is translated vertically and slid horizontally to find the minimum error between the query curve and the pattern curve in a similar fashion as in [18] (see Figure 3.7). The difference, $e(\beta, \gamma)$, between two CAFs is calculated as:

$$e(\beta, \gamma) = \int_0^{\zeta_q} (T_q(\zeta) - T_m(\zeta + \beta) + \gamma)^2 d\zeta \tag{3.8}$$

where the two cumulative angular functions are denoted by T_q and T_m, the position-rotation search space is parameterised by (β, γ) and ζ parameterises the arc-length of the segment.
A scalar similarity measure η_1 lying in $[0, 1]$ is then calculated as

$$\eta_1 = \frac{1}{1 + e} \tag{3.9}$$

Match Length Disparity

A second scalar η_2 is calculated as the ratio of the matched length to the total length:

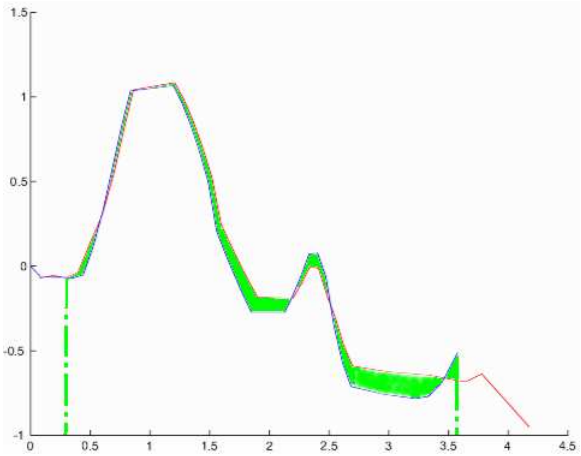


Figure 3.7: The figure shows the disparity between the two CAFs of the two segments, Seg_i^n and Seg_j^m. The difference between the two CAFs is the area between the two curves. Segments that are similar to each other will have similar angular functions and consequently will have a small disparity between their CAFs.

$$\eta_2 = \frac{l(m)}{\zeta_q} \tag{3.10}$$

where $l(m)$ is the length of the matched segment portion and ζ_q is the total length of the query segment.

In Figure 3.7, the matched length is the portion of the abscissa where there is overlap between the two CAFs, and the total length is the length of the query CAF. The larger the portion of the segment that is matched (based on η_1), the more similar the segments are, based on the match length disparity criterion.

Relative Entropy

A measure of the *complexity* of the contour of a segment is desired so that matches between 'complex' shapes can be given emphasis over matches between 'simple' shapes. The reasoning is that a positive match between two complex shapes is more likely to be a true positive than a match between one simple and one complex or two simple shapes. Consider two straight line segments extracted from a laser scan of a long, straight corridor; these straight line segments will

match easily with straight line segments from a laser scan taken at a different portion of the same or different corridor.

A natural way to encode complexity is via entropy. The concept of saliency measure for segments was similarly explored in [37]. However, their definition of saliency was based on the proportion of coverage, which is similar to the match length disparity metric described previously. A segment that covers a large region of the overall shape is considered salient, whereas our measure of entropy is based on the complexity of the shape of the segment, regardless of size. The integral is calculated from a histogram of the cumulative angle function. A typical histogram is shown in Figure 3.6. Each histogram bin contains the number of points along the cumulative angular function that have angular values that fall within the bin value.

In this case, an expression for entropy in the form of Shannon entropy, which is similar to Equation 3.11, may be written in the discrete form:

$$H_D \triangleq -\sum_{\phi \in D} P_\phi \log_2 P_\phi \tag{3.11}$$

where P_ϕ is the probability that angular value ϕ takes on values in D, the set of all angular values along a particular CAF.

Figure 3.8 illustrates the process. Given two segments, respective CAFs are formed. The values along the CAF are then binned and integrated according to Equation 3.11. Note how the line segment in Figure 3.8 has a smaller entropy value (1.8) while the more "interesting" curve segment has a larger entropy (4.3). A complex segment has a CAF with multiple peaks and troughs while a simple segment has a relatively flat CAF. In deriving shape descriptors, emphasis (via thresholding) is placed on encoding segments with high entropy.

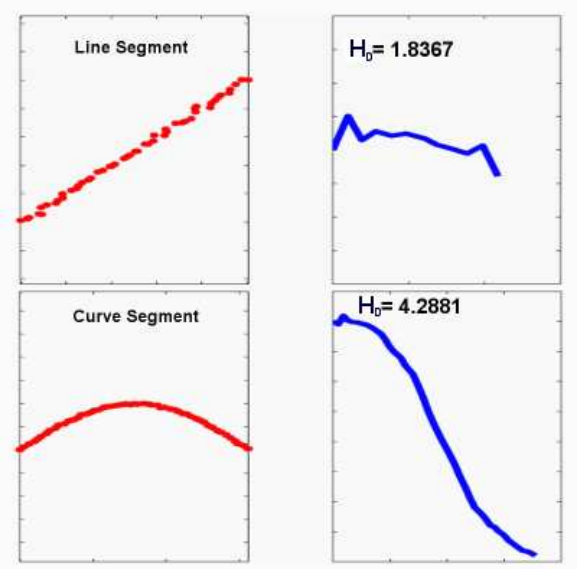


Figure 3.8: Segments of laser data (left) are mapped to cumulative angle functions (right). H_D values are calculated from Equation 3.11.

Entropy Disparity

Relative entropy is used to measure similarity between segments. The relative entropy , or Kullback-Leibler distance, is given by:

$$K(f||f') = \sum_{i=1}^{m} f_i \times \ln \frac{f_i}{f'_i} \tag{3.12}$$

where m is the number of bins and f and f' are the probability distributions approximated by the angle histograms. The smaller the relative entropy, the more similar the two histograms are. When both distributions are equivalent, $K(f||f') = 0$. The relative entropy is normalised to lie within $[0, 1]$ to produce a third scalar η_3.

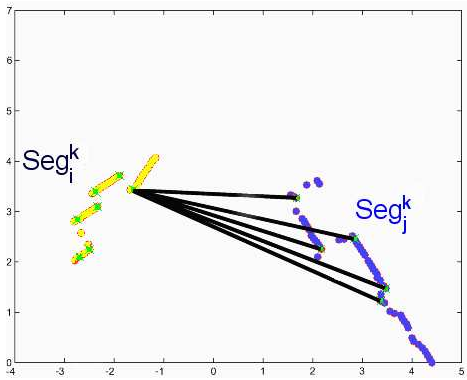


Figure 3.9: Inter-segment links are used to describe the spatial relationship (SE2 transformation) between segments found in S_{L1} and S_{L2}.

3.5.2 Inter-Segment Description

Critical Points

This subsection describes a method for encoding the spatial relationship between segments, which will form the inter-segment descriptors (the edges of the graph in Figure 3.5). Points of high curvature along segments are extracted. These are called 'critical points'. Critical points along a segment correspond to junctures where sharp changes occur along the CAF. They are marked as crosses in the laser scan shown in Figure 3.9. These critical points can be reliably extracted given that a CAF is translational and rotational invariant. The thresholding on entropy selects in favor of segments possessing points of sharp curvature (critical points) that are likely to be visible over a range of vantage points.

Segment Configurations

The distance and relative orientation between critical points form the links (the lines joining the two segments shown in Figure 3.9) that lock two segments in a fixed configuration. Two segments Seg_i^k and Seg_j^k contain n_i and n_j critical points respectively. For each critical point in Seg_i^k a"bundle" of links is formed to all n_j critical points in Seg_j^k. There will be n_i bundles and $n_i \times n_j$ links in total. Only one bundle is shown here. Each bundle, (so long as it contains more than one

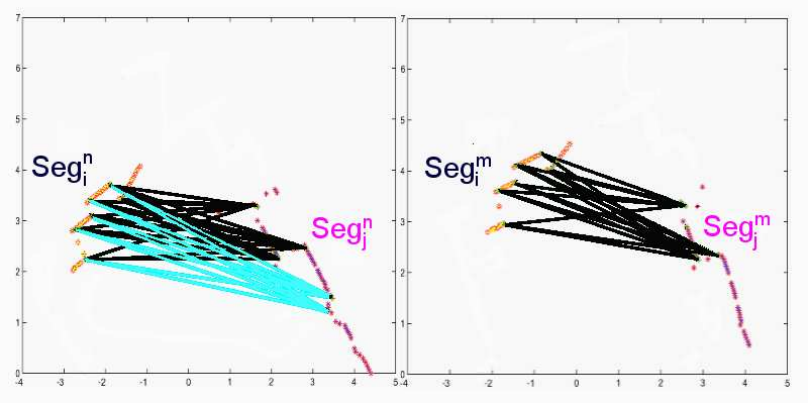


Figure 3.10: A method of comparing inter-segment relationships (edges). In the similarity comparison between edges, the bundles of links that comprise an edge between two segments (Seg_i^n and Seg_j^n) are compared against the bundles of links between two segments (Seg_i^m and Seg_j^m) from another laser scan using distance and relative angle criteria. The dark links represent those that have been successfully matched with links of the other edge. The links are marked in light blue represent links that have not been matched, due to occlusions or the segmentation process.

link) defines a rigid transformation between a critical point in Seg_i^k and the entire segment Seg_j^k. Each edge in the graph in Figure 3.5 is defined to be the set of all bundles from Seg_i^k to Seg_j^k. This is by intent a redundant way to store the relationship between the two segments. To determine the relative orientation between the critical points, the orientation of the segment has to be determined. This is done using simply the eigenvector with the largest eigenvalue of the segment.

Edge Comparison

Besides comparing the shape characteristics of the segments, the matching technique described in this subsection asks if the spatial configurations *between* of segments within a patch are similar to those in another laser patch. As suggested in [136], similarity between the segment-segment edges is determined by matching arrays of distances and relative orientations of the segment-segment edges. The bundles of links that comprise an edge between two segments (Seg_i^n and Seg_j^n) are compared against the bundles of links between two segments (Seg_i^m and Seg_j^m) from another laser scan. It should be noted that comparison is made between individual bundle of links corresponding to each critical point. In Figure 3.10, the segment-segment relationships for two laser scans are

shown. Due to occlusions, a minority of the critical points found in one laser scan is not seen in the other. The links that are successfully matched are highlighted in a darker tone. The quality of the match between the edges is determined by the ratio of the matched links versus the total number of links:

$$q_m = \frac{n_m}{n_r} \tag{3.13}$$

where n_m is the number of matched links and n_r is the number of links between segments in the query scan.

3.6 Scene Comparison

Scene matching is achieved by computing a measure of similarity between the observations. Using the extraction processes $\mathcal{E}$ described in Section 3.4 and Section 3.5, an observation, S_i, is represented by a set of descriptors, $\overrightarrow{S}_i$. A similarity function $Sim(\overrightarrow{S}_i, \overrightarrow{S}_j) \in [0, 1]$ measures the similarity between observations S_i and S_j and assigns a pairwise similarity score. A potential match between observations is allowed if the similarity value is above a certain threshold. Different similarity metrics may be appropriate for different descriptors. This section looks into various comparison techniques individually and as a combination.

3.6.1 Comparison Techniques

A brief overview of comparison techniques that are used in this work are explained. This is by no means an exhaustive list. These techniques are chosen for their simplicity and relevance to our application.

Voting Algorithm

The basis of a voting algorithm is to sum the number of matches between descriptors of two observations. Consider the case in which a query observation is compared with a database consisting of a set of observations $S_1, \cdots, S_k$. Each observation is described by a vector of

descriptors $\overrightarrow{S}_i$. During the comparison process, each descriptor from one vector is matched against the descriptors from another vector. For example, similarity between two SIFT descriptors can be quantified using the Euclidean distance as follows:

$$d_{euc} = ||d_i - d_j|| = \sqrt{\sum_{k=1}^{128}(d_i(k) - d_j(k))^2} \tag{3.14}$$

where d_{euc} is the Euclidean distance, d_i and d_j are SIFT descriptors from observation S_i and S_j respectively and k is the index number.

If the distance, d_{euc}, is below a certain threshold, a match $id(d_i, d_j)$ is considered found and a vote is added to the observation vector.

$$id(\overrightarrow{S}_i(k), \overrightarrow{S}_j(r)) = \begin{cases} 1 & \text{if for } d_k \in \overrightarrow{S}_i \text{ and } d_r \in \overrightarrow{S}_j, \min\limits_{r=1:m} || d_k - d_r || < \epsilon \\ 0 & \text{otherwise} \end{cases}$$

where $\overrightarrow{S}_i$ and $\overrightarrow{S}_j$ are observation vectors with n and m as dimension sizes, k and r are index numbers, d_k and d_r are corresponding descriptors and ϵ is a set threshold.

The observation vector that obtains the highest number of votes is considered most similar to the query observation:

$$Sim(\overrightarrow{S}_i, \overrightarrow{S}_j) = \sum_{k=1}^{n} id(\overrightarrow{S}_i(k).\overrightarrow{S}_j(r))$$

Cosine Distance Function

The similarity between two observations, S_i and S_j can be measured by calculating the cosine of angle (θ_c) between the two representative vectors of descriptors, $\overrightarrow{S}_i$ and $\overrightarrow{S}_j$. A vector can be normalised by dividing each of its component by its length. This ensures that observations comprising of more descriptors do not score better just by virtue of the number of descriptors.

$$Sim(\overrightarrow{S}_i, \overrightarrow{S}_j) = cosine(\theta_c) = \frac{\overrightarrow{S}_i.\overrightarrow{S}_j}{|\ \overrightarrow{S}_i\ ||\ \overrightarrow{S}_j\ |} \tag{3.15}$$

where $\overrightarrow{S}_i$ and $\overrightarrow{S}_j$ are vectors of descriptors representing observations S_i and S_j respectively.

$$d_{cos}(\overrightarrow{S}_i, \overrightarrow{S}_j) = \frac{\sum_{k=1}^{n} \overrightarrow{S}_i(k)\overrightarrow{S}_j(k)}{\sqrt{\sum_{k=1}^{n} \overrightarrow{S}_i(k)^2}\sqrt{\sum_{k=1}^{n} \overrightarrow{S}_j(k)^2}}. \tag{3.16}$$

where $\overrightarrow{S}_i(k)$ and $\overrightarrow{S}_j(k)$ are descriptor weights from the respective vectors and n is the size of the vector.

Combination of Similarity Measures

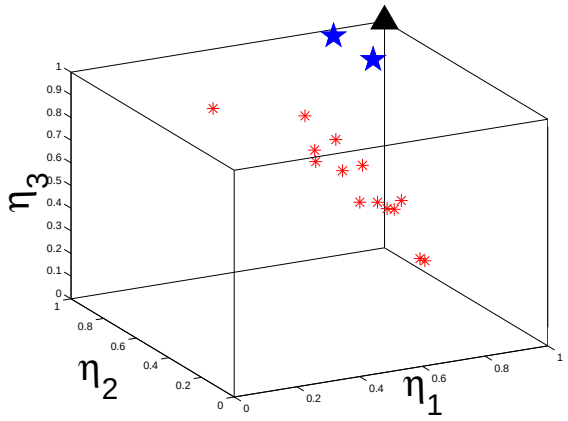


Figure 3.11: Segment to segment matching using the similarity vector $\eta_{Seg_i^n, Seg_j^m}$. The triangle represents a perfect match (identical segments). The figure shows the similarity between a query segment, Seg_q and all segments from two other scans L_n and L_m. Two close matches are found which are depicted as stars. The axes are angular function similarity measure η_1, matched length ratio measure η_2 and entropy similarity measure η_3.

Different similarity metrics based on different characteristics of an observation can be used conjunctively to give an overall similarity score. The rationale behind using a combination of similarity metrics is that even though one comparison technique may accidentally allow an invalid match, it is unlikely several techniques will, given that these techniques are independent.
Here, a description is given of how two generated segment descriptors generated can be compared with one another using a combination of similarity metrics. Each segment is described by a CAF,

its entropy measure and a list of critical points. When comparing two such segments, three disparity measures are used. The above three similarity scalars are stacked in vector $\eta_{Seg_i^n, Seg_j^m} = [\eta_1, \eta_2, \eta_3]^T$ that describes the degree of similarity between Seg_i^n and Seg_j^m. If Seg_i^n and Seg_j^m are identical segments, $\eta_{Seg_i^n, Seg_j^m}$ will be $[1, 1, 1]^T$. In Figure 3.11, the position of every segment is displayed in η-space. The triangle represents the position of a perfect match with the query segment in all three similarity measures. The stars correspond to the segments that are most similar to the query segment. The asterisks are the positions of other segments in η-space.
The segment similarity metric is comprised of two parts: the shape similarity between two segments and the spatial configuration similarity between the segments. The quality of match between segment Seg_i^n from the query scan and segment Seg_j^m from the reference scan is defined by Equation 3.17.

$$SegSim_{i,j}^{n,m} = \lambda \times \eta_{Seg_i^n, Seg_j^m} + (1 - \lambda) \times q_m \tag{3.17}$$

where $SegSim_{i,j}^{n,m} \in [0, 1]$ and the parameter $\lambda \in [0, 1]$, determines the relative importance attached to the matching of the shapes of the segments and the links between the segments. λ is determined experimentally so as to produce optimal matching results. q_m was previously defined in Equation 3.13.
Given the set of matching scores $SegSim_{i,j}^{n,m}$ between all segments from query laser scan, L_n, with all segments from reference laser scan, L_m, total matching score is maximised subject to the constraint that matching between segments must be one to one. This is solved by using the Hungarian method [101], which finds the optimal combination of segments matches that gives the highest matching score possible between the query and reference laser scans.

3.6.2 Results

Image Comparison

For the following experiments, image features are detected by the Harris-affine detection algorithm and described by the SIFT descriptor algorithm. Similarity between images is based on a voting

(a) Query Image (b) Most similar Image (c) Second most similar Image

(d) Third most similar Image (e) Fourth most similar Image (f) Fifth most similar Image

Figure 3.12: A ranked list of images deemed most similar to the query image. It can be observed that image similarity decreases as the change in viewpoint increases. For the last image, a substantial part of the image was obscured with entities not found in the query image.

algorithm on putative correspondence between SIFT descriptors. A ranked list of images deemed most similar to the query image is produced. In Figure 3.12, it appears there is a relationship between the similarity ranking and the degree of change in viewpoint from the query image. The exception is the last ranked image, in which a substantial part of the image was obscured with entities not found in the query image.

Figure 3.13 shows more experimental results from the proposed image matching system with a different image database. The top row contains the query images. Down the column are the corresponding matches in descending order of similarity. Notice the robustness of the image matching system despite dynamic objects such as humans and moving vehicles. However, the image in row 2, column 1 is an example of how visual match alone will produce wrong loop detection in environments where there are repetitive visual artefacts This is a good example of the perceptual aliasing problem. This problem will be looked at in greater details in Chapters 5 and 6.

Figure 3.13: The top row contains the query images. Down the column are the corresponding matches in descending order of similarity. Notice the robustness despite dynamic objects such as humans and moving vehicles. In row 2, column 1, the image is an example of how visual match alone can produce wrong loop closing detection when there are repetitive visual artefacts.

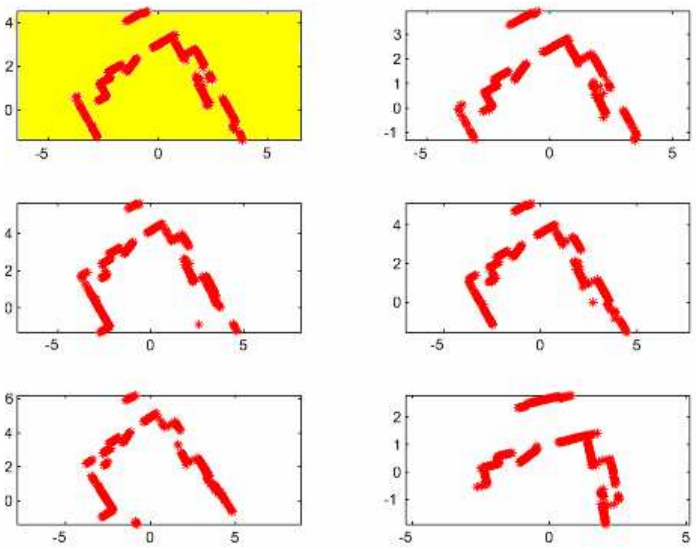


Figure 3.14: A query laser scan, shown in the yellow box, is matched against a database of laser scans. The top 5 laser scans are shown above. Laser scans are ranked accordingly to their similarity. The best ranked laser scan is shown at the top right hand corner and lower ranked laser scans are arranged in a zigzag fashion with the lowest ranked laser scan at the bottom right hand corner.

Scan Comparison

In Figure 3.14, the query laser scan, shown in the yellow box, is matched against a set of 184 laser scans. τ (previously defined in Section 3.5) is set at 0.3 and λ is set at 0.3. The top 5 laser scans are shown above. The laser scans are ranked accordingly to their similarity using our similarity measure. The best ranked laser scan in terms of similarity is shown at the top right-hand corner and the respective lower ranked laser scans are arranged in a zigzag fashion with the lowest ranked laser scan shown at the bottom right-hand corner. Figure 3.15 demonstrates more clearly the rotational and translational invariance of our descriptors and the ability of our system to match despite substantial occlusions.

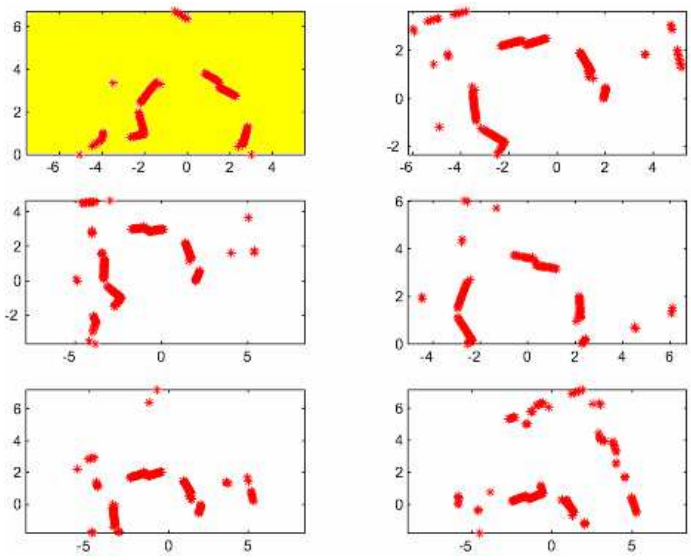


Figure 3.15: These results demonstrate more clearly the rotational and translational invariance of our descriptors and the ability of our system to match despite substantial occlusions.

3.7 Summary

In this chapter, the notion of a local scene and how an environment can be represented as a set of observations is introduced. An extraction process encodes an observation associated with a local

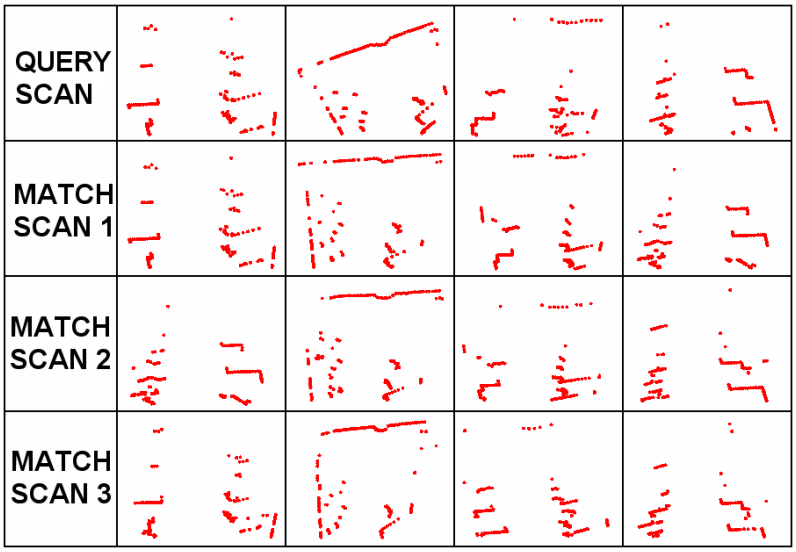


Figure 3.16: The top row contains the query laser scans. Down the column are the corresponding matches in descending order of similarity.

scene into a set of invariant descriptors. Specific techniques to derive a set of invariant descriptors from an image or a laser scan are described. If and when better and more robust description algorithms are available, such algorithms may be utilised in our approach. Consequently, local scene comparison is likely to improve with the advent of better description algorithms. In the next chapter, the focus will be to improve the performance of the "one-shot" algorithm that has been described here by taking in consideration difference in the discriminative power of individual descriptors.

Chapter 4

Scene Retrieval for Loop Closure

4.1 Introduction

This chapter discusses the role of scene retrieval in detecting loop closing. Section 4.2 describes the SLAM formulation used in this work. Section 4.3 demonstrates how an image retrieval system can be used as an one-shot loop closure system in a simple environment. Section 4.4 illustrate the problem of retrieval ambiguity because not all descriptors are equally discriminative. It motivates the need to assign different weights to different descriptor matches. Section 4.5 describes how to mitigate retrieval ambiguity by creating a visual vocabulary through clustering of descriptors. From these clusters, weights are assigned to individual descriptors based on the *inverse document frequency* formulation. The effects of algorithm and environment context on visual vocabulary are demonstrated. Section 4.6 outlines how to improve retrieval efficiency by indexing visual words and using a k-d tree structure to store descriptors. Finally, Section 4.7 concludes the chapter.

4.2 A SLAM System

The section provides a brief summary of the SLAM system which will be augmented with loop closure detection. The SLAM formulation will be used for all experiments described in the rest of the thesis.

4.2.1 SLAM Implementation

A laser-based scan-matching, delayed-state EKF SLAM algorithm is employed. This choice is made entirely without prejudice – any SLAM algorithm could have been used. This particular method is chosen because it is simple to explain and offers good performance in the chosen environment. The SLAM technique described below is close in spirit to references [83, 62], uses the delayed-state ideas in references [75, 87] and is similar to one of the SLAM schemes employed by Bosse *et al.* [9], although a different scan matching technique is used.

Vehicle Model

In the 2D case, $\mathbf{x}_v$ is a three element vector $[x, y, \theta]^T$, whereas $\mathbf{x}_v$ is a six element vector of $[x, y, z, \theta, \phi, \psi]^T$ for the 3D case. At some time t_{k+1}, the robot is subject to a noisy control vector $\mathbf{u}(k+1)$ such that the predicted position of the robot can be written as a function of the control and the last state estimate. The common notation that the quantity $\mathbf{x}(i|j)$ is the estimate of the true state $\mathbf{x}$ at time t_i given measurement up until time t_j is adopted.

$$\mathbf{x}_v(k+1|k) = \mathbf{x}_v(k|k) \oplus \mathbf{u}(k+1) \tag{4.1}$$

where $\oplus$ is a 3D or 6D transformation composition operator as used originally by Smith *et al.* [115], which has the following two Jacobians associated with it:

$$\mathbf{J}_1(\mathbf{x}_1, \mathbf{x}_2) = \frac{\partial(\mathbf{x}_1 \oplus \mathbf{x}_2)}{\partial \mathbf{x}_1}$$
$$\mathbf{J}_2(\mathbf{x}_1, \mathbf{x}_2) = \frac{\partial(\mathbf{x}_1 \oplus \mathbf{x}_2)}{\partial \mathbf{x}_2}$$

These allow the second-order statistics of $\mathbf{x}(k+1|k)$ following a control input to be written as

$$\mathbf{P}_v(k+1|k) = \mathbf{J}_1(\mathbf{x}_v, \mathbf{u})\, \mathbf{P}_v(k|k)\, \mathbf{J}_1(\mathbf{x}_v, \mathbf{u})^T + \mathbf{J}_2(\mathbf{x}_v, \mathbf{u})\, \mathbf{R}\, \mathbf{J}_2(\mathbf{x}_v, \mathbf{u})^T$$

where the $(k|k)$ and $(k+1)$ indices have been dropped from $\mathbf{x}_v$ and $\mathbf{u}$ respectively for clarity and $\mathbf{R}$ is the covariance of the noise process in control $\mathbf{u}$.

State Vector

The estimated quantity is a state vector $\mathbf{X}(i|j)$ which initially contains $\mathbf{x}_v(0|0)$, a single robot pose. Associated with this is a covariance matrix $\mathbf{P}(0|0)$. A delayed-state model is employed in which, at every time step, the state vector is augmented as follows:

$$\mathbf{X}(k+1|k) = \begin{bmatrix} \mathbf{X}(k|k) \\ \mathbf{x}_{vn}(k|k) \oplus \mathbf{u}(k+1) \end{bmatrix} \tag{4.2}$$

$$= \begin{bmatrix} \mathbf{x}_{v1} \\ \vdots \\ \mathbf{x}_{vn} \\ \mathbf{x}_{vn+1} \end{bmatrix} (k+1|k) \tag{4.3}$$

The state vector is simply a vector of previous robot poses, where the notation is extended to write the i^{th} pose as $\mathbf{x}_{vi}$. No environment landmark is stored. The map is implicitly formed from the laser scans associated with each stored vehicle pose. Associated with each pose is a laser scan (2D or 3D) and the latest image captured.

The augmented covariance matrix $\mathbf{P}$ can be written as:

$$\mathbf{P}(k+1|k) = \begin{bmatrix} \mathbf{P}(k|k) & \mathbf{P}_{vp}(k+1|k) \\ \mathbf{P}_{vp}(k+1|k)^T & \mathbf{P}_v(k+1|k) \end{bmatrix} \tag{4.4}$$

It should be noted that k is not incremented in every iteration of the algorithm. The odometry readings of the robot are compounded until the overall change in pose is significant (for example in most implementations approx. 0.5m in distance or 15^o in heading). This overall, compounded transformation becomes $\mathbf{u}(k)$ and the k is incremented and the above described state project step undertaken. In this way the state vector grows linearly with the exploration path length and not with time.

Inter-pose measurement

Measurements relating to $\mathbf{X}$ are made with a scan matching algorithm which works as follows: Consider two poses at times t_i and t_j. Poses at times t_i and t_j have associated laser scans L_i and L_j, each containing n_i and n_j sets of x, y points (or x, y, z points in 3D) in the vehicle frame of reference. Assuming that there is a substantial overlap between the surfaces sampled in these two scans, it finds a transformation T parameterised by the vector $\mathbf{z}_{ij} = [x, y, \theta]^T$ or $\mathbf{z}_{ij} = [x, y, z, \theta, \phi, \psi]^T$ such that:

$$\kappa = \sum_{k=1:n_j} \Phi(L_i, T(L_j^k, \mathbf{z}_{ij})) \tag{4.5}$$

is minimised. The function $\Phi(L_i, T(L_j^k, \mathbf{z}_{ij}))$ returns the unsigned distance between the k^{th} point in scan L_j transformed by $\mathbf{z}_{ij}$, and all points of scan L_i. Note that point-to-point association, which is common in *iterative closest point* algorithm [32], is not performed here. In the implementation, Φ uses the distance transform of L_i and uses the coordinates of the transformed points of L_j to look up the distance to the template scan L_i. Two important points should be made. Firstly, the scan-matcher needs to be seeded with an approximate initial estimate of $\mathbf{z}_{ij}$. The current implementation has a convergence basin for typical indoor environments (laboratories, offices and corridors) of approximately $\pm$ 30° and $\pm$ 5 metres and takes 40 ms to compute. The

need for a ball-park initial estimate is not surprising, as scan-matching is a non-linear optimisation problem and thus is vulnerable to the presence of local minima. Secondly, as Lu and Milios [83] described, scan matching can be used to provide constraints or "measurements" of the relationship between poses. In this case the output of the scan matcher is the transformation between pose i and pose j in the state vector. For example, matching between scans $k+1$ and k allows the following measurement equation to be formed:

$$\mathbf{x}_v(k+1|k) = \mathbf{x}_v(k|k) \oplus \mathbf{z}_{k,k+1} \tag{4.6}$$

There are several ways to use Equation 4.6. The measurements could simply be stored and used as an observation in a sparse bundle adjustment, as proposed by Konolige [62]. It can also be used in a minimum mean-squared error update step. Essentially, the equation is linearised and used as an observation in a non-linear Kalman filter which explains the observation as a function of just the last two pose entries in the state vector. Nevertheless it is important to note that the update will alter the entire state vector (which is the robot's past trajectory).

4.3 An One-shot Loop Closure System

This section demonstrates how an image retrieval system can be utilised to detect loop closing events in a delayed state EKF SLAM system. Every image associated with a robot pose is stored in a database as a set of descriptors. Each newly captured image is compared against every other image stored in the database and similar images are identified. When a robot returns to a previously visited location after a circuitous route, a newly captured image will start to match with a previously stored image. The principle of how an one-shot loop closure system works is the same as an appearance-based localisation system.

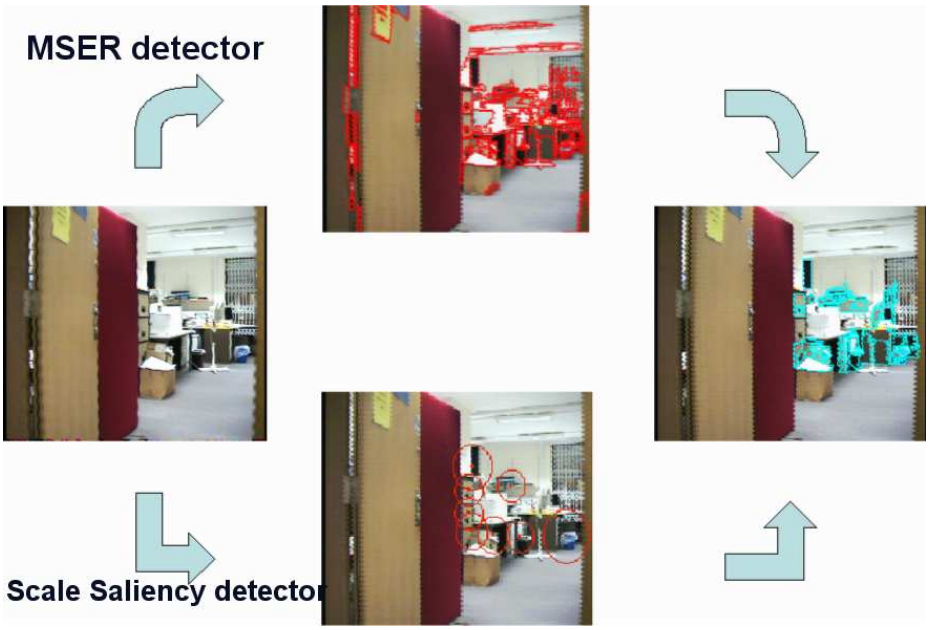


Figure 4.1: Image features detected through two pipelines involving MSER and scale saliency detection algorithms. Different image patches are selected by the two algorithms. The final image patches selected are MSER features that are found within salient regions.

4.3.1 Description of Scene Appearance

As a robot moves around its environment and explores new areas, it regularly (every few seconds or metres of path driven) takes an image using an onboard camera. In this experiment, a set of descriptors is extracted from an image using a combination of the scale saliency and MSER detection algorithms and the SIFT description algorithm [98]. Figure 4.1 illustrates the process for selecting image regions based on the scale saliency and the MSER algorithms. The raw image input (leftmost image) is at the start of the flow process. The image is first passed through the MSER algorithm described in subsection 3.4.1. MSERs are selected from the raw image, as depicted by regions with red boundaries in the top image. Next, the image is passed through the scale saliency algorithm. Salient regions are selected from the raw image, as depicted by red circles in the bottom image. The final image regions that are selected are MSER regions that are found within the salient regions. These regions are depicted with cyan boundaries in the rightmost image.

4.3.2 Scene matching to prompt loop closure

The choice of SLAM algorithm described in Section 4.2 makes loop closing events particularly easy to handle. Imagine an oracle provides a correspondence $c_{i,k}$ where $i << k$ — i.e. that relates the current pose (end of state vector) to a pose encountered a long time ago. This may well be a loop-closure event. All that is needed to use the measurement is to rewrite Equation 4.6 in terms of pose states k and i and proceed as before with a standard EKF update. Because the whole trajectory is stored in the state vector, all previous pose states are adjusted in proportion to their uncertainty in order to accommodate the loop closure assertion. The question now is how can such an oracle be built? If the current view from the camera is matched to a previous view, and there is confidence that the matching process is highly discriminative, as is certainly the case with the visual saliency scheme in use here, then it is highly likely that the robot is in the neighbourhood of the earlier pose.

A database of images is incrementally built as the robot continues to explore. Each image is time-stamped. The database is queried every time a new image is acquired before adding it to the

database. The mechanism employed to perform the query is based on the voting algorithm described in subsection 3.6.1. It has $O(n^2)$ complexity in terms of the number of descriptors. The query image I_Q generates n_q descriptors. $I_Q = \{d_q(1), \cdots, d_q(n_q)\}$. For each stored candidate image I_C in the database with n_c descriptors, $I_C = \{d_c(1), \cdots, d_c(n_c)\}$, an $n_q \times n_c$ adjacency matrix $A_{q,c}$ is created where the $(i,j)^{th}$ entry in $A_{q,c}(i,j)$ is the $\mathcal{L}_2$ norm $|| \; d_q(i) - d_c(j) \; ||$. These distances are compared to a threshold, resulting in n_{qc} matched descriptors between the query image and the candidate image in the database. When all images have been compared, those candidates producing the largest number of feature matches n_{qc} are selected as wide-baseline matches.

Consider the case when a new image is added to the database and a match is found between that and an image taken much earlier in the SLAM session. One field of the query result contains the time t_m at which the earlier image was taken. Under reasonable assumptions (which are discussed later), this match makes a strong assertion that the robot is again close to where it was at time t_m. By keeping an external journal of the position and time (which must be updated if the SLAM algorithm employed makes substantive changes to old position estimates), a search can be initiated to relocate the robot near where it was at time t_m or to make a concerted effort to associate current measurements with components of the map built earlier at t_m.

The scan matcher can be run (with an initial zero transformation seed) to find the transformation between the two proximate poses. Scan matching is not run only between a laser scan associated with the most recent pose, k, and a laser scan associated with some historical pose. Rather, a scan patch [42] consisting of a set of laser scans associated with a set of recent poses is used for scan matching with another scan patch from a set of previous poses. This set of recent poses is presented by a single pose $q = k - n/2$, where n is the number of poses within the set. If the query image at pose time t_q matches an image taken at the time t_m associated with pose m, then another scan patch is produced around m. The query and match scan patches are described with respect to the pose frames q and m respectively. The motivation for the use of the patches – essentially simulating a multi-viewpoint scanner – is as suggested by Gutmann and Konolige [42]

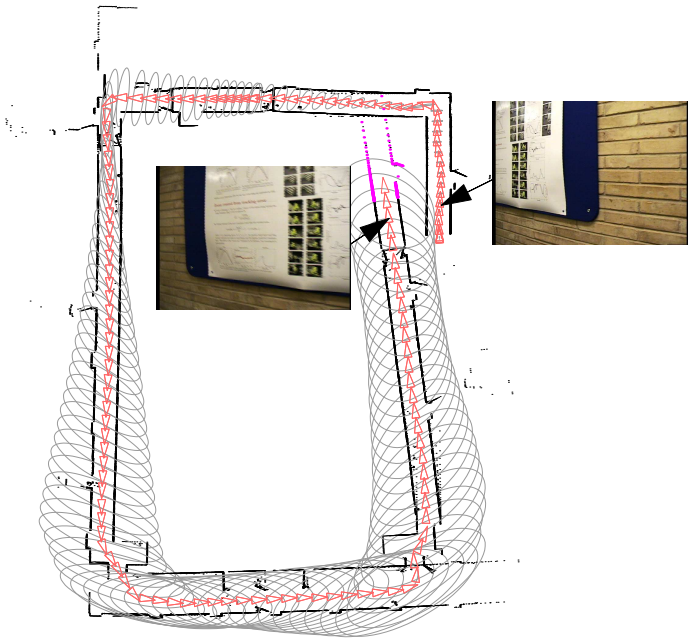


Figure 4.2: The robot poses stored in the state vector are shown as triangles. The performance of the SLAM algorithm is just as would be expected. The global uncertainty (grey ellipses) increases with the length of the excursion from the start location increases. A poor scan match at the bottom right introduced a small angular error that leads to a gross error in the pose estimate, when in reality the robot has returned to near its starting locations (top right). The inset images shown are the two camera views used in the loop-closing process. The left-hand image is the query image and the right-hand one is the retrieved, matching image. The poses that correspond closest in time to the two images are indicated with arrows.

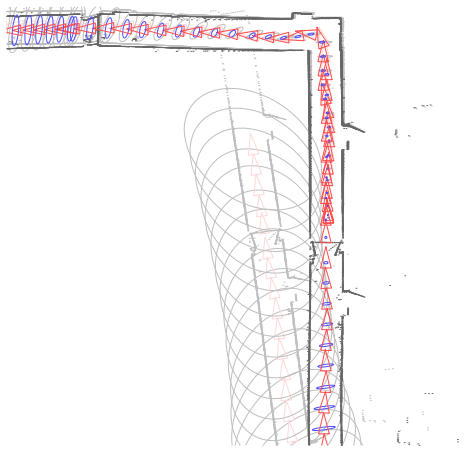


Figure 4.3: A close up of the region around the point of loop-closure. The pre-closure trajectory and uncertainties are shown in faint ink. Note how the insertion of a loop-closing constraint between two poses that are temporally very distant causes a marked reduction in uncertainty (ellipses) in the recent poses.

to decrease the interpose ambiguity during the scan matching process. Finally, the scan-matcher is run to produce a transformation between pose m and q. An estimate of the uncertainty in this match is derived by fitting a quadric to the error surface near the optimised transformation. From this quadric, the Hessian and hence a suitable covariance matrix can be derived. The entire procedure is summarised in Algorithm 1.

4.3.3 Results

A small ATRV-Jnr mobile robot was driven around a building with a loop of approximately 100m in length. It should be noted that this is by no means a large loop or an extremely challenging environment for contemporary SLAM algorithms. However, the accumulated spatial error is significant and serves to highlight the effectiveness of using an image retrieval system to close loops without recourse to pose estimates. The robot camera kept a constant orientation in robot coordinates – looking forward and slightly to the right. Every two seconds an image was grabbed and written to disk. The robot was equipped with a LMS 200 SICK laser. The range output from the SICK laser was logged along with the odometer reading from the wheel encoders. Each image

Algorithm 1 Algorithm for loop closing by image retrieval

for every given distance or time interval **do**
  Capture a new image I_k and time stamp image t_k
  Reduce image into a set of descriptors with chosen detection and description algorithm: $\mathcal{E}_I(I_k) \rightarrow \{d_1, \cdots, d_{n_k}\}$
  if first image I_1 **then**
    Store set of descriptors into database
  else
    Compare new set of descriptors with previously stored sets of descriptor with an adjacency matrix, $A_{q,c}$
    if Number of descriptor matches exceeds threshold: $n_{qc} > n_{threshold}$ **then**
      Trigger loop closure
      Retrieve corresponding laser scans, L_k, L_m, from time stamp, t_m
      Employ exhaustive scan matching to realign map
    end if
    Store set of descriptors into database
  end if
end for

was time-stamped, processed and finally entered into a database as a collection of feature descriptors. The simple SLAM algorithm described in section 4.2 was run using only the raw laser data and odometry data. Figure 4.2 shows and describes the state of the algorithm just before the first loop closing event occurred. Figure 4.3 shows a close-up of the region close to the point of loop-closure before and after loop closing is applied.

The top row of Figure 4.4 shows the correspondences found between the loop-closing images. Note how most of the lines are parallel, but not all. This apparent mismatching is a peculiarity in the scene by chance. The images are of a poster that contains multiple, self-similar pictures (of a luminous green, gloved hand). The challenge of having common descriptors within an environment (retrieval ambiguity) will be discussed in greater detail in the following section. Nevertheless, the probability of a false positive is low given the number of correspondences found. Furthermore, false correspondences can be removed through epipolar constraints. The bottom two images in Figure 4.4 show two other similar images in the database that were successfully discriminated against. In all, the database comprised 250 images and the robot drove twice round the loop shown in Figure 4.5.

Finally, Figure 4.5 shows the final map after applying the loop closing constraint. As expected, the

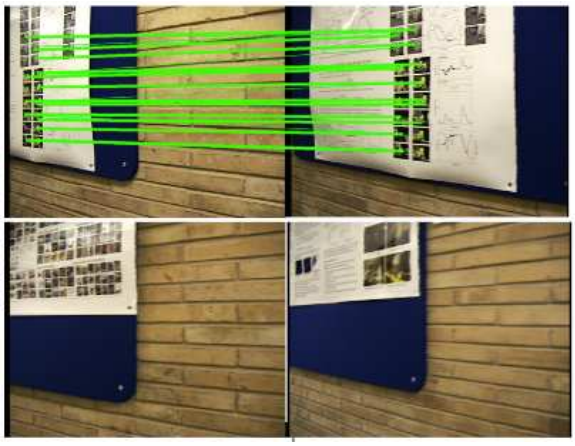


Figure 4.4: Feature-to-feature correspondences found between two images (top right and top left) used in the loop closing event shown in Figure 4.2. The bottom two images are similar images in the database that were successfully discriminated against.

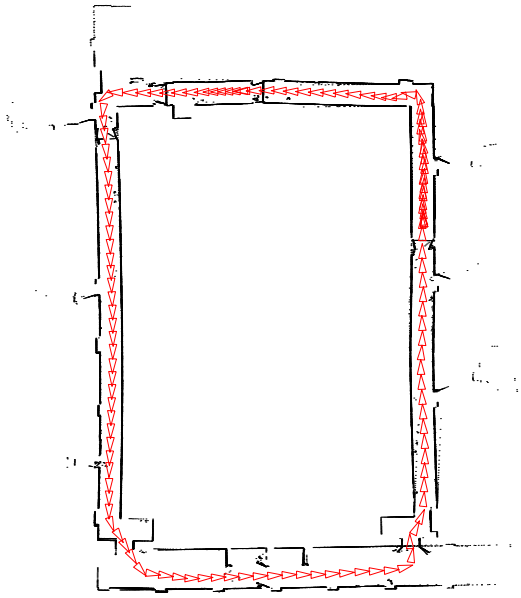


Figure 4.5: Complete map of the test area just after loop closing. Triangles represent robot pose estimates.

marginal covariances on each robot pose decrease and a crisp map results. Although it is incidental to the focus of this research, it is worth noting that the multiple-pose formulation used here has the disadvantage of not being able to refine the map over multiple passes without inserting more robot states. A landmark-based approach does not have this issue. However, it falls short of fully utilising the richness of the laser data by limiting the map to a collection of simple geometric primitives.

4.4 Retrieval Ambiguity

Earlier, similarity between images was measured using a voting scheme based on the number of correspondences between descriptors. It is contended that descriptors are not equally discriminative. Consequently, different descriptor correspondences should not be given equal importance. Figure 4.6 shows a common descriptor (defined as within a certain Euclidean distance) found in multiple images. These images are captured at different locations in a building. This particular descriptor is not unique to any particular image. It makes sense to place less importance on matching this descriptor as compared to another descriptor that is only found in a single image throughout the database. The question is how to assign the correct importance or weight to a descriptor.

4.5 Mitigation of Ambiguity

4.5.1 Clustering of Descriptors

An image can be considered as a text document in which an image is comprised of 'visual words' (descriptors). The vector space model [10, 114] is employed. The first step is to build up a visual vocabulary, $\mathcal{V}$, for a sequence of images collected from an explored environment. Image features detected by the Harris-affine detector and described by SIFT descriptors form the basis of the visual vocabulary for this work. The visual vocabulary is formed by clustering SIFT descriptors that are similar to each other. Descriptor similarity is defined in terms of the Euclidean distance in Equation 3.14. Each SIFT descriptor, d_i, is associated with a visual word $\hat{\mathbf{d}}_\mathbf{i}$ in the visual

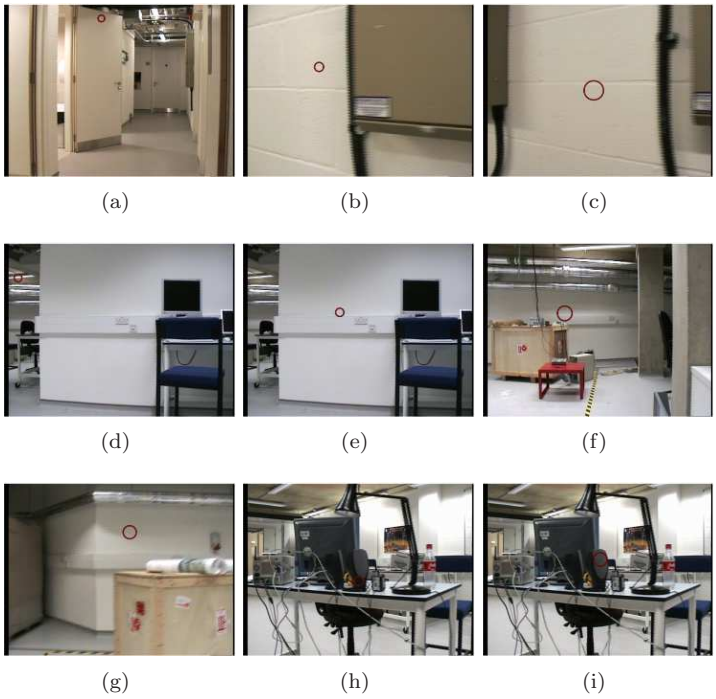


Figure 4.6: A common image descriptor (represented by a red circle) is found in multiple images from a dataset of images taken from the basement of the Oxford Information Engineering building

vocabulary, $\mathcal{V} = \{\hat{\mathbf{d}}_\mathbf{1}, \hat{\mathbf{d}}_\mathbf{2}...\hat{\mathbf{d}}_{|\mathcal{V}|}\}$.

Agglomerative clustering

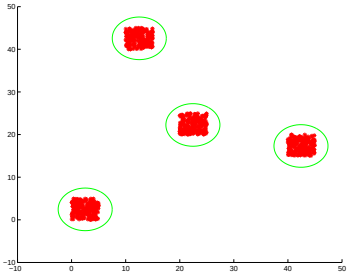
Agglomerative clustering is a technique employed to form the visual vocabulary. As an online clustering algorithm, it is adaptive and able to create new clusters with new streaming data without the need to know the number of clusters a priori. It starts with each SIFT descriptor as a singleton cluster and iteratively add to the clusters SIFT descriptors that are most similar according to a discriminant threshold, ϵ.

The agglomerative clustering algorithm is summarised as follows:

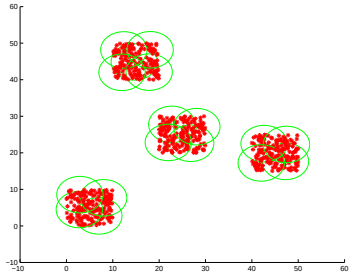
Algorithm 2 Agglomerative clustering of SIFT descriptor

while A new SIFT descriptor, d_i, is input into the database
do
  Find nearest cluster, $\hat{\mathbf{d}}_\mathbf{j}$, to SIFT descriptor, d_i
  if SIFT descriptor is within distance threshold from centre of nearest cluster $||d_i - \hat{\mathbf{d}}_\mathbf{j}|| < \epsilon$ **then**
    Add SIFT descriptor into cluster and readjust the centre of the cluster
  else if SIFT descriptor is not within a certain Euclidean distance from centre of nearest cluster **then**
    Start a new cluster with SIFT descriptor at centre of new cluster
  end if
end while

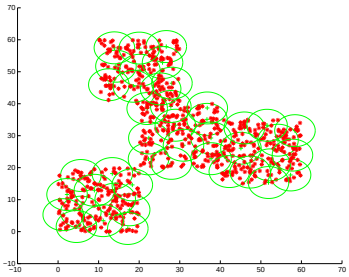
It is important to set the discriminant criterion correctly in order to reflect the true structure of groupings within the data set. Given the difficulty of visualising with 128-dimensional SIFT descriptors, Figure 4.7 illustrates the problem using 2D points. In Figure 4.7(a), the points are clustered into 4 separate distinct groupings. When a suitable threshold criteria is set, the 2D points are clustered according to their natural grouping. The points in Figure 4.7(b) are found in 16 separate distinct groups. However, the threshold criteria was set incorrectly and consequently the clusters do not reflect the true natural groupings of the points. It is very unlikely that points in a data set are always found in distinct groupings. Very often, there is no discernable boundaries between different groups as illustrated by Figure 4.7(c). In this case, the threshold criteria becomes extremely important. Different clusters are formed based on different settings of the



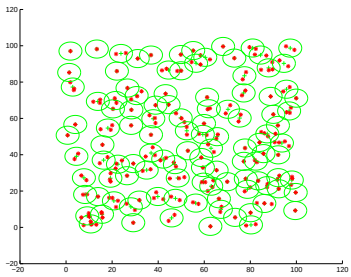
(a) 4 distinct clusters clearly defined



(b) 4 distinct clusters not well defined



(c) No distinct clusters



(d) Randomly distributed clusters

Figure 4.7: (a) shows 2D points clustered in four separate distinct groupings. (b) shows 16 clusters for four distinct groupings due to poor discriminant criteria selection. (c) shows clusters for a cloud of points with no discernible boundaries; and (d) shows clusters for randomly distributed points.

threshold criteria. In Figure 4.7(d), clustering is performed on a data set of points that are randomly distributed. It is recognised that clustering is in general an unsolved problem but the performance has been satisfactorily for the needs of our algorithm. A suitable threshold, ϵ, for the clustering algorithm has to be determined. Based on experimental settings from other visual matching implementations, a value of 300 for ϵ was selected. This setting produces a visual vocabulary of suitable size (around 6000 visual words) and has been found to produce good matching performance.

4.5.2 Assignment of weights

In existing text retrieval systems, higher weights are given to discriminative words by using the inverse document frequency (IDF) formulation. The term 'specificity', later to be known as inverse document frequency, was first proposed by Sparck Jones [56] based on the intuition that a query term that is found in many documents is not as good a discriminator as a query term that is found in only a few documents. Consequently, query terms found in many documents should be given a lower weight. Since the introduction of the IDF, it has been used in many term weighting schemes [103] coupled with term frequency, known as TF*IDF (where TF is the frequency of a term found within a document). The inverse document frequency provides an established theoretical framework from which a qualitative weight, as defined by Equation 4.7, can be given to the importance of a descriptor based on its rarity throughout the entire database:

$$W_i = \log_{10}(N/n_i) \tag{4.7}$$

where W_i is the weight given to descriptors in cluster i, N is the number of images in the database and n_i is the number of the images that contain the visual word $\hat{\mathbf{d}}_i$.

Cosine Similarity Function

A simple voting algorithm was used previously in subsection 4.3.2 to find the most similar images, given equal weights for all descriptors. Similarity between images was simply a function of the

number of putative descriptor correspondences. However, this approach invariably favours images with a greater number of descriptors and does not take into account the differences in the discriminative powers of descriptors.

As such, a different similarity comparison function is proposed. Here, the cosine similarity method is employed to measure the similarity between two images, I_u and I_v. Since each image is represented as a vector of visual words with different weights, their cosine similarity can be measured by the inner product of the two image vectors as shown in Equation 4.8. Each image, I_u, which consists of n descriptors, has become a collection of $|\mathcal{V}|$ words with different weights. If $\mathcal{V}$ contains $|\mathcal{V}|$ distinct words (clusters), a vector $\overrightarrow{I}_u = [u_1 \cdots u_{|\mathcal{V}|}]^T$ is created, where

$$u_i = \begin{cases} w_i & \text{if for } d_j \in I_u, \min\limits_{j=1:n} \mid\mid d_j - \hat{\mathbf{d}}_\mathbf{i} \mid\mid < \epsilon \\ 0 & \text{otherwise} \end{cases}$$

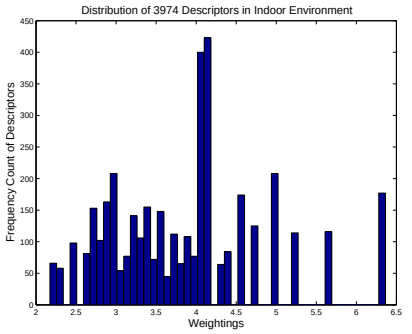
for some distance threshold ϵ (typically 300). The normalised inner product of $\overrightarrow{I}_u$ and $\overrightarrow{I}_v$ can now be used to measure the similarity, $S(I_u, I_v) \in [0, 1]$, between images I_u and I_v:

$$S(I_u, I_v) = \frac{\sum\limits_{i=1}^{|\mathcal{V}|} u_i v_i}{\sqrt{\sum\limits_{i=1}^{|\mathcal{V}|} u_i^2}\sqrt{\sum\limits_{i=1}^{|\mathcal{V}|} v_i^2}}. \tag{4.8}$$

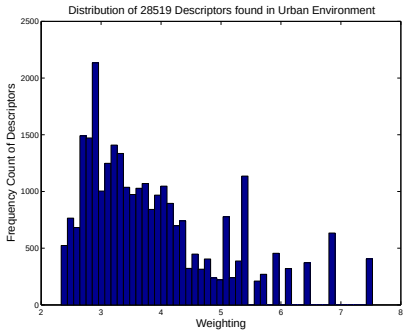
where u_i and v_i are weights for visual word $\hat{\mathbf{d}}_i$ from the respective images I_u and I_v and $|\mathcal{V}|$ is the number of visual words in the visual vocabulary.

4.5.3 Environment Context

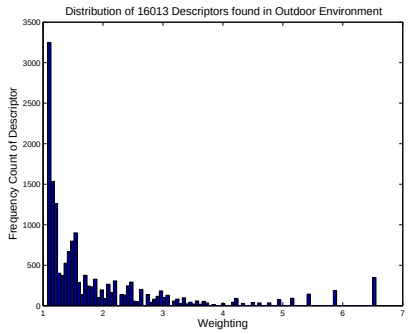
Different visual vocabularies are created for different environments owing to differences in the visual words extracted. The weight associated with a visual word varies according to the environment. A particular visual word may be commonly found in an indoor environment but may be rare in an outdoor environment. The importance of a visual word is likely to be environment-dependent.



(a) Indoor Environment



(b) Urban Environment

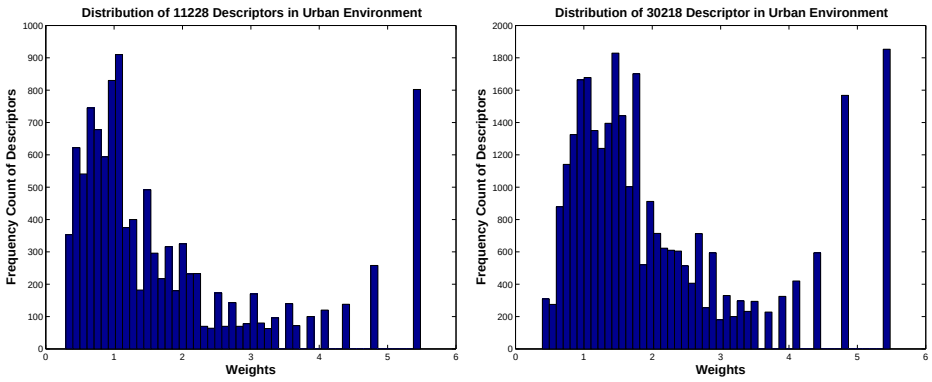


(c) Outdoor Terrain Environment

Figure 4.8: Typical distributions of visual words across different weights for various environments, classified as indoor, urban and outdoor terrain environments. The distributions vary with the type of environments.

Clustering is performed on image sequences collected from different environments, classified as indoor environment, outdoor urban environment and outdoor terrain environment. Weights are assigned to each visual word using inverse document frequency. Figure 4.8 illustrates the distribution of visual words across different weights for different environments. In Figure 4.8(a), most of the visual words are found in the middle range of $3.5 - 4.5$ for the case of an indoor environment. There are some very distinctive visual words at the higher range of $5 - 6.5$ and some very common visual word at the lower range of $2 - 3$. Figure 4.8(b) illustrates the distribution of visual words across different weights for an urban environment. Discriminative visual words with weights over 7 are found in the vocabulary. Figure 4.8(c) illustrate the distribution for an outdoor terrain environment. A high proportion of the visual words have very low weights as evidenced by the distribution. It is clear that these distributions are different for each environment. This is expected since the natural groupings of SIFT descriptors are dependent on the environment context.

4.5.4 Algorithm Context



(a) MSER (b) Harris-affine

Figure 4.9: (a) Typical distribution of visual 'words' across different weights for the MSER detection algorithm. (b) Typical distribution of visual words across different weights for the Harris-affine detection algorithm, given the same image sequence.

Even with the same image sequence, different sets of visual words are obtained by using different

detection algorithms. Using the same image sequence, two different visual vocabularies are created using the MSER and the Harris-affine detection algorithms respectively. Figure 4.9 illustrates the distributions of visual 'words' across different weight for the vocabularies. A total of 30218 descriptors were extracted by the Harris-affine algorithm from a sequence of 240 images. The descriptors are clustered into a vocabulary size of 3760 visual words. In comparison, 11228 descriptors were extracted by the MSER algorithm from the same sequence of images. The descriptors are clustered into a vocabulary size of 1262 visual words. The difference in the number of descriptors extracted can be resolved by readjusting the algorithm parameters.
The focus is on the difference in the shape of the distributions between the two vocabularies. Generally, a visual vocabulary with a higher proportion of discriminative visual words achieves a better matching performance. Based on experimentation, it was confirmed that the Harris-affine detection algorithm generally produces a visual vocabulary with a greater diversity of visual words. Consequently, most of the following experiments use the Harris-affine algorithm as the default detection algorithm. Another practical implication is having a common language for multiple robots building maps cooperatively. The same detection and description algorithms, along with exact algorithm parameters, have to be used in order to ensure that a common visual language is shared.

4.6 Improving Retrieval Efficiency

4.6.1 Inverted File Indexing

Instead of performing a single image-to-image comparison (computational complexity $O(nm)$ where n is the number of visual words in the query image and m is the number of visual words in the candidate image) for all images in the database, the comparison process for a query image can be speeded up using the *inverted file indexing* technique. During the creation of a visual vocabulary, each visual word is associated with an inverted list of pointers to images that contain that particular visual word [138], as shown in Table 4.1. Inverse document frequency formulation

No.	Visual Words	Number of Images; Images containing visual word	Weights
1	$\hat{\mathbf{d}}_\mathbf{1}$	$< n_1; I_a, I_b, I_c, I_d, \cdots >$	$w_1 = log_{10}(N/n_1)$
2	$\hat{\mathbf{d}}_\mathbf{2}$	$< n_2; I_e, I_f, I_g, I_h, I_i, I_j, I_k, \cdots >$	$w_2 = log_{10}(N/n_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\lvert\mathcal{V}\rvert$	$\hat{\mathbf{d}}_{\lvert\mathcal{V}\rvert}$	$< n_{\lvert\mathcal{V}\rvert}; I_l, I_m, I_n, \cdots >$	$w_{\lvert\mathcal{V}\rvert} = log_{10}(N/n_{\lvert\mathcal{V}\rvert})$

Table 4.1: Example Inverted List

in subsection 4.5.2 is used to assign weights to visual words.

The visual words of the query image are matched against visual words in the set of visual vocabulary. Similarity scores between images are calculated using the cosine distance method based on weights of matching visual words. Images that do not contain any matching visual words are implicitly left out of the comparison process and are assigned with a similarity score of zero. With inverted file indexing, the computational complexity of descriptor comparison for the entire database is reduced to $O(|\mathcal{V}|)$, where $|\mathcal{V}|$ is the number of visual words. Figure 4.10 illustrates how inverted file indexing and inverse document frequency formulation are integrated with a standard image retrieval system architecture.

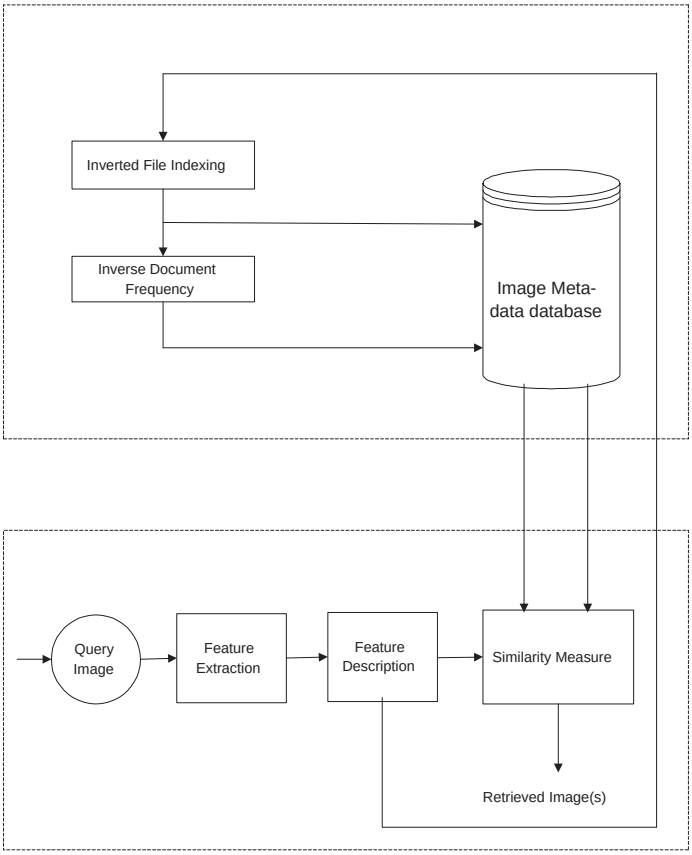


Figure 4.10: Architecture describing the framework within which the image database works. Descriptors (visual words) are extracted from images. Visual words are indexed and stored in a database. To compare an image with the image database, a vector set of visual words from the query image is compared with vector sets of visual words from other images in the database.

4.6.2 k-d Tree

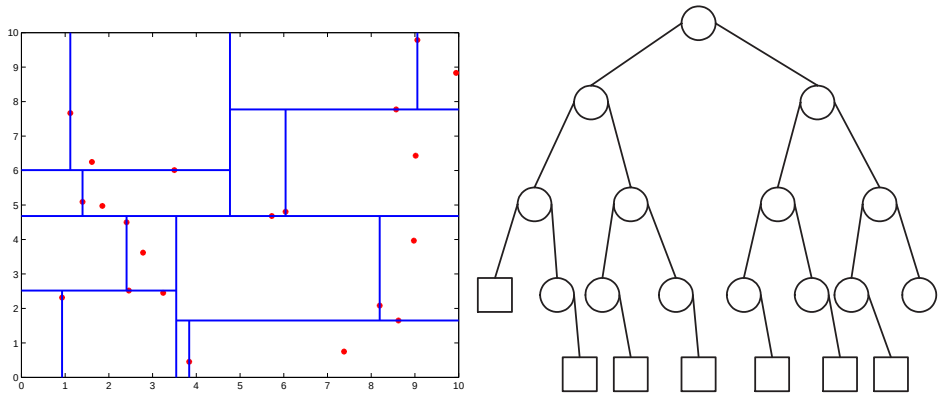


Figure 4.11: Illustration of how a two-dimensional k-d tree can be constructed from a random set of 2D points on a plane.

To further speed up image similarity comparison, a kd-tree is built from a vocabulary of visual words (represented by cluster centres). A kd-tree (where k is the dimensionality of the search tree) is a multi-dimensional binary search tree developed by Bentley [7] as a data structure for storage of information to be retrieved by associative searching. Each node of a kd-tree is a visual word. Associated with each node is a key (cluster centre), two pointers which can be either null or point to another node, and a discriminator. The search for matching visual words can be carried out with an kd-tree with an adaption for approximate nearest neighbour search in a manner similar to that previously reported [112][5]. The technique searches the tree by ranking nodes (visual words) in the tree by their distances from from the query descriptor and searching from the closest nodes first. This increases the chances of finding the true nearest neighbour earlier.

Figure 4.11 illustrates how a 2D k-d tree can be constructed from a random set of 2D points on a plane. The 2D points are represented as crosses. A line that cuts across a cross divides the 2D points associated with each node into approximately two equal-sized parts. This concept extends to multiple dimensions, such as the 128 dimensions of a SIFT descriptor. This reduces the computational complexity of finding a visual word match from $O(|\mathcal{V}|)$ to $O(\log(|\mathcal{V}|))$. This allows a quantised input image to be compared against the visual vocabulary quickly. Figure 4.12 plots the

search time as a function of the size of the image database, which shows that the time taken to complete an image comparison in a database increases logarithmically with the number of images stored in the database.

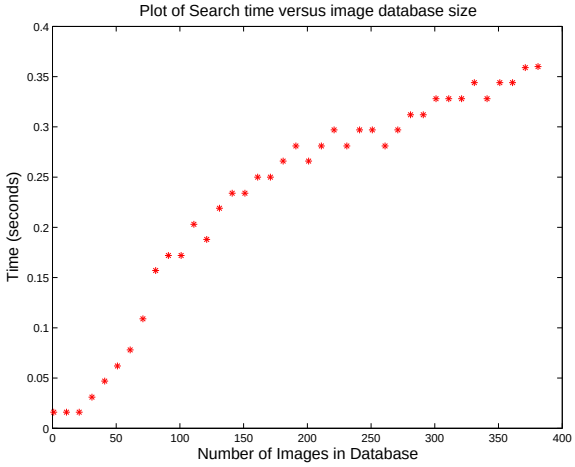


Figure 4.12: Search time versus Image Database size.

4.7 Summary

This chapter has demonstrated how an image retrieval system can be used to detect loop closing events with a delayed-state EKF. Improvements to the image retrieval system are implemented by the adoption of text retrieval techniques. An image is considered as a document of visual words. A visual vocabulary is created from the clustering of descriptors in the database. Weight is assigned to each visual word based on its rarity within the database. Similarity between images is measured using the cosine distance function. Retrieval speed is improved through the implementation of inverted file indexing and a k-d tree. As previously discussed in subsection 2.5, image retrieval systems have been employed in robot localisation applications [67, 132, 133]. An image retrieval system provides a useful framework for local scene comparison, but does not adequately resolve the issue of perceptual aliasing. Building on the framework of a retrieval system, two approaches are proposed in Chapters 5 and 6 to directly tackle the problem of perceptual aliasing.

Chapter 5

Scene Sequence for Loop Closure Detection

5.1 Introduction

The role of sequence detection is investigated in this chapter. Section 5.2 explains the motivation for using sequences to tackle the perceptual aliasing problem. Section 5.3 introduces the notion of a similarity matrix that encodes pairwise similarity between local scenes. It is highlighted how loop closure events or map intersections manifest as bright off-diagonals in a similarity matrix. Section 5.4 describes a sequence detection algorithm that can extract such off-diagonals. Section 5.5 applies the algorithm on similarity matrices to detect loop closing events. Section 5.6 looks into how multi-modal sensing can be incorporated into a similarity matrix framework to assist detection of loop closure. The role of scene appearance in a cooperative map joining application is demonstrated through the application of the sequence detection algorithm in Section 5.7. Section 5.8 concludes the chapter.

5.2 Motivation and Background

The problem of perceptual aliasing is tackled by exploiting the topological relationships of local scenes in a sequence. Instead of detecting matching pairs of observations associated with local

scenes, matching *sequences* of observations are detected [49, 50]. The motivation for detecting matching sequences of observations is as follows: It is possible that an environment contains multiple local scenes that are mutually similar. For example, multiple, identical doors may be found along an office corridor. The idea is that a sequence of similar observations corresponding to a true loop closing or map intersection (overlap between local maps) will accumulate enough evidence to make it distinguishable from other false, shorter sequences as a robot continues to retrace its previous route. This is almost equivalent to the concept of comparing a larger area to prevent false positives. However, a predetermined sequence length is not required for the proposed technique. The loop closing detection problem is posed as finding two subsequences in S^k, $\mathcal{A} = \{a_1, a_2, \cdots\}$ and $\mathcal{B} = \{b_1, b_2, \cdots\}$ where a_i and b_i are index variables, whose overall similarity strongly suggest a robot is revisiting a region.

Taking advantage of sequence order is not a novel concept. Robust alignment of images has been achieved by exploiting the temporal relationships of images in a sequence [12, 13]. In a robot localisation implementation [130, 65], temporal relationships between images were considered through the Hidden Markov Model, when trying to localise a mobile robot in a set of pre-classified locations such as offices, corridors, kitchens, etc. Taking into consideration the temporal relationships between the images helps prevent misclassifications due to dynamic changes and inherent appearance ambiguities. However, the approach required supervised partitioning of subsets of images into select locations after the initial exploration stage.

To adopt a sequence detection approach, certain assumptions are made.

- The sequence order of observations in an observation set, $S^k = \{S_1, \cdots, S_k\}$, is assumed to represent the topological relationship between local scenes.
- The local scenes are assumed to be not all identical to each other. This is a fair assumption since a highly repetitive environment (a maze-like environment) is rare. Such environments will pose a formidable challenge even to humans.
- It is assumed a robot will retrace approximately the same route it took previously during loop closing. This is often a valid assumption since most urban environments are composed

of structured paths (like corridors within a building or pathways in a town).

5.3 Similarity Matrix

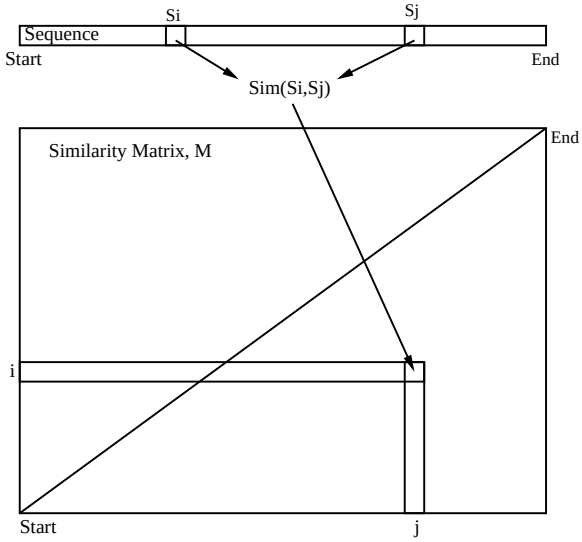


Figure 5.1: Similarity matrix constructed from pairwise comparison of a sequence of observations

This section introduces the similarity matrix, M. A similarity matrix is a representation of pairwise similarity score between all observations in a scene set. For the case of multiple-robot mapping discussed in Section 5.7, a similarity matrix is a representation of pairwise similarity score of observations between two observation sets. Each element $M_{i,j}$ is the similarity score, $Sim(S_i, S_j)$, between observation S_i and observation S_j. Figure 5.1 shows the construction of a similarity matrix from the similarity comparison of a sequence of observations. For example, each element $M_{i,j}$ of a visual similarity matrix (VSM) is the similarity score, $d_{cos}(\overrightarrow{I}_i, \overrightarrow{I}_j)$, between image I_i and image I_j.

The size of a similarity matrix grows at the rate of n^2, where n is the number of observations stored in the database. The computational complexity of updating a similarity matrix is only $O(n)$. This is because the update only involves adding the similarity scores between the latest observation and all stored observations. The similarity scores for the rest of the matrix remain

unchanged. The computational complexity in updating a VSM is further reduced to $O(log(|\mathcal{V}|))$ (where $|\mathcal{V}|$ is the size of the visual vocabulary) by utilising inverted file indexing and k-d tree techniques, as described in Section 4.6.

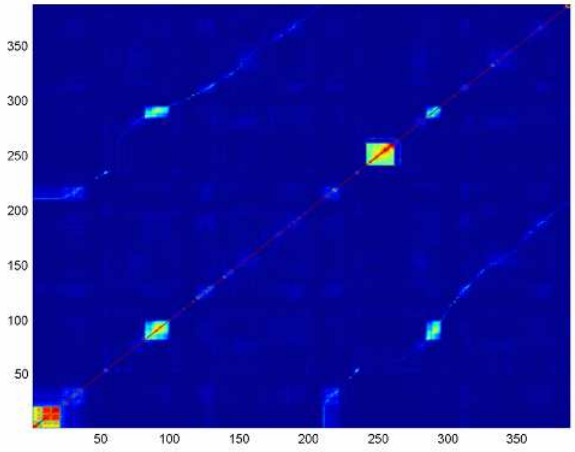


Figure 5.2: A typical similarity matrix is shown. Loop closure appears as bright off-diagonal lines. The main diagonal is a bright red line. There are regions of bright patches along the red diagonal line, representing regions where the local environments are highly similar.

A typical similarity matrix is shown in Figure 5.2. Elements with high similarity scores are coloured in bright tones while elements with low similarity scores are coloured in dark tones. Along the main diagonal, observations are matched against themselves, resulting in similarity scores of one. The main diagonal is coloured bright red. Along the main diagonal, bright "squares" can be observed. A bright square indicates there is a short sequence of mutually similar observations. For example, these observation might be images captured by a robot while traversing along a long metal fence. Consequently, these images are visually similarly to one another.
An interesting property of a similarity matrix is how loop closing events manifest themselves. When there is a loop closure, there will be a connected sequence of elements with high similarity scores. This is shown by the off-diagonal bright lines in Figure 5.2. This phenomenon of off-diagonals appearing in distance matrices when the same path is revisited and reverse-diagonals appearing when the reverse path is taken was similarly observed in [76, 112]. How these

off-diagonals can be detected will be discussed in the next section.

5.4 Finding Sequences

A modified form of the Smith-Waterman algorithm [116], which is a dynamic programming algorithm, is used to find the off-diagonals in the similarity matrix. The Smith-Waterman algorithm has been widely used to find regions of similarity between protein and nucleic acid sequences that may otherwise have little overall similarity. The shared pattern, however, may have biological significance. Given an observation set S^k, the algorithm finds two sequences $\mathcal{A} = \{a_1, a_2, \cdots\}$ and $\mathcal{B} = \{b_1, b_2, \cdots\}$ where a_i and b_i are observations, whose overall similarity strongly suggests a region of repeated pattern.

Given a similarity matrix, the algorithm proceeds to build a H-matrix. Each cell, $H_{i,j}$, is a cumulative sum of the similarity scores along an optimal sequence of moves through a similarity matrix, starting at $M_{k,l}$ (for observations a_k and b_l) and ending at $M_{i,j}$ (for observations a_i and b_j). From the start of the matrix, $H_{1,1}$, each cell of the H-matrix is calculated using dynamic programming. Three move types are possible; diagonal, horizontal and vertical. The latter two movements, though viable, are less desirable (one-to-many matching) and so have a penalty term δ associated with them. For a diagonal movement, the similarity score $Sim(a_i, b_j)$ from the similarity matrix cell, $M_{i,j}$, is added with the maximal cumulative score of the cell $H_{i-1,j-1}$ to derive the maximal score for cell $H_{i,j}$.

Moving from $H_{i-1,j-1}, H_{i,j-1}$ and $H_{i-1,j}$, $H_{i,j}$ becomes:

$$H_{i,j} = \begin{cases} H_{i-1,j-1} + M_{i,j} & \text{if } H_{i-1,j-1} \text{ is maximal,} \\ H_{i,j-1} + M_{i,j} - \delta & \text{if } H_{i,j-1} \text{ is maximal,} \\ H_{i-1,j} + M_{i,j} - \delta & \text{if } H_{i-1,j} \text{ is maximal} \\ 0 & \text{if resultant } H_{i,j} < 0 \text{ for the other options} \end{cases}$$

where "maximal" refers to the largest of $H_{i-1,j-1}, H_{i,j-1}$ and $H_{i-1,j}$.

In order for the Smith-Waterman algorithm to work, the similarity function must give a negative score when two observations are very dissimilar. In our implementation, observation pairs with a similarity score that falls below a set threshold are deemed to be dissimilar and are re-scored with a fixed negative value. This threshold value is generally selected from a predetermined percentile of all the scores in the similarity matrix. The maximum value in the H-matrix, the "maximal alignment score" $\eta_{\mathcal{A},\mathcal{B}}$, is the endpoint of a subsequence of observations with the greatest similarity. No other pair of subsequences has greater similarity.

5.5 Application to Loop Closure Detection

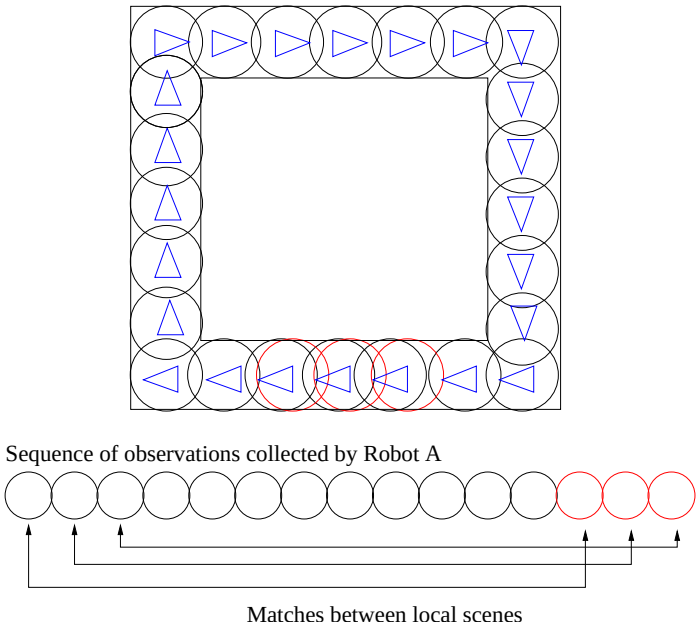


Figure 5.3: A robot travels around a building in a loop. The robot poses are represented by triangles. At each pose, an observation of the local scene is captured (represented by circles). When a robot has completed a loop, it retraces its previous route and captures similar observations (represented by red circles).

Consider a robot travelling around a building in a loop as shown in Figure 5.3. The robot poses are represented by triangles. At each pose, an observation of the local scene is captured

		a_1	a_2	a_3	a_4	a_5	a_6
	a_6	0.88	-2	<u>0.71</u>	-2	0.22	1
	a_5	-2	<u>0.65</u>	0.25	-2	1	0.22
M	a_4	<u>0.64</u>	0.21	0.37	1	-2	-2
	a_3	-2	0.23	1	0.37	0.25	<u>0.71</u>
	a_2	-2	1	0.23	0.21	<u>0.65</u>	-2
	a_1	1	-1	-2	<u>0.64</u>	-2	0.88

		a_1	a_2	a_3	a_4	a_5	a_6
	a_6	0.88	0	<u>2.0</u>	0	0.22	0
	a_5	0	<u>1.29</u>	1.44	0	0	0
H	a_4	<u>0.64</u>	0.75	1.02	0	0	0
	a_3	0	0.23	0	0	0	0
	a_2	0	0	0	0	0	0
	a_1	0	0	0	0	0	0

The top matrix is an example of a simple similarity matrix where each cell $M_{i,j}$ is the similarity score between the element i and j. Cells below a threshold (0.1) are re-scored to -2. Below is the corresponding H-matrix calculated from the lower triangular matrix of the similarity matrix shown above. A penalty (δ) of 0.1 has been used for this example. The sequence selected is underlined.

Table 5.1: Sequence extraction from a similarity matrix.

(represented by circles). When a robot has completed a loop, it retraces its previous trajectory and captures similar observations (represented by red circles). Consequently, the pattern within the sequence of observations starts repeating. Given a sequence such that $A^n = a_1, a_2, \cdots, a_n$, the similarity function $Sim(a_i, a_j)$ gives a similarity score between sequence elements a_i and a_j. Observation pairs with a similarity score that falls below a set threshold are deemed to be dissimilar and are rescored with a fixed negative value.

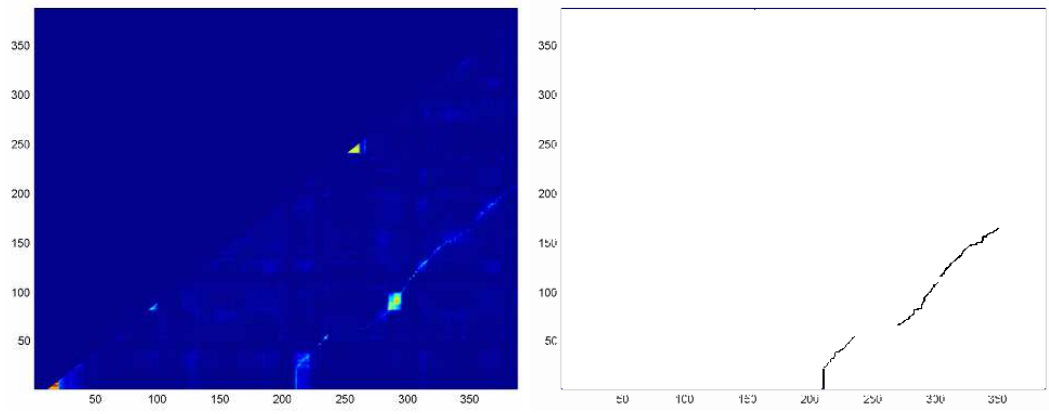
To find a pair of subsequences of observation with a high degree of similarity, a matrix H is constructed. Since the similarity matrix is symmetric, only the lower triangular matrix (LTM) of the similarity matrix is worked upon as shown in Figure 5.4, excluding the main diagonal. For practical implementation, a band of elements close to the main diagonal is masked out. This is equivalent as not looking for loop closure with observations captured at locations that are less than a short distance away. Small loop closure can be easily handled with existing SLAM techniques (eg. nearest neighbour gating).

From the H-matrix at the bottom of Table 5.1, the maximal alignment score, $\eta_{\mathcal{A},\mathcal{B}}$, is found at $H(a_6, a_3)$, which is an accumulation of similarity scores from $M(a_4, a_1)$ to $M(a_6, a_3)$ (subsequence of underlined elements). To take into account the fact that the robot might have traversed through the same area in opposite directions, the row order of the similarity matrix is reversed and the algorithm is repeated for the new matrix order. The larger of the two maximal alignment scores is chosen. To determine which observations have contributed to the maximal alignment score, the algorithm stores a pointer at each cell in the H-matrix, to indicate which previous cell contributed to its value. From the matrix cell of the H-matrix with the maximal alignment score, the path of the other matrix cells that contributed to this maximum value is sequentially traced back. This yields the best matching pair of subsequences.

5.5.1 Results

Figure 5.4(a) shows the LTM of the VSM shown in Figure 5.2. This VSM is constructed from an image sequence collected from two passes around a 1963 tower block, the Thom building (See Appendix C-1). As expected, it can be observed that there is a sequence of connected, high scoring elements within the LTM that indicates loop closure. The goal is for the sequence detection algorithm to pick out this sequence of high scoring elements. Figure 5.4(b) shows the results of applying the sequence detection algorithm. Three significant sequences, (represented as black-coloured curves) were found instead of one single, significant sequence. The breaks in what should be one sequence are caused by local scenes not being well re-observed. Various factors could have caused this. One factor could be extreme changes in perspective when re-observing the same local scene. Another factor could be dynamic changes. The local scene might have been changed structurally or dynamic objects such as humans or robots might have been inserted into the local scene.

This highlights a potential shortfall of this algorithm. If small breaks occur along different junctions of a sequence, a long sequence will be broken into many small sequences. Each small sequence will be too short to accumulate a sufficient maximal alignment score in order to be



(a) (b)

Figure 5.4: (a) shows a LTM of a VSM. A band of elements close to the main diagonal has been masked out. Loop closure appears as an off-diagonal. (b) shows the result of applying the sequence detection algorithm to find significant local alignments.

significant. Consequently, an otherwise significant sequence will not be detected if many local scenes are not well re-observed. Therefore, considerable emphasis has been placed on dealing with the perceptual variability problem. A potential solution is to enable the algorithm to accept small breaks in between sequences. However, this approach has not been investigated within the scope of this research.

Another point of interest is how the sequence detection performs in visually ambiguous regions (VARs) along the loop closing sequence. VARs are shown in Figure 5.4 as bright squares. For the bright square at the start of the loop closing sequence, it can be observed in this particular example that the sequence detection algorithm manages to pick out the higher scoring elements within the bright squares despite penalisation for one-to-many image matching. This is because the similarity scores for these elements are higher than those of the surrounding elements. For the bright square in the middle of the loop closing sequence, it can be observed that the algorithm picked out elements that cut almost diagonally through the square. This is because the similarity scores of elements within the squares are all about the same. Figure 5.5 shows some of the actual images involved in the significant match sequences as shown in Figure 5.4(b).

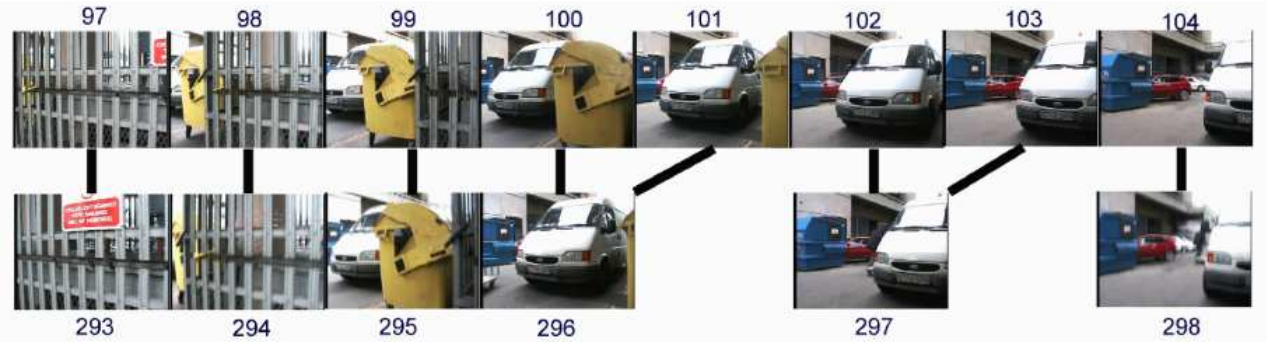


Figure 5.5: Two matching sequences $\mathcal{A}$ and $\mathcal{B}$ annotated with time step. Note the occasional one-to-two pairings which correspond to vertical and horizontal moves through the VSM.

5.6 Multi-modal Sensing

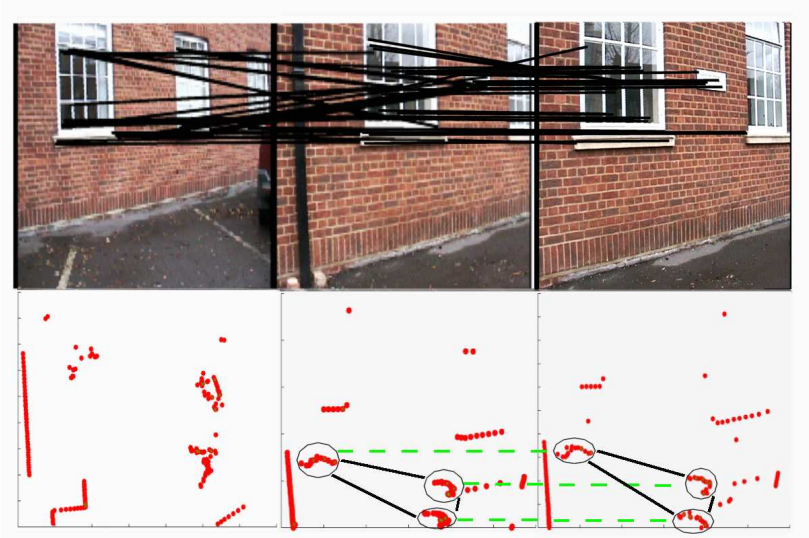


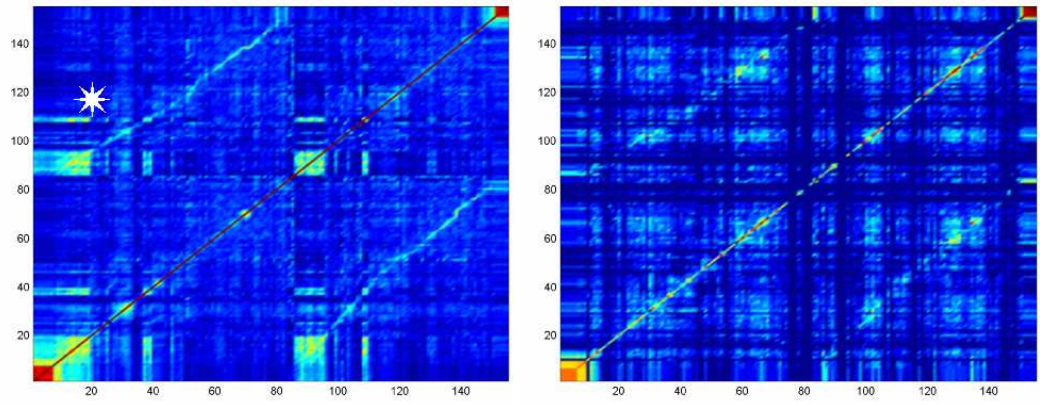
Figure 5.6: The query image and patch is in the middle and a correct match (spatially and visually) is shown on the right. A false positive loop closure is signalled with respect to the left-hand image when only visual information is taken into consideration. Black lines link matching features across images. This is discounted when spatial descriptors are used in addition.

The sequence detection algorithm can be applied on a spatial similarity matrix (SSM) as for a VSM. In this section, the feasibility of using the sequence detection algorithm on a SSM as well as for a combined similarity matrix is demonstrated. A combined similarity matrix is a matrix

representation of pairwise similarity scores obtained from similarity comparison of multiple, heterogeneous sensory observations from local scenes. The motivation for having a combined similarity matrix is as follows. Multi-modal sensing naturally leads to more discriminative descriptors for each local scene [48]. It may be possible to mitigate to a certain extent the problem of perceptual aliasing within an environment by incorporating heterogeneous sensory descriptors. The left-hand column of Figure 5.6 illustrates an anomaly where the matched visual scene is visually similar to the query but the robot is actually at a different location. This is an example where the visual image matching system is working as hoped yet it incorrectly suggests a loop closure event. It is true that some of the false image feature correspondences can be removed through the enforcement of epipolar constraints. However, matching of image features on repetitive artefacts will still occur. On the other hand, the geometries of the local scenes are truly different and can help to differentiate the two local scenes.

5.6.1 Results

In this case, a combined similarity matrix is a matrix representation of similarity scores between local scenes, which are described by a set of visual and spatial descriptors [50]. In this experiment, an ATRV-Jr robot was driven around a carpark in front of a building in two loops. An image and a laser scan were captured for every 0.5m and 30^o change in heading. A visual similarity matrix is constructed as shown in Figure 5.7(a). It is composed of similarity scores calculated from matching each image in the database against every other image in the database. A spatial similarity matrix, as shown in Figure 5.7(b), is also constructed by matching each laser scan in the database against every other laser scan with the technique described in Section 3.5. To construct the combined visual and spatial similarity matrix as shown in Figure 5.8, the normalised similarity scores from the visual and spatial similarity matrices are simply added in a similar fashion as [55].
As soon as the second loop begins (around the 85th pose) off-diagonals start appearing, indicating matches with earlier data. In the VSM shown in Figure 5.7(a), the off-diagonals are for the most part well defined. However there are some large blurred off-diagonal patches that stem from VARs.



(a) Visual Similarity Matrix (b) Spatial Similarity Matrix

Figure 5.7: (a) VSM constructed from an image sequence. High similarity scores are illustrated by bright tones while low similarity scores are illustrated by dark tones. The loop closure event is marked by the onset of bright off-diagonals. Notice a second potential loop closure is found around image 110 (highlighted by an asterisk). This is actually a false loop closure caused by repetitive visual patterns within the environment. (b) SSM constructed for a sequence of laser scans. The off-diagonals that indicate loop closure are less well-defined, reflecting the diminished certainty in matches coming from less discriminative (relative to visual images) data.

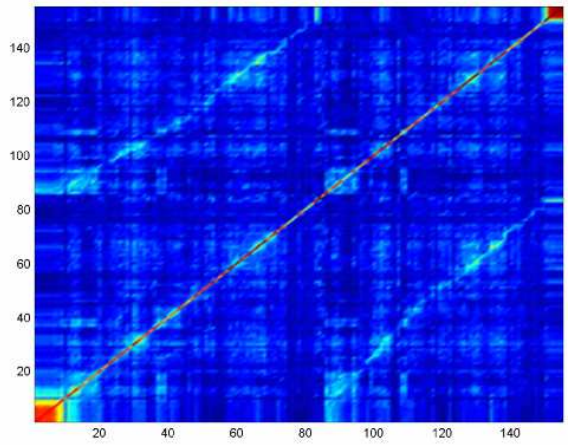


Figure 5.8: A combined similarity matrix constructed using combined similarity comparison of images and laser scans. The blurred off-diagonal regions present in the visual matrix have been reduced in magnitude leaving a clear well-defined off-diagonal trail of first loop to second loop correspondences. In particular, note how the false positive visual match highlighted with an asterisk in Figure 5.7(a) is down weighted when considered in conjunction with the local spatial appearance.

Off-diagonals are found within the spatial similarity matrix in Figure 5.7(b) but they are less defined, reflecting the diminished certainty in matches coming from less discriminative (relative to the visual images) data. Finally, Figure 5.8 shows the similarity matrix resulting from the combination of visual *and* spatial descriptions in the matching process. Importantly, the blurred off-diagonal regions present in the visual matrix have been reduced in magnitude, leaving a well-defined off-diagonal trail of first-pass to second-pass correspondences. In particular, note how the false positive visual match highlighted with an asterisk is down weighted when considered in conjunction with the local spatial appearance. A simple scene appearance is generally captured when the robot is very close to an object – a wall or parked car in our case. The dark bands appearing in all three matrices occur when a local scene is described by so few descriptors (laser or visual) that it cannot be reliably matched to other local scenes (which consist of much more descriptors), given the cosine distance function is used.

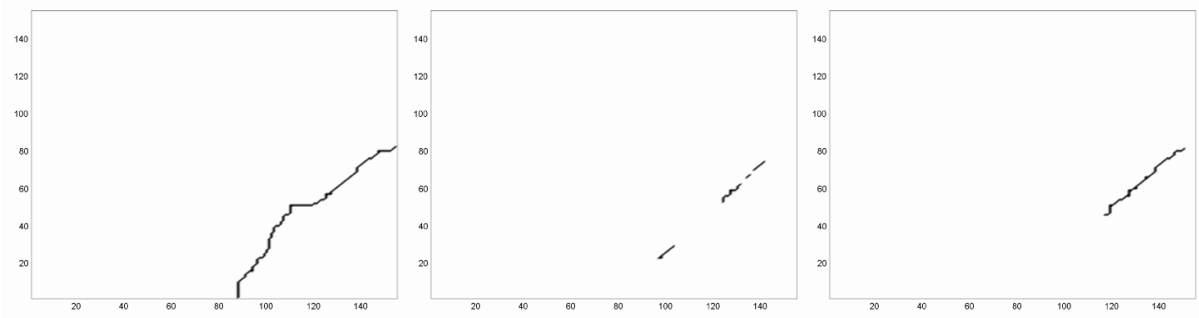


Figure 5.9: From left to right: Sequence detection results for VSM, SSM and combined similarity matrix. The dark lines represents sequences of matching pairs of observations.

From these three similarity matrices, a comparison of the sequence detection performance between using the VSM, SSM and the combined similarity matrix can be made. Figure 5.9 shows the results for the three similarity matrices. It is noted that the actual loop closure occurred from the $85th$ pose to the $155th$ pose – a sequence of 71 observations. For the visual similarity matrix, a sequence of 104 matching pairs of images was detected. For the spatial similarity matrix, the top three most significant sequences are illustrated, with the longest sequence consisting of 8 matching

pairs of laser scans. For the combined similarity matrix, a sequence of 46 matching pairs of image-laser scans is extracted.

The performances of sequence detection for the three similarity matrices are varied. A long sequence of matching images is extracted in the visual similarity matrix but the sequence consists of substantial amount of false positive matches. Although sequences detected from the spatial similarity matrix are very short, they do not contain any false positive matches. The sequence detected from the combined similarity matrix is significantly longer and, again, does not contain any false positive matches. In the light of a policy of preferring Type II errors over Type I errors (tolerating missed detections over false positives) the combined spatial/visual approach resulting in long substantial sequence of positive matches may be considered more advantageous for loop closure detection. It is noted in this example that loop closure detection is set off later for the combined similarity matrix as compared to the visual similarity matrix and spatial similarity matrix. This is due to the particular nature of the specific environment. It just happened that the later part of the loop consists of a longer connected sequence of local scenes that were well re-observed both visually and spatially.

5.7 Application to Multi-robot Mapping

The approaches of current collaborative multi-robot map building algorithms can be broadly classified into three main categories: (1) merging sensory data from multiple robots with known data association between landmarks in local maps built by different robots [31] (2) detecting other robots to determine relative position and orientation between local maps [34, 64] or assuming relative poses between robots are known [123] (3) deriving the transformation between robots' coordinate systems through the matching of landmarks [21, 126, 53]. Generally, algorithms with strong assumptions about known data association or relative poses have been limited to theoretical experiments or highly engineered experiments. The algorithms that have worked with real world data on weaker assumptions have been limited to those that rely on detection of other robots. This approach means that the robots might duplicate each other's work when exploring the same

environment for long periods of time without being aware of each others' poses. Otherwise, the robots have to hypothesize their relative positions and try to congregate at a hypothesized meeting point [64]. This allows the robots to determine accurately each others' relative poses but distracts them from the task of exploration. A more exploration-efficient way of joining local maps is to detect map intersections, independently of coordinate frames, and then align the maps.

Data association is an infrequently considered problem in multi-robot mapping. An attempt to address this problem was made by introducing an algorithm that aligned local maps into a global map by a tree-based algorithm for searching similar-looking landmark configurations [126]. The landmark configuration consists of relative distances and angle between a triplet of adjacent landmarks. Another landmark-based algorithm for map matching combined topological maps of indoor environments [21]. Landmarks such as corners, T-junctions, ends-of-corridor and closed doors were stored in the search space for correspondences. However, spatial configuration of three landmarks or simple geometric primitives are not by themselves very discriminative features, that can enable distinction of different location without any need for pose estimates.

Alternatively, a vision-based approach was used to combine maps built by a team of robots in the same worksite [44]. Images described by colour histograms are compared against each other to find the best matching image pairs. In the experimental setup, only images of planar surfaces are captured. Therefore, an inter-image homography can be calculated for selected image pairs. If the homography is supported by a sufficiently high number of corners, intersection is found and robot paths can be registered with respect to one another. However, the use of a single image pair for matching is prone to false positives (hence the motivation for using sequences). Importantly, none of the algorithms described above have any mechanism to determine that two local maps have *no* common overlap. They simply find the 'best' alignment possible between two maps.

Local maps built by distributed robots may have little overall similarity due to mapping of different areas but there may be overlap between their mapped environments, termed as "map intersection" in this work. A map intersection between two local maps can be used to determine the relative transformation between the maps and, consequently, to align the maps together. Finding a

matching pair of subsequences of observations between the robots' observation sequences is a strong indicator of overlap between their maps. The detection of such subsequences of observations is therefore the precursor of map joining. Map intersection detection can be considered to be a loop closing problem; where one robot "closes the loop" of the map built by another [49].

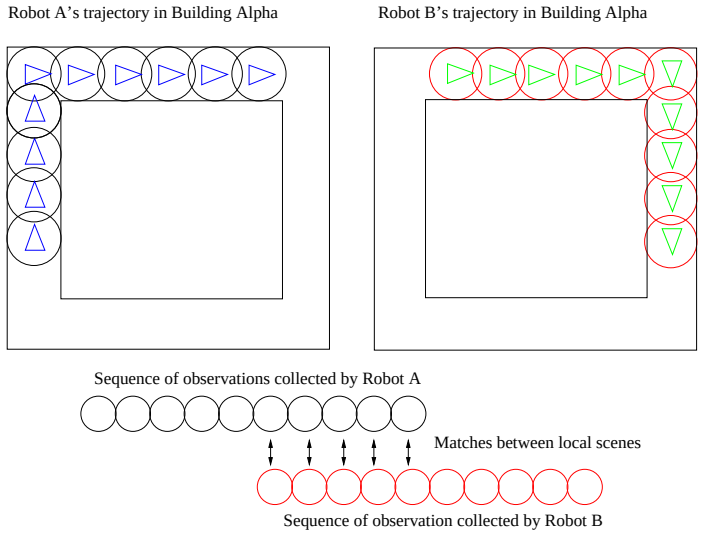


Figure 5.10: Robots A and B started exploring the same building at different start locations. Robot poses are represented by triangles. At each pose, an observation of the local scene was captured, represented as circles. As the robots continued to explore, an overlap between the environments they have explored occurs. As such, there will be a pair of matching subsequences between the robots' sequences of observations.

Let's consider a scenario where robot A and robot B, started out exploring different parts of the same building, as illustrated by Figure 5.10. At each pose (represented by a triangle), an observation from the local scene (represented by a circle) is captured. Each robot will collect a sequence of observations as they explore the building. If there is an overlap between the environment explored by the robots, there will be an region of similarity between the two sequences collected by the robots. This section describes how a sequence detection algorithm can be applied to the task of detecting intersections between two maps built by different robots.
A similarity matrix between two sequences of observations can be constructed. Each observation from sequence, $S^A = \{a_1, \cdots, a_n\}$, collected by robot A is compared with all observations from

sequence, $S^B = \{b_1, \cdots, b_m\}$, collected by robot B. Each element $M_{A,B}(i,j)$ of the similarity matrix is the similarity score between observation a_i and observation b_j. Consider the case where the primary sensor is a camera. A typical VSM constructed by comparing two different image sequences with an overlap is shown in Figure 5.11. When there is an intersection between the local maps of the robots, there will be a connected sequence of elements with high similarity scores found within the VSM [49]. This is shown by the bright line in Figure 5.11. The sequence detection algorithm, described in Section 5.4, is used to extract the off-diagonals that signify intersections between maps.

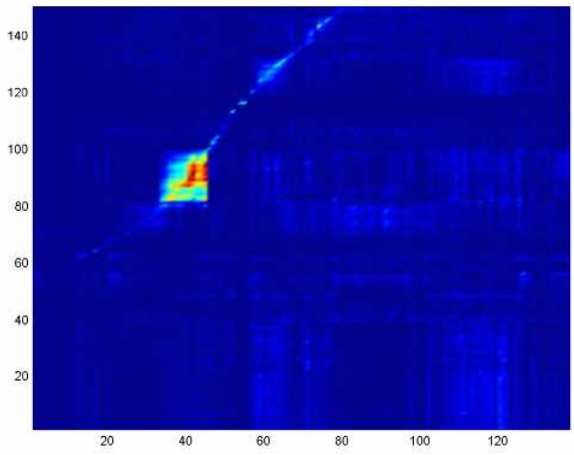


Figure 5.11: A similarity matrix constructed from the comparison of two sequences of observations collected by two robots. Elements with high similarity scores are coloured in bright red while elements with low similarity scores are coloured in dark blue. The bright line highlights the sequence of observations that are similar to each other – indicating that there is an overlap in the two environments explored. The bright square (VAR) in the similarity matrix is the result of a region of repetitive pattern in the environment such as a long fence.

From the H-matrix at the bottom of Table 5.2, the maximal alignment score, $\eta_{\mathcal{A},\mathcal{B}}$, is found at $H(a_5, b_5)$, which is an accumulation of similarity scores of the subsequence of underlined elements from $M_{a2,b2}$ to $M_{a5,b5}$. To take into account that the robots might have traversed through the same area in the opposite direction, the order of one robot's image sequence is reversed and the algorithm is repeated for that sequence order. The larger of the two maximal alignment scores is chosen. To determine which images have contributed to the maximal alignment score, the

M		b_1	b_2	b_3	b_4	b_5	b_6
	a_6	-2	0.35	-2	-2	-2	-2
	a_5	-2	-2	-2	-2	<u>0.32</u>	-2
	a_4	-2	-2	0.26	<u>0.37</u>	0.26	-2
	a_3	-2	0.21	<u>0.27</u>	<u>0.33</u>	-2	-2
	a_2	-2	<u>0.32</u>	0.25	0.18	-2	-2
	a_1	-2	-2	-2	0.15	-2	-2

H		b_1	b_2	b_3	b_4	b_5	b_6
	a_6	0	0.35	0	0	0	0
	a_5	0	0	0	0	<u>1.41</u>	0
	a_4	0	0	0.75	<u>1.09</u>	1.25	0
	a_3	0	0.43	<u>0.59</u>	<u>0.82</u>	0	0
	a_2	0	<u>0.32</u>	0.47	0.55	0	0
	a_1	0	0	0	0.15	0	0

The top matrix is an example of a simple similarity matrix where each cell $M_{ai,bj}$ is the similarity score between element a_i from sequence A and element b_j from sequence B. Cells below a threshold (0.1) are re-scored to -2. Below is the corresponding H-matrix calculated from the similarity matrix shown above. A penalty (δ) of 0.1 is used for this example. The sequence selected is underlined.

Table 5.2: Sequence extraction to detect map intersection.

algorithm stores a pointer at each cell in the H-matrix to indicate which previous cell contributed to its value. From the matrix element of H with the maximal alignment score, the path of the other matrix elements that contributed to this maximum value can be traced back. This yields the best matching pair of subsequences.

5.7.1 Results

Here visual appearance is used to detect intersections between local maps built by multiple robots. A VSM, which is composed of pairwise similarity scores between images captured by each robot, is constructed. Each element $M_{A,B}(i,j)$ in the VSM is a measure of similarity between image i from robot A and image j from robot B. Despite the highly discriminative nature of photometric information, false positives still exist because repetitive entities occur frequently in urban environments eg. windows. This work exploits the topological structure of the VSM to enhance robustness in the detection of intersections between maps. The idea is simple; by matching subsequences of images captured from topologically linked locations, the probability of false

positives is greatly reduced.

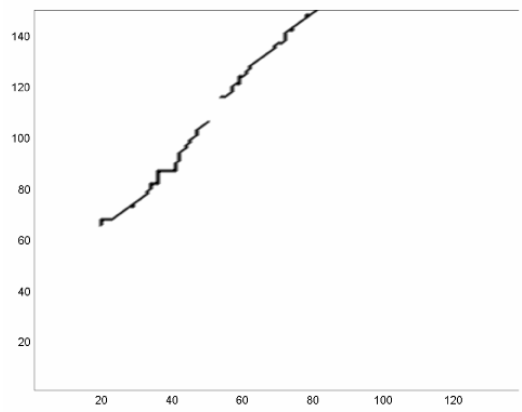


Figure 5.12: Two sequences, which indicate potential intersections between local maps compared, are detected by the sequence detection algorithm.

In our experiment, four robots start exploring from different locations of the same building. Each robot builds its own local map as shown in Figure 5.13. It can be seen that it is difficult to determine if overlaps between these maps exist based on geometric information alone. By comparing the image sequences collected by robot A and robot B, a 114 by 146 VSM is constructed. The time complexity of the sequence detection algorithm is $O(nm)$ where n and m are the lengths of the respective sequences. For the size of this particular similarity matrix, the sequence detection algorithm takes less than 0.3 second to find the optimal alignment using a Pentium 4, 2.40GHz CPU. The average time to compare one image against a sequence of 146 images is 0.269 second.

Figure 5.14 shows typical pairs of image subsequences found by the sequence detection algorithm. Since each image and laser scan is time-stamped, the portion of the local map that correspond to when the images were taken can be extracted, as shown in Figure (5.14). An estimated 2D transformation alignment between maps can then be calculated (using principal moment alignment for example) and used to bring the two geometric maps into close proximity. From here, scan matching produces accurate map-to-map transformations, allowing the four maps to be fused resulting in the map shown in Figure 5.15.

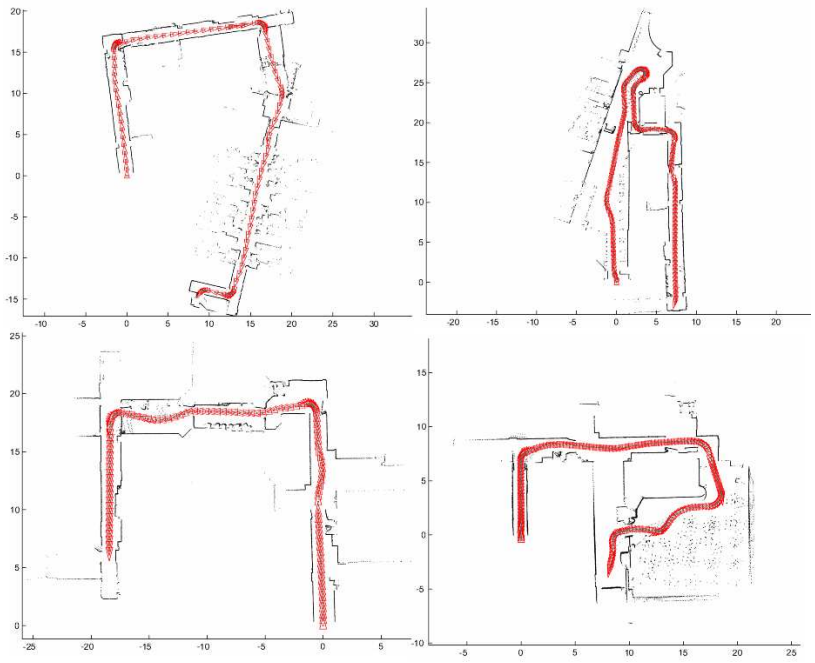


Figure 5.13: Local maps of different parts of the same building built by different robots. There is an overlap between each of the maps but it is not easy to discern from the laser patches alone.

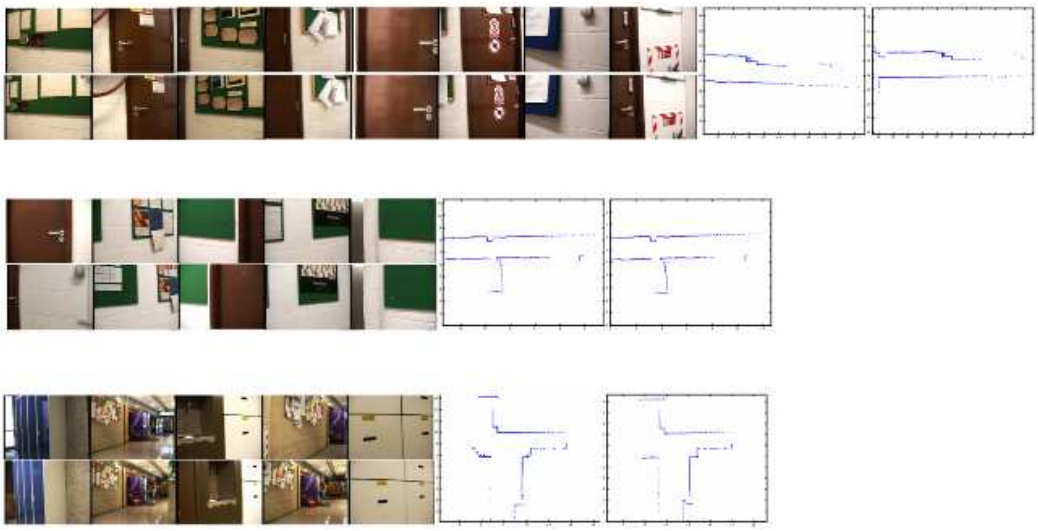


Figure 5.14: Matching subsequences between image streams gathered by different robots. The local regions in each map are shown to the right of each pairing — the first for the top sequence and the second (far right) for the second (lower) sequence.

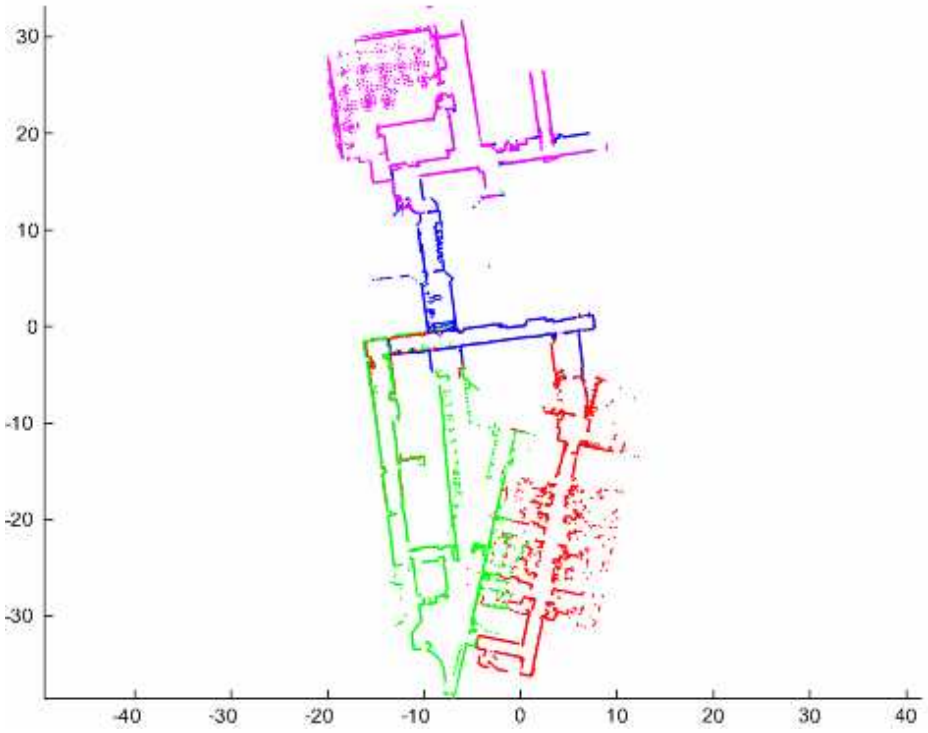


Figure 5.15: A combined map in a single coordinate frame that was formed by aligning the four local maps shown in Figure 5.13

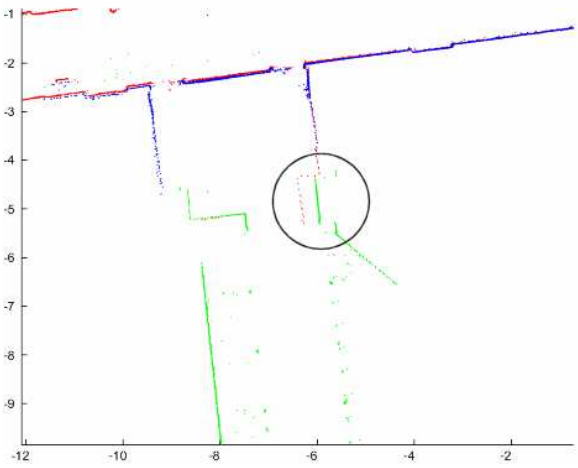


Figure 5.16:

However, the astute observer will notice that despite the overall map is properly aligned, there are small portions of the maps that are not very well-aligned as shown in Figure 5.16. This is because as a whole, there are small errors in each local map representation. Moreover, none of the local maps has made a loop. Consequently, accumulated linearisation and perception errors are still present in each local map when they are combined together. Nevertheless, the local intersections of local maps are locally aligned with each other.

5.8 Summary

In this chapter, the notion of a similarity matrix that encodes pairwise similarity relationships between all local scenes is introduced. A sequence detection algorithm to tackle the problem of perceptual aliasing by exploiting topological links between local scenes in a similarity matrix has been described. In contrast to detecting loop closure with a single match via an image retrieval system, loop closure is now triggered by detecting a connected sequence of images within the similarity matrix. Crucially, the algorithm remains independent of robot pose estimate. The algorithm is used to detect off-diagonals in similarity matrices that indicate loop closing events. The algorithm is test on varied similarity matrices which are constructed from similarity comparison of multiple, heterogeneous sensory observations. The concept of employing the sequence detection algorithm for a map joining application is demonstrated. Intersections between local maps built by different robots are detected by using the algorithm to detect off-diagonals. However, the assumption that a robot will retrace its previous route poses certain limitations. A robot may not detect a loop closure if its route intersects perpendicularly with its previous path. A more detailed analysis of the limitations of the algorithm is presented in Chapter 7.

Chapter 6

Ambiguity Management

6.1 Introduction

This chapter deals with the management of ambiguity when making loop closing decisions. It describes how spectral decomposition of a similarity matrix can help to remove the effects of ambiguous artefacts and, as such, improve the sequence detection performance. Section 6.2 presents the problem of perceptual aliasing in the context of a similarity matrix. It motivates the need for a complementary approach to manage perceptual ambiguity along with the sequence detection algorithm. Section 6.3 describes the decomposition of a similarity matrix by eigenvalue decomposition. Section 6.4 demonstrates how the effects of repetitive artefacts can be removed through rank reduction. Section 6.5 shows some experimental results from applying the sequence detection algorithm on rank reduced similarity matrices. A method of evaluation of the probability that a matching pair with a maximal score $\eta_{\mathcal{A},\mathcal{B}}$ could have been generated from random from a similarity matrix.

6.2 Effects of Perceptual Aliasing in Similarity Matrix

Generally, place recognition should be based on distinctive places within an environment [69], instead of locations whose appearances are common. This follows the concept that humans tend to remember distinctive places, which serves as decision points during travel [84]. Instead of relying on human intervention to designate distinctive places, our proposed approach [51] filters out effects of common artefacts within the environment before looking for loop closure. It is is non-trivial to determine if an observation is ambiguous [26]. A naive method will be to just remove mutually similar observations from any consideration. However, a scene may contain both ambiguous artefacts and globally salient[1] artefacts. A distinction between the effects of ambiguous and uncommon artefacts is needed and only the effects of ambiguous artefacts should be removed.

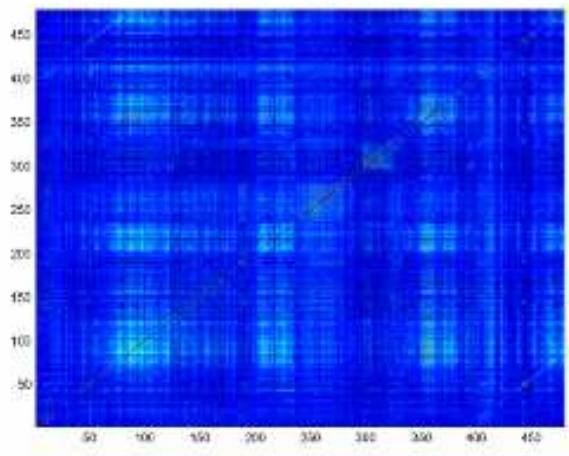


Figure 6.1: Above illustrates a VSM constructed from a sequence of images captured from a visually confusing environment. Loop closure appears as bright off-diagonal lines. There are many regions of bright patches across the similarity matrix, representing regions where the local environments are highly visually similar.

Figure 6.1 is an illustration of the problem of perceptual aliasing that manifests itself as visually ambiguous regions (VARs) within a VSM. VARs are shown as bright square regions. VARs occur when a sequence of images contains a subsequence of mutually similar images. It is a potential

[1] An artefact is globally salient if the artefact is uncommon within the environment.

case in which the sequence detection algorithm might not work optimally. The problem is that these mutually similar local scenes are topologically close together. Topological links between local scenes cannot be exploited effectively in such a scenario. The concern now is that the VARs will prompt incorrect loop closures. False loop closures are a real disaster for SLAM systems, leading to catastrophic map damage and "lost" vehicles. The use of spectral decomposition to tackle the perceptual aliasing problem in a similarity matrix is investigated in the following sections.

6.3 Eigenvalue Decomposition of Similarity Matrix

The use of eigenvalue decomposition (EVD) [38, 120], also known as Karhunen-Loève expansion and principal component analysis [134], to remove the effects of ambiguous artefacts within a similarity matrix is described here. In a similar fashion, singular value decomposition (SVD) has been used to take advantage of implicit semantic structure in the association of terms with documents [22] while object categories within a set of unlabeled images are discovered through probabilistic latent semantic analysis [113]. SVD was similarly used to summarise video content [39] and used to segment media content such as video and music [33].

A real symmetric matrix has the following properties; the eigenvalues are real, the eigenvectors are orthogonal and the matrix is diagonalisable. For a $n \times n$ matrix, M, which is symmetric and real, there is an orthogonal matrix V and a diagonal matrix Σ such that $M = V\Sigma V^T$. The columns of V are eigenvectors, $v_1, \cdots, v_m$ of M and form an orthonormal basis for R^m ($m = n$ only if M has full rank). The diagonal entries of Σ are the eigenvalues, $\lambda_1, \cdots, \lambda_m$ of M. As such, a symmetric similarity matrix, M, can be expressed as follows:

$$
\begin{aligned}
M = & \quad V\Sigma V^T \\
= & \quad [v_1 \cdots v_m] \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_m^T \end{bmatrix} \\
= & \quad \sum_{i=1}^{m} v_i \lambda_i v_i^T
\end{aligned}
$$

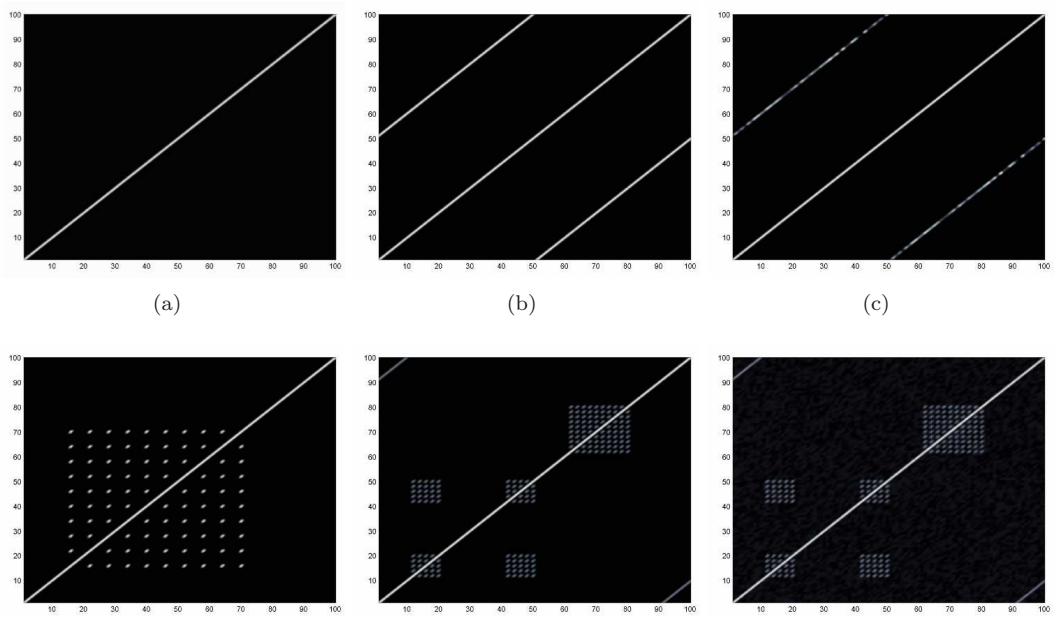
where v_i is the i^{th} column vector of V, λ_i is the ith diagonal entry of D, $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_m \geq 0$ and the rank of M is equal to m. The outer product expansion form for the eigenvalue decomposition, $M = \sum v_i \lambda_i v_i^T$, expresses M as a sum of rank one matrices $M_i^{\ominus} = v_i \lambda_i v_i^T$.

An interesting property of a symmetric matrix is that the sum of the m eigenvalues equals the trace, the sum of diagonal entries of the matrix. For the particular case of a similarity matrix, the trace is also equal to the number of images in the database. This is because the main diagonal is made up of every images in the database matching against themselves, each giving a similarity score of one. Therefore, the sum of m eigenvalues of the similarity matrix will equal the number of images in the database.

6.3.1 Synthetic Similarity Matrices

Given that a similarity matrix can be decomposed into a sum of rank one matrices, the question is how to make use of decomposition to tackle perceptual aliasing in similarity matrix. The effects of decomposition on synthetic similarity matrices are investigated in this subsection. In Figure 6.2, six different synthetic similarity matrices representing are shown. Each of these matrices represent different simulated environments. Figure 6.2(a) shows a synthetic similarity matrix that represents a sequence of observations with no loop. It is a perfect case in which none of the observations matches with each other. Pairwise similarity score between any two different observations is zero. Figure 6.2(b) shows a similarity matrix that represents a sequence of observations collected from two passes of an environment. The first pass comprises of a subsequence of observations from local scenes $1 - 50$. The second pass comprises of a subsequence of observations from local scenes $51 - 100$. The second subsequence matches the first subsequence perfectly. In this synthetic case, observations that match are given a perfect similarity score of 1 while observations that do not match are given a similarity score of 0.

Figure 6.2(c) shows a similarity matrix that represents a sequence of observations obtained from two passes. However, the sequence of observations from local scenes $51 - 100$ in the second pass does not match perfectly with the sequence of observations in the first pass. Similarity scores were



(a) (b) (c)

(d) (e) (f)

Figure 6.2: (a) shows a synthetic similarity matrix with no loop. All observations are perfectly dissimilar. (b) shows a case with loop closure. The sequence of observations from local scenes $51 - 100$ is a repeat of the sequence $1 - 50$. Observations that match are perfectly similar while observations that do not match are perfectly dissimilar. (c) shows a case with loop closure as well. However, in this case, the sequence of observations in the second pass does not match perfectly with the sequence of observations in the first pass. (d) shows a case with no loop closure but has ambiguous regions. Matches between observations are given perfect similarity scores of 1. (e) shows a synthetic case of a similarity matrix with loop closure and ambiguous regions. Matches between observations are given a similarity score of 0.5. (f) shows a synthetic case of a similarity matrix that is similar to (e) but with added noise. Values from $0 - 0.1$ are added randomly to similarity scores.

randomly assigned from $0 - 1$ to simulate differences in scene appearance due to errors in measurements, perspective distortion and occlusions. Figure 6.2(d) shows a similarity matrix that represents a sequence with no loop closure but with one ambiguous region. Local scenes associated with this ambiguous region are given perfect similarity scores of 1. Figure 6.2(e) shows a similarity matrix with loop closure and multiple ambiguous regions. Matches between observations are all given a similarity score of 0.5. Figure 6.2(f) shows a synthetic case of a similarity matrix that is similar to Figure 6.2(e) but with added noise. Values from $0 - 0.1$ are added randomly to similarity scores.

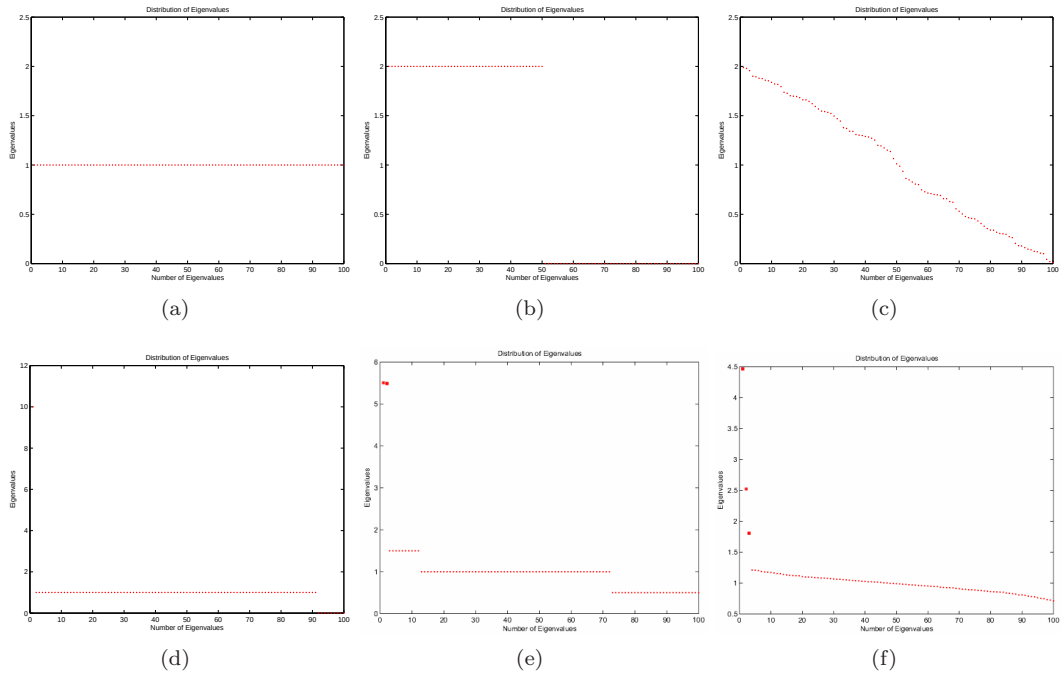


Figure 6.3: (a-f) are the distributions of eigenvalues obtained from decomposition of corresponding similarity matrix shown in Figure 6.2(a-f).

The six synthetic similarity matrices shown in Figure 6.2 are decomposed using EVD. Each matrix is decomposed into a set of orthogonal eigenvectors and a corresponding set of eigenvalues. The set of eigenvalues for each matrix is represented as a plot graph Figure 6.3. It appears that large eigenvalues and their corresponding eigenvectors may be associated with ambiguous regions within

a similarity matrix. To investigate if such a relationship does exist, the hypothesis is tested on similarity matrices constructed from two separate datasets. The datasets can be described as follows:

- An image sequence containing a subsequences of images that are captured when a robot is stationary.
- An image sequence containing two subsequences of similar images that correspond to different locations within an environment. This is a consequence of the nature of the environment where there are repetitive visual patterns.

6.3.2 Experiment A: Removing the effects of VARs

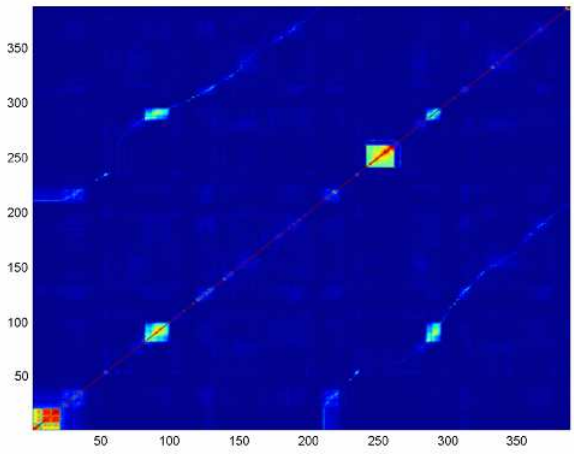
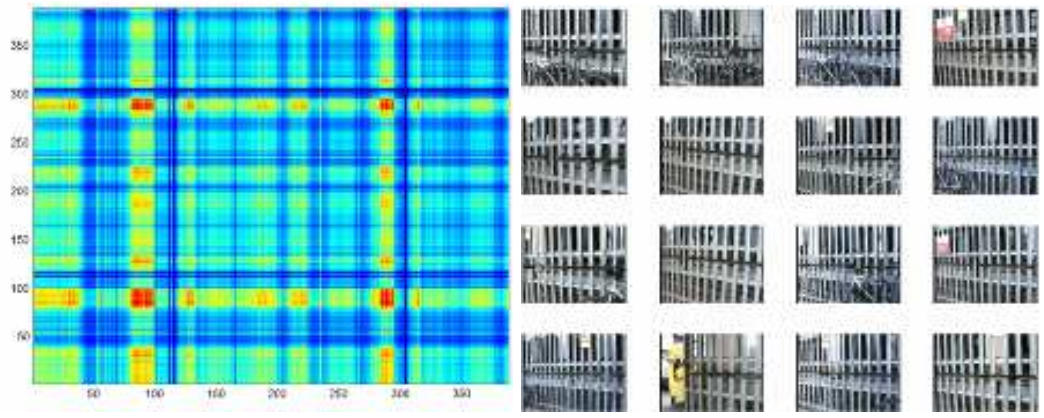


Figure 6.4: (a) shows a VSM with limited VARs. Visually ambiguous regions are the result of groups of identical images.

So far, decomposition of a similarity matrix has been described in abstract terms using synthetic data. EVD is now applied on a VSM constructed from real data. In this experiment, a robot was driven around a single building (Thom Building) twice. An image is captured for every time interval of two seconds. The camera orientation is fixed at 60^o right of the robot's heading throughout the experiment. A sequence of 349 images, $I^{Thom} = \{I_{Thom1}, \cdots, I_{Thom349}\}$, was

collected from this exploration run. See Appendix C-1. M^{Thom}, is constructed from the image sequence and is illustrated in Figure 6.4. Loop closure appears in Figure 6.4 as an off-diagonal bright line starting at $I_{Thom209}$. A thing to note about this experiment is that at certain junctures, the robot stopped for short periods of time. Consequently, highly similar images were captured within these periods. These groups of images manifest themselves as VARs (bright square regions). This can be easily avoided if images were captured for a fixed distance interval instead of a time interval. However, it is not difficult to imagine environments, in which self-repeating patterns occur frequently, exist. For this particular nature of the environment, the resultant similarity matrix has limited VARs.

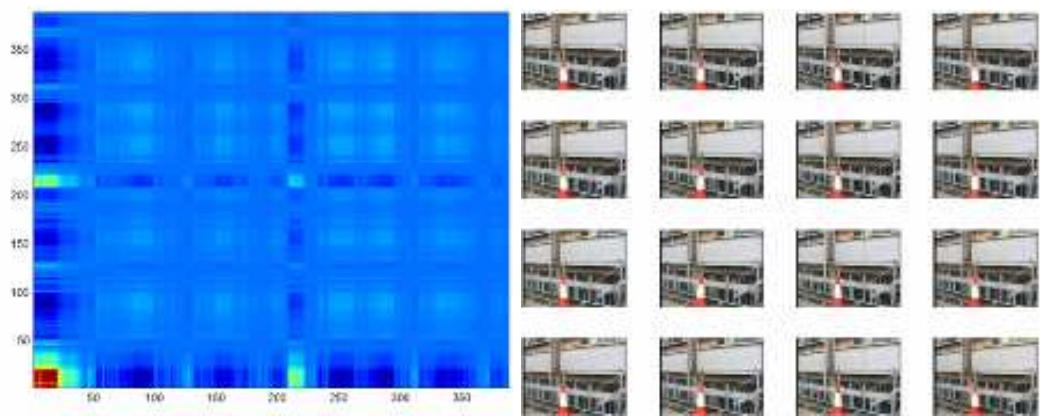


(a) (b)

Figure 6.5: (a) shows $M_1^{Thom} = v_1 \lambda_1 v_1^T$ of the VSM shown in Figure 6.4. (b) shows 16 images associated with high scoring cells in the matrix. These are mostly images of metal fences with sharp vertical edges. The bright regions are mostly concentrated in four regions (red squares).

The outer product expansion form for the eigenvalue decomposition (EVD), $M = \sum v_i \lambda_i v_i^T$, $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_k \geq 0$, expresses M as a sum of rank one matrices $M_i^{\ominus} = v_i \lambda_i v_i^T$. Here, the nature of the principal rank one matrices is investigated. Figure 6.5(a) shows the first rank one matrix, $M_1^{Thom} = v_1 \lambda_1 v_1^T$, of the VSM shown in Figure 6.4. Figure 6.5(b) shows sixteen images associated with high scoring cells in the rank one matrix. This grouping consists of images of metal fences with sharp vertical edges. The decomposition has extracted the dominant theme within this

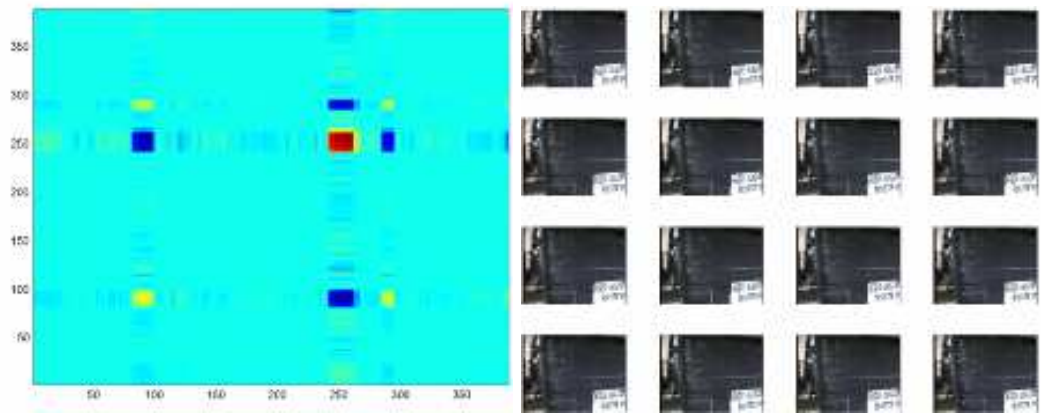
particular environment. Indeed, the surroundings of the urban building is well defined by vertical edges as witnessed by the distribution of high scoring cell along the main diagonal of the rank one matrix. These bright regions are spread across the matrix as bright coloured cells though mostly concentrated in four regions (red squares).



(a) (b)

Figure 6.6: (a) shows $M_2^{Thom} = v_2 \lambda_2 v_2^T$ of the VSM shown in Figure 6.4. (b) shows 16 images associated with high scoring cells in the matrix. These images are all the same because images were captured for a fixed time interval and the robot did not move initially when the experiment start. The bright regions are mostly concentrated at the bottom left corner of the matrix.

Figure 6.6(a) shows the second rank one matrix, $M_2^{Thom} = v_2 \lambda_2 v_2^T$. Figure 6.6(b) shows sixteen images associated with high scoring cells in the matrix. These images are almost identical because they were captured within a short time interval when the robot did not move at the start of the exploration. As such, the bright regions are mostly concentrated to a square region at the bottom left corner of the matrix. This means that this particular theme does not permeate throughout the environment but by virtue of the high scores (bright red) of cells in the region, it still constitutes as a significant theme. Figure 6.7(a) shows the third rank one matrix, $M_3^{Thom} = v_3 \lambda_3 v_3^T$. The images in Figure 6.7(b) are almost identical (images of a garbage bin) because images were captured within a short time period when the robot was stopped momentarily. The bright cells are mostly concentrated at one square region.



(a) (b)

Figure 6.7: (a) shows $M_3^{Thom} = v_3 \lambda_3 v_3^T$ of the VSM shown in figure 6.4. (b) shows 16 images associated with high scoring cells in the matrix. These images are all the same because images were captured for a fixed time interval and the robot was stopped momentarily. The bright regions are mostly concentrated at one square region.

6.3.3 Experiment B: Removing False Loop closure

Another experiment with a different set of environmental settings is considered here. In this experiment, the robot was driven around a carpark in front of a red brick wall building (Acland building) twice. The parameter settings were similar to that set in Experiment B except that camera orientation is now fixed at 60^o left of robot's heading. A sequence of 155 images, $I^{Acland} = \{I_{Acland1}, \cdots, I_{Acland155}\}$, was collected from the exploration run. This is a relatively small exploration run. The reason why this environment is selected is because the visually repetitive brick wall patterns. This brick wall pattern is separated in the middle by the entrance of the building. The left side of the building looks visually similar to the right side of the building. This 'symmetrical' appearance can potentially trigger a false loop closure.

A VSM, M^{Acland}, is constructed for this image sequence as illustrated in Figure 6.8(a). Loop closure occurs in Figure 6.8(a) as an off-diagonal bright line starting at $I_{Acland90}$. Images captured along the length of the building are visually similar. These group of similar images manifest themselves as bright square blocks within the VSM. It can be observed that there are long, thin

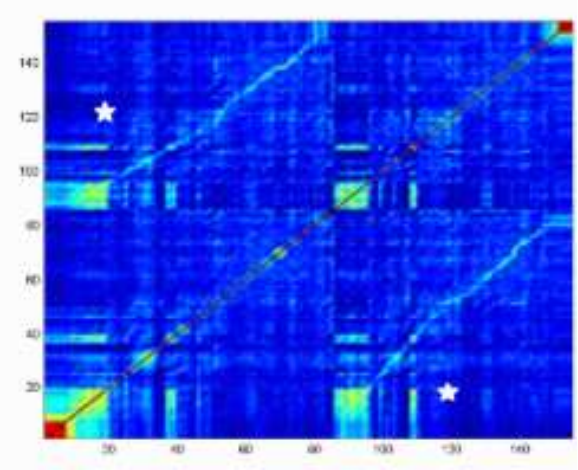
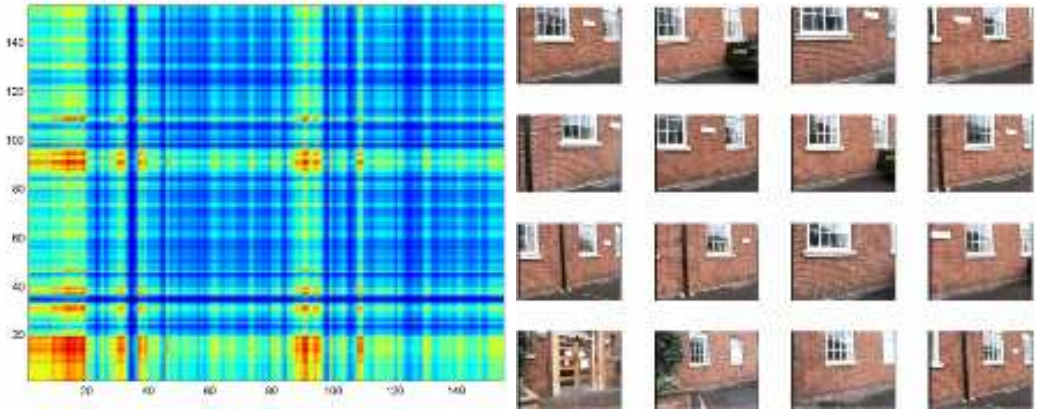


Figure 6.8: A VSM with two off-diagonals. The longer off-diagonal represents a true loop closing event and while the shorter off-diagonal represents a false loop closing event.

rectangular blocks within this similarity matrix (marked by stars). These blocks correspond to images captured from the far-side of building from where the robot started.

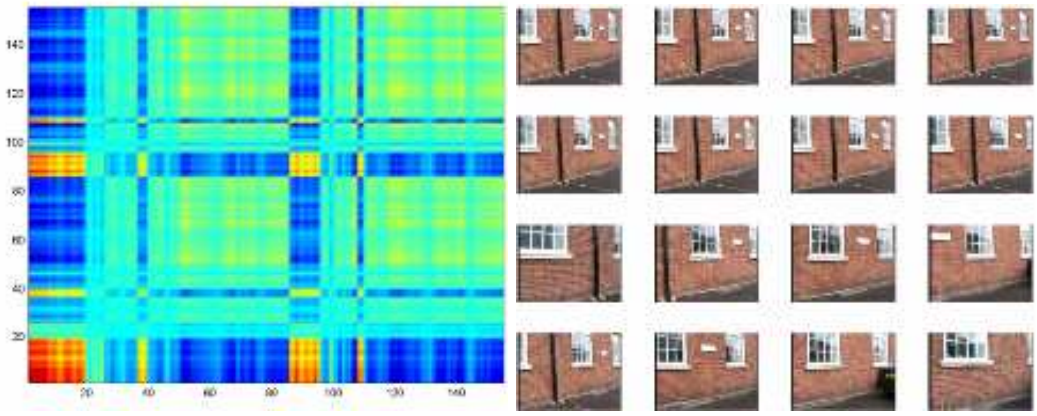
Figure 6.9(a) shows the rank one matrix, $M_1^{Acland} = v_1 \lambda_1 v_1^T$, of the VSM shown in Figure 6.8. Figure 6.9(b) shows sixteen images associated with high scoring cells in the rank one matrix. These are mostly images of the facade of the red brick wall building. In other words, the environment explored by the robot is mostly categorised by images with the brick wall pattern. These images are spread across the matrix as bright coloured cells though mostly concentrated in four regions.

Figure 6.10(a) shows the second rank one matrix, $M_2^{Acland} = v_2 \lambda_2 v_2^T$. Figure 6.10(b) shows 16 images associated with high scoring cells in the rank one matrix. Once again, these are images of the red brick wall of the building. These images are highly similar to those shown in Figure 6.9(b). The only slight distinction is that these images seem to have more emphasis on vertical edges as evident from the vertical pipe found in many of the images. The bright regions are mostly concentrated in the same regions as Figure 6.9(a). It is interesting to contrast Figure 6.9 and Figure 6.10. Even though the grouping of images (red brick wall) may appear similar, the distributions of scores across the rank one matrices are different except for the red regions. This implies that two different themes have been extracted even though it happens that certain images



(a) (b)

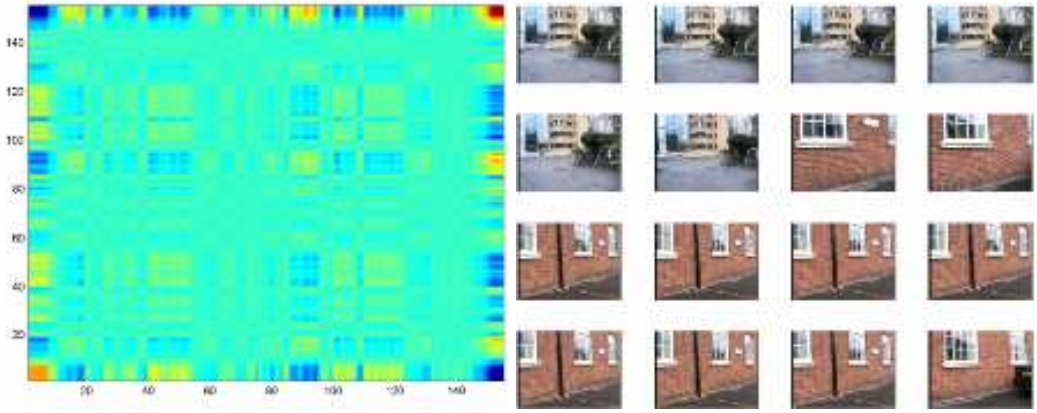
Figure 6.9: (a) shows $M_1^{Acland} = v_1 \lambda_1 v_1^T$ of the VSM shown in Figure 6.8. (b) shows 16 images associated with high scoring cells in the rank one matrix. These are mostly images of the red brick wall of the building. These images are mostly concentrated in multiple regions.



(a) (b)

Figure 6.10: (a) shows $M_2^{Acland} = v_2 \lambda_2 v_2^T$ of the VSM shown in Figure 6.8. (b) shows 16 images associated with high scoring cells in the matrix. These are all images of the red brick wall of the building. These images are highly similar to those shown in Figure (6.9b). The only slight distinction is that these particular images seem to have more emphasis on vertical edges as evident from the vertical pipe found in the top few images. The bright regions are mostly concentrated in the same regions as Figure (6.9a).

shared both themes.



(a) (b)

Figure 6.11: (a) shows $M_3^{Acland} = v_3 \lambda_3 v_3^T$ of the VSM shown in Figure 6.8. (b) shows 16 images associated with high scoring cells in the rank one matrix. Half of the images are images of a brown building beside a glass building while the other half are images of the red brick wall of the building. The common theme between these two categories of images is the shiny glass window. The bright regions are mostly concentrated at the top right and bottom left corner of the similarity matrix.

Figure 6.11(a) shows the third rank one matrix, $M_3^{Acland} = v_3 \lambda_3 v_3^T$. Half of the images in Figure 6.11(b) are images of a brown building beside a glass building while the other half are images of the red brick wall of the building. The commonality between these two categories of images appears to be shiny glass windows. The bright regions are mostly concentrated at the top right and bottom left corner of the similarity matrix. These few examples suggest that as conjured, the principal eigenvectors of M are associated with "themes" which permeate a particular environment. More practical examples are shown in Chapter 7. While these themes (which capture common similarity between many images) are useful for summarizing an environment, they are detrimental when detecting loop closure.

6.4 Rank Reduction

The previous subsection suggests that there are principal themes (represented as eigenvectors) within an environment, which broadly represent the structure of a similarity matrix. The relative

magnitude of λ_i is a measure of the degree to which the outer product matrix $M_i^{\ominus} = v_i \lambda_i v_i^T$ expresses the *overall* structure of the similarity matrix M. If themes are responsible for the dominant structure in M then, because $\sum_{i=1}^{r} \mathbf{v}_i \lambda_i \mathbf{v}_i^T$ is the best rank-r approximation to M under the Frobenius norm, it can be expected their effect in M to be captured in the dominant eigenvalues and eigenvectors. Thus, the effect of visual ambiguity (repetitive scene structure) can be diminished by reconstructing M by omitting the first r terms of the summation in Equation 6.1. This section discusses on how to choose r.

6.4.1 Rank Reduction based on Entropy Maximisation

For a $n \times n$ matrix, M, the relative significance is defined as, $\rho(i, r)$, of λ_i to the last $n - r + 1$ eigenvalues

$$\rho(i, r) = \lambda_i / \sum_{k=r}^{n} \lambda_k \tag{6.1}$$

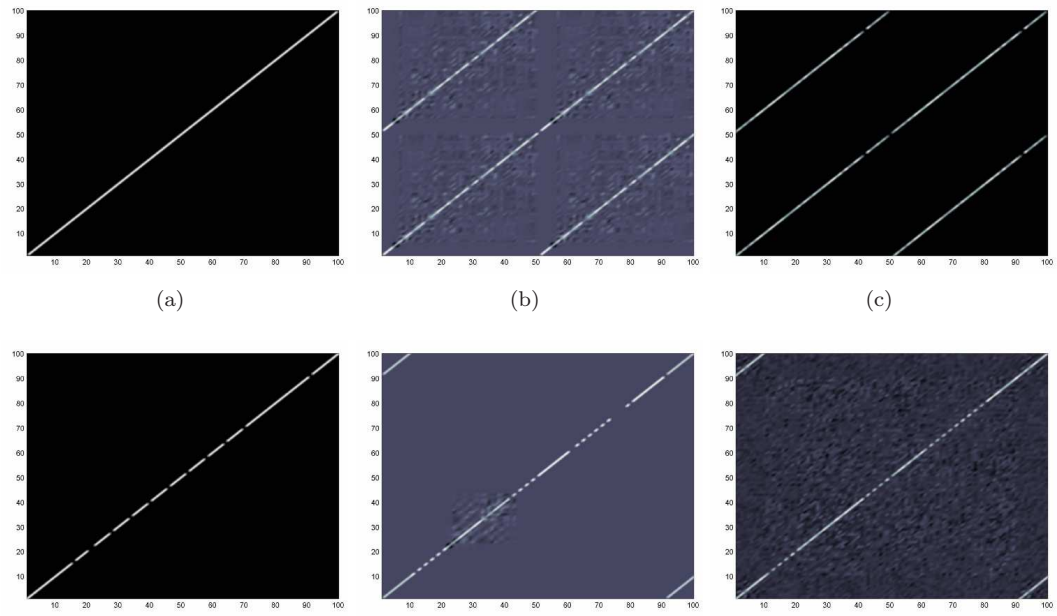
Using this, the complexity of decomposition of M can be measured as an entropy

$$H(M, r) = \frac{-1}{log(n)} \sum_{k=r}^{n} \rho(k, r) log(\rho(k, r)). \tag{6.2}$$

The 'Shannon entropy' of a similarity matrix, expressed in Equation 6.2, measures the complexity of the data from the distribution of the overall eigenvalues between the eigenvectors. It measures the complexity of the composition of M with first $r - 1$ dyads removed. $H = 0$ corresponds to an ordered and redundant similarity matrix in which all of the dataset can be captured by a single eigenvector. $H = 1$ corresponds to a disordered and random similarity matrix where all eigenvectors are equally expressed. Our approach is to sequentially remove outer-products from M until $H(M, r)$ is maximised, leaving a similarity matrix in which no one single theme dominates. The original similarity matrix M may be replaced with a rank reduced version.

$$\tilde{M} = \sum_{i=r^\star}^{n} \mathbf{v}_i \lambda_i \mathbf{v}_i^T \quad r^\star = \arg\max_r H(M, r) \tag{6.3}$$

6.4.2 Rank Reduced Synthetic Similarity Matrices

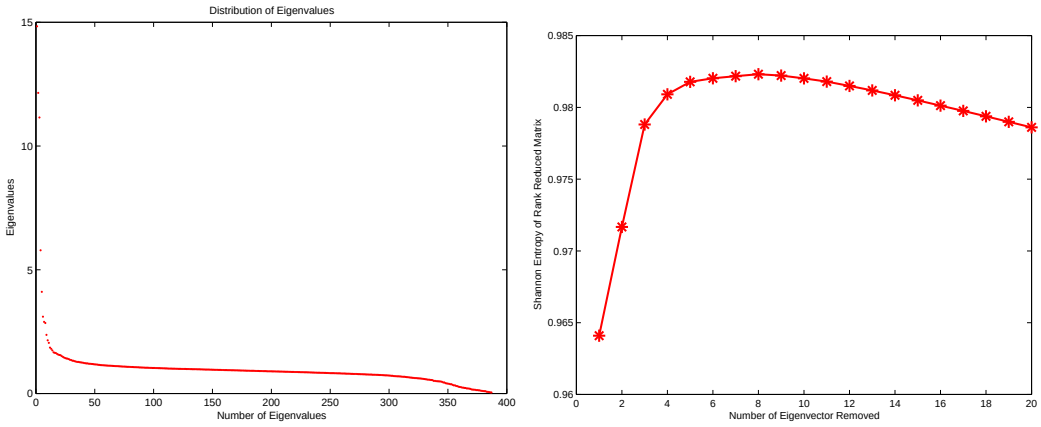


(a) (b) (c)

(d) (e) (f)

Figure 6.12: (a-f) shows the corresponding rank reduced similarity matrices of Figure 6.2(a-f).

The rank reduction approach based on entropy maximisation is applied on the synthetic similarity matrices. Figure 6.12 shows the ranked reduced synthetic matrices. For most parts, these rank reduced similarity matrices are similar to the original similarity matrices, except that ambiguous regions have been removed. The off-diagonals are not removed through the rank reduction process. From these few examples, it appears that spectral decomposition of similarity matrix may be a potentially useful technique to remove ambiguous regions (which might prompt false loop closure) while retaining the actual loop closures. The applicability of the entropy maximisation approach for rank reduction of actual similarity matrices is tested in the next subsection.



(a) (b)

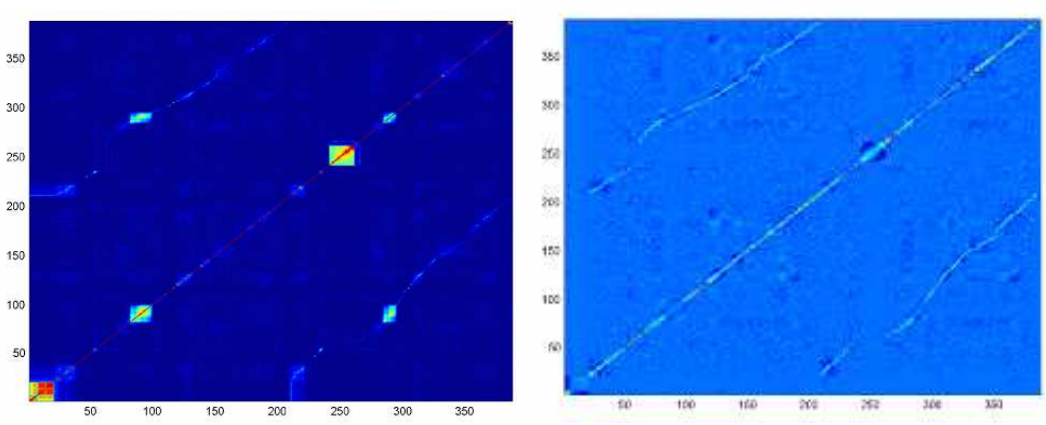
Figure 6.13: (a) shows a typical plot of eigenvalues. This set of eigenvalues is obtained from the EVD of M^{Thom}. (b) shows a typical distribution of the 'Shannon entropy' of successively ranked reduced similarity matrices. This distribution of entropy is based on the set of eigenvalues in (a).

6.4.3 Rank Reduction in Experiment A

In Figure 6.13, a distribution of the eigenvalues obtained from the EVD of M^{Thom} from the Thom Building dataset is shown. The eigenvalues drop drastically for the first few initial values before levelling off. Figure 6.13(b) depicts $H(M, r)$ as a function of r (using the data set whose eigenvalues are shown in Figure 6.13(a)). For this particular case, the maxima was reached after removing the first eight outer products before it begin to decrease as more outer products are removed. Figure 6.14(b) shows the final rank reduced matrix, $\tilde{M}^{Thom}$. Notice the effects of the visually ambiguous regions have been diminished while the off-diagonals are still visible.

6.4.4 Rank Reduction in Experiment B

In Figure 6.15, a distribution of the eigenvalues obtained from the EVD of M^{Acland} from the Acland Building dataset is shown. Similarly, the eigenvalues drop drastically for the first few initial values before levelling off. Figure 6.15(b) depicts $H(M, r)$ as a function of r (using the data set whose eigenvalues are shown in Figure 6.15(a)). For this particular case, the maxima was reached after removing the first four outer products. Figure 6.16(b) shows the final rank reduced



(a) (b)

Figure 6.14: (a) shows M^{Thom} with limited VARs. (b) shows the rank reduced VSM, $\tilde{M}^{Thom}$. Notice that the VARs have been removed through the process of rank reduction while the loop closing off-diagonal remains.

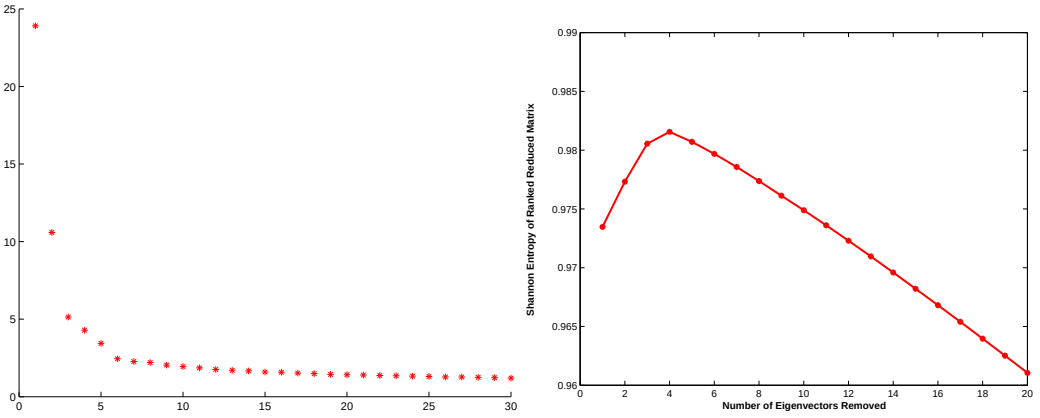
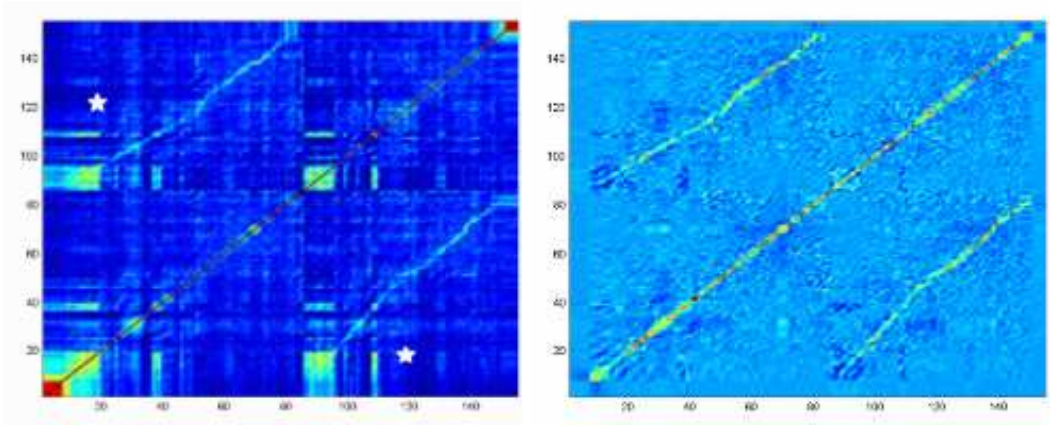


Figure 6.15: (a) shows a plot of eigenvalues. This set of eigenvalues is obtained from the EVD of M^{Acland}. (b) shows a typical distribution of 'Shannon entropy' of successively ranked reduced similarity matrices. This distribution of entropy is based on the set of eigenvalues in (a).

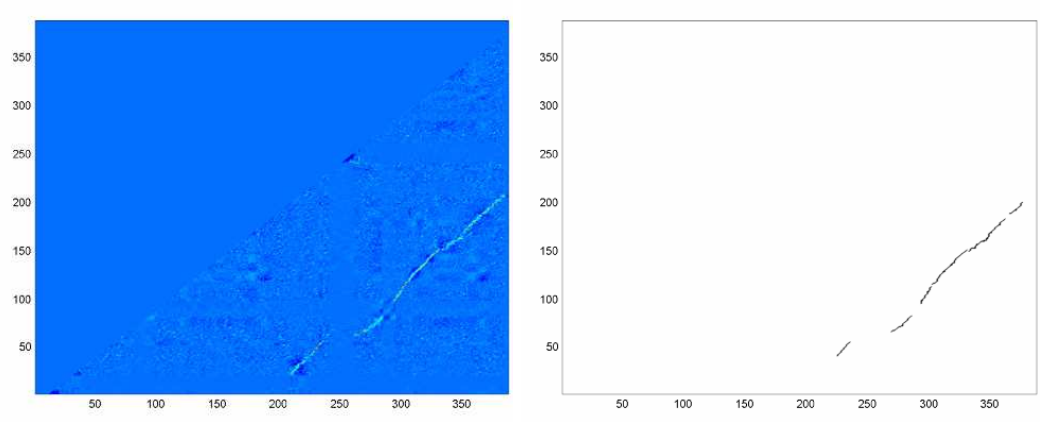
matrix, $\tilde{M}^{Acland}$. Notice the false off-diagonal has been removed, leaving behind the off-diagonal representing a true loop closing event.



(a) (b)

Figure 6.16: (a) shows M^{Acland} with limited VARs. (b) shows the rank reduced VSM, $\tilde{M}^{Acland}$. Notice that the VARs have been removed through the process of rank reduction while the loop closing off-diagonal remains.

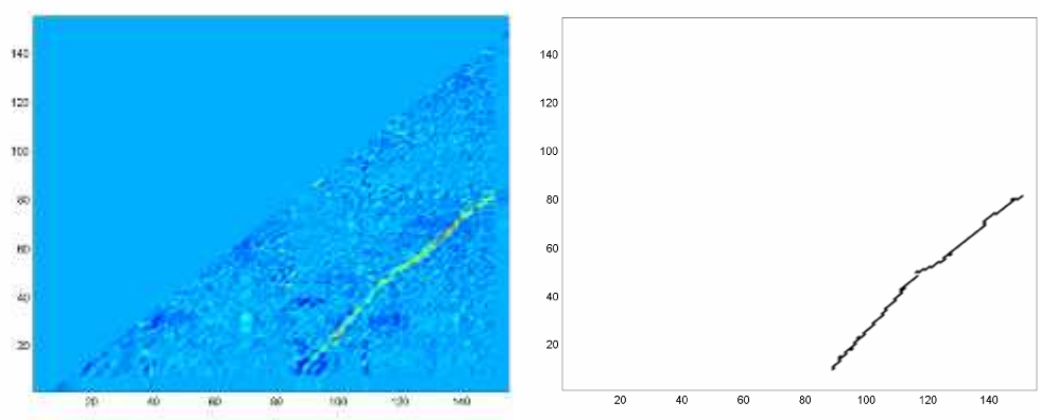
6.5 Sequence Detection in Rank Reduced Similarity Matrix



(a) (b)

Figure 6.17: (a) shows a LTM of a $\tilde{M}^{Thom}$ (LTM of M^{Thom} was shown as Figure 6.14(a)). The bright off-diagonal represents a loop closure event. The VARs have been removed through rank reduction. (b) shows the result of applying the sequence detection algorithm to find significant sequences.

One of the concerns is that spectral decomposition of a similarity matrix has an adverse effect on the off-diagonals (corresponding to true loop closing events) even though they remain visible. This is especially the case when loop closing has taken place at an ambiguous area. Consequently, the performance of sequence detection on rank reduced matrices is investigated here. Figure 6.17(a) shows a LTM of $\tilde{M}^{Thom}$ shown in Figure 6.14(b). Figure 6.17(b) shows the result of applying the sequence detection algorithm to find significant sequences. Instead of detecting a single long sequence, multiple, shorter sequences are detected. One of the reasons is that VARs are no longer considered, causing breaks along the original long sequence. However, this is not a matter of concern as a single, reliable detection is all that is required for a map to corrected accurately. Figure 6.18(a) shows a LTM of $\tilde{M}^{Acland}$. Figure 6.18(b) shows the result of applying the sequence detection algorithm to find significant sequences.



(a) (b)

Figure 6.18: (a) shows a LTM of a rank reduced VSM, $\tilde{M}^{Acland}$. The bright off-diagonal represents a loop closure event. The VARs have been removed through rank reduction. (b) shows the result of applying the sequence detection algorithm to find significant sequences.

6.6 Statistical Significance of Sequences

Given a similarity matrix, the sequence detection algorithm detects the best matching pair of subsequences, $\mathcal{A}$ and $\mathcal{B}$, between two sequences of elements. A pair of subsequences of similar elements suggests the possibility of an overlap (loop closing or intersection between two local maps) but is it really due to a genuine loop closure? A method to evaluate the confidence of the pairing between $\mathcal{A}$ and $\mathcal{B}$ is desired. The absolute value of the maximal alignment score, $\eta_{\mathcal{A},\mathcal{B}}$, is not a reliable indication of how likely it is that an overlap has occurred. It is unwise to trigger loop closing or map intersection each time a sequence is detected. Instead, loop closing or map intersection should only be triggered if the maximal alignment score is statistically significant. The question is how to determine the statistical significance of a maximal alignment score.

The left-hand side of Figure 6.19 shows a similarity matrix that has significant overlap and the right-hand side shows a similarity matrix with no overlap. Intuitively, the maximal alignment score from the matrix on the left-hand side should be higher than the one from the right-hand side. More importantly, the maximal alignment score from the matrix on the left-hand side is likely to be significantly higher than other possible alignment scores from shorter sequences. A principled

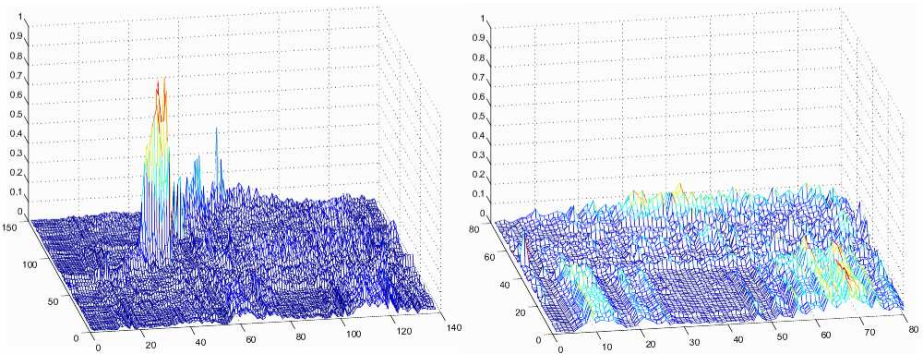


Figure 6.19: Left: A 3D plot of the VSM between two image sequences that have overlapped intersection. Right: A 3D plot of the VSM between two image sequences that have no overlapped intersection.

manner for determining the statistical significance of the maximal alignment score has been proposed [117]. Maximal alignment scores can be well described by a Gumbel distribution (an extreme value distribution [41]), given by Equation 6.4. This probability density function can be used to judge the significance of the maximal alignment score for the pair of matching subsequences [15, 78]. Extreme value distribution has been used to describe the distribution of extreme events. Examples of such distributions include highest daily temperature and highest daily water level over extended periods of time.

The probability density function of $\eta_{\mathcal{A},\mathcal{B}}$ occurring is described by an Gumbel distribution as follows:

$$p(\eta_{\mathcal{A},\mathcal{B}}) = \frac{1}{\beta} e^{-z} e^{-e^{-z}} \tag{6.4}$$

where

$$z = \frac{\eta_{\mathcal{A},\mathcal{B}} - \mu}{\beta} \tag{6.5}$$

where $\eta_{A,B}$ is the maximal alignment score, μ is the mean of the distribution and β is the standard

deviation of the distribution. This p.d.f can be used to judge the significance of the maximal alignment score for the pair of matching subsequences.

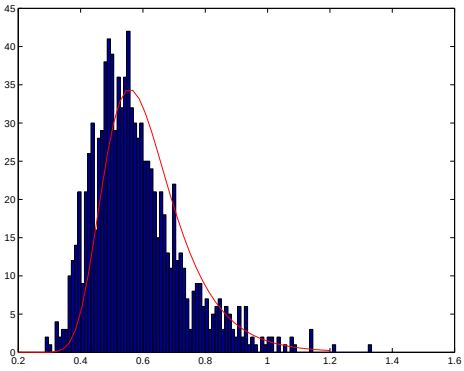


Figure 6.20: Typical distribution of maximal segment scores from 1000 random shuffles of the similarity matrix.

Adopting the approach in [1], a similarity matrix is randomly shuffled and a maximal alignment score is obtained each time. This results in a distribution such as that shown in Figure 6.20. The distribution parameters μ and β are estimated from the histogram of maximal alignment scores by fitting the parameters using a Levenburg Marquardt algorithm. Equipped with estimates $\hat{\mu}$ and $\hat{\beta}$ and the closed form cumulative distribution function we can evaluate the probability of scores greater than or equal to $\eta_{A,B}$ conditioned on all n images:

$$P(\eta \geq \eta_{A,B}|M) = 1 - e^{-e^{-z}} \tag{6.6}$$

Equation 6.6 allows the evaluation of the probability that an extracted sequence of observation matches, $\langle \mathcal{A}, \mathcal{B} \rangle$, with score $\eta_{A,B}$, could have been generated at random from M. The differences between the sequence score $\eta_{A,B}$ obtained from the original, temporally ordered M and those obtained from the randomly shuffled versions are solely attributable to the topology or connectedness of the spatial locations at which the vehicle captured the images. Thus Equation 6.6 can be used to evaluate the probability, conditioned on all previous scene appearances, that the detected sequence does indeed indicate a bona-fide loop closure.

The entire loop-closure detection process can now be summarised:

1. From n images build a $n \times n$ similarity matrix M as described in Section 5.3.
2. Remove common mode similarity via rank reduction as described in Section 6.4.
3. Estimate extreme value distribution parameters from the rank reduced similarity matrix, $\tilde{M}$ as described in Section 6.6.
4. Test significance of the alignment score, $\eta_{\mathcal{A},\mathcal{B}}$ using Equation 6.6. If acceptable, advise loop closure, go to 5.
5. Extract highest scoring sequence from $\tilde{M}$.
6. End

6.7 Summary

This chapter has described how spectral decomposition of a similarity matrix can help to remove the effects of ambiguous artefacts and, as such, tackle the perceptual aliasing problem. The effects of eigenvalue decomposition on synthetic similarity matrices are investigated. Experimental results from eigenvalue decomposition of different visual similarity matrices are demonstrated. A principal method of rank reduction through entropy maximisation of a similarity matrix is employed to remove the effects of ambiguous artefacts. The sequence detection algorithm is applied on rank reduced similarity matrices to detect significant sequences. A method of assessing the significance of sequences detected has been described. The entire loop closing process is finally summarised. More experimental results from employing this loop closing process will be shown in the following chapter.

Chapter 7

Experiments

7.1 Introduction

In this chapter, extensive results from testing our proposed loop closing process in varied and challenging environments are shown. This chapter starts off with Section 7.2 examining the limitations of the sequence detection algorithm without explicit ambiguity management by providing a failure case. It then demonstrate how spectral decomposition technique can help to enhance robust sequence detection in rank reduced similarity matrices. The loop closing algorithm is then subjected to further testing in specific experimental settings and scenarios, as described in Section 7.3 and Section 7.4. After which, the loop closing algorithm is repeated in Section 7.5 for an experiment in which laser ranging is the primary sensor modality. This chapter concludes with a summary in Section 7.6 of the results from the experiments.

Sample sets of images and laser scans captured from various exploration runs can be found in Appendix C and the full data sets can be found in http://www.robots.ox.ac.uk/∼klh/dataset.htm. They are divided into four sets: "Thom", "Jenkin", "New College" and "Cloister". The qualities of the data sets are summarised in Table 7.1.

Data Set	Purpose	Discussion & Illustration
Thom	Benign workspace, producing a clean similarity matrix with little ambiguity	Section 6.3.2. Figure 6.4 and Figure 5.5.
Jenkin	Medium sized loop around a cluster of buildings. Used to showcase combination of loop closure detection with SLAM system	Section 7.2. Figure 7.16 and Figure 7.11.
New College	Larger data set with a combination of visual themes. A noisy data similarity matrix results with hard to discern loop closures.	Section 7.3. Figure 7.19 and Figure 7.18.
Cloisters	Repetitive visual structure producing a noisy similarity matrix	Section 7.4. Figure 7.29 and Figure 7.35.

Table 7.1: Description and references to four data sets

7.2 Scenario I: Outdoor Urban Environment

In this experiment, the exploration run involves an extended loop (over 400 metres in length) around an area that has multiple similar-looking buildings. The camera settings are as follows: An image is captured for every metre travelled by the robot. Camera orientation changes after an image is captured. The camera orientation toggles from 60^o left of robot's heading to 60^o right of robot's heading and vice versa. A sequence of 487 images was collected from the exploration run. The image sequence, $I^{Jenkin} = \{I_{Jenkin1}, \cdots, I_{Jenkin487}\}$, contains multiple images that are visually similar to each other. (See Appendix C-2.) A VSM, M^{Jenkin}, is constructed for this image sequence and is illustrated in Figure 7.1[1]. The astute observer will notice an off-diagonal dark line starting at $I_{Jenkin400}$ in Figure 7.1. This is the start of the loop closure sequence. However, one will also notice there are huge chunks of bright square regions (VARs) found within the VSM. The sequence detection algorithm is applied on the LTM of M^{Jenkin}, as shown in Figure 7.2(a). Figure 7.2(b) shows the result of applying the algorithm to extract the correct sequence. In this exemplar case, it demonstrates the algorithm is still able to detect the loop closure despite presence of significant VARs. The concern however is that the VARs will prompt incorrect loop

[1] Throughout this chapter, similarity matrices are displayed with artificially increased contrast so their fine structure survives the reproduction process.

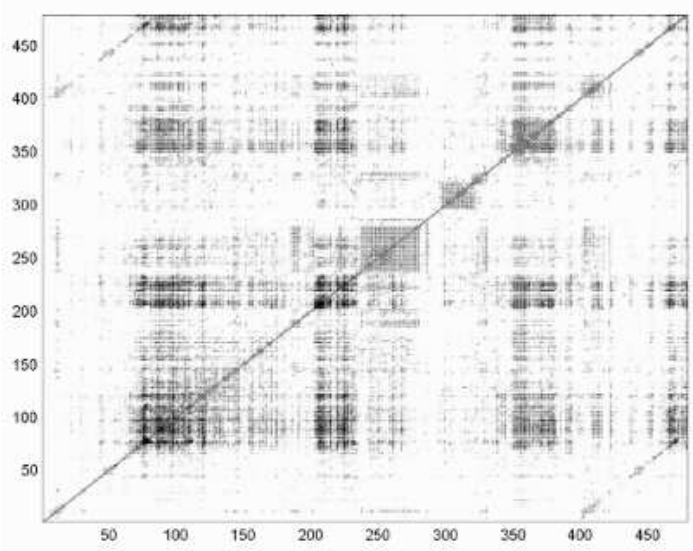
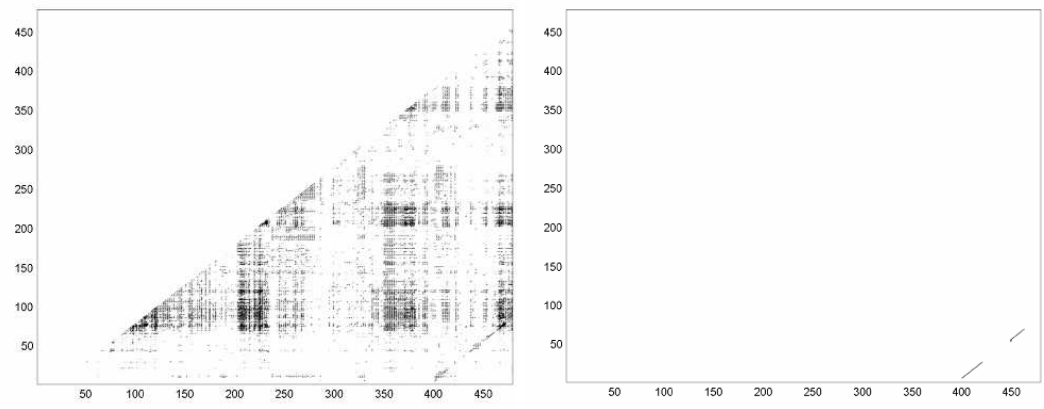


Figure 7.1: Above illustrates a VSM, M^{Jenkin}, constructed from a sequence of images captured from a visually confusing environment. Dark tone represents high similarity scores while light tone represents low similarity scores. Loop closure appears as bright off-diagonal lines. There are many bright regions (VARs) across the similarity matrix, representing regions where local scenes are mutually similar.

closure earlier[2] on before the actual loop closure. To test for such a scenario, the loop closing subsequence, $\{I_{Jenkin400}, \cdots, I_{Jenkin487}\}$ of the image sequence, I^{Jenkin}, is truncated and a new image sequence, $I^{Jenkin'} = \{I_{Jenkin1}, \cdots, I_{Jenkin399}\}$ is obtained. A new VSM, $M^{Jenkin'}$, is constructed for this truncated image sequence.

Figure 7.3(a) is a LTM of the truncated VSM, $M^{Jenkin'}$. The loop closing portion of the similarity matrix has been removed but significant VARs remain. Figure 7.3(b) shows the result of applying the sequence detection algorithm on $M^{Jenkin'}$. The algorithm erroneously finds a significant sequence within a VAR. The two image subsequences (shown in Figure 7.4) are indeed similar; repeating visual entities such as wall patterns, window styles, vegetation result in a Type I error. Not only do these image subsequences contain repetitive visually similar artefacts, similar images are arranged in approximately the same order; image of wall followed by image of road.

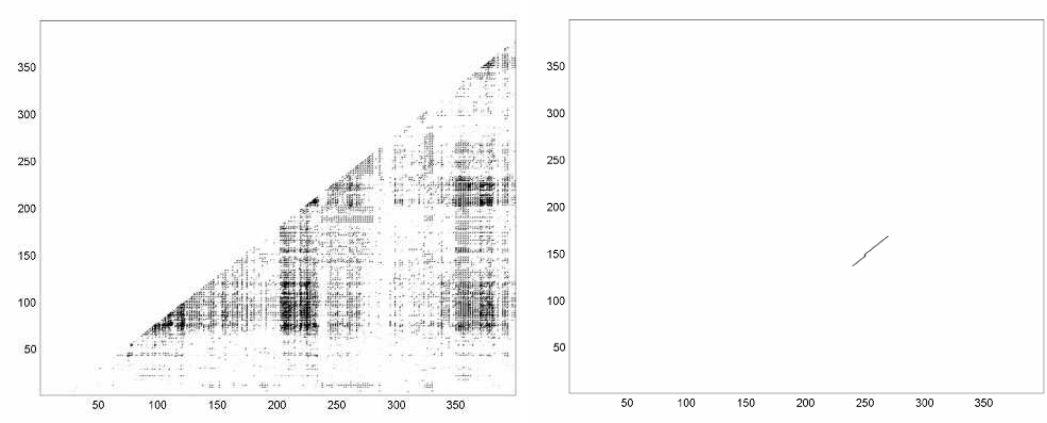
[2]The image sequence is constantly growing as more images are appended. Consequently, the similarity matrix grows along with the length of the image sequence



(a) (b)

Figure 7.2: (a) is a LTM of the VSM, M^{Jenkin}, shown in Figure (7.1). There are significant VARs as depicted by bright square regions. Loop closure has occurred as depicted by an off-diagonal bright line. (b) shows the result of applying the sequence detection algorithm to find the most significant sequence.

Consequently, the sequence detection algorithm is unable to remove this false positive. A false loop closure will be prompted. False loop closures are a real disaster for SLAM systems, leading to almost irreversible deviation of estimated map and robot pose. The next subsection demonstrates how spectral decomposition of a VSM can remove the effects of visually ambiguous artefacts from consideration in loop closure detection.

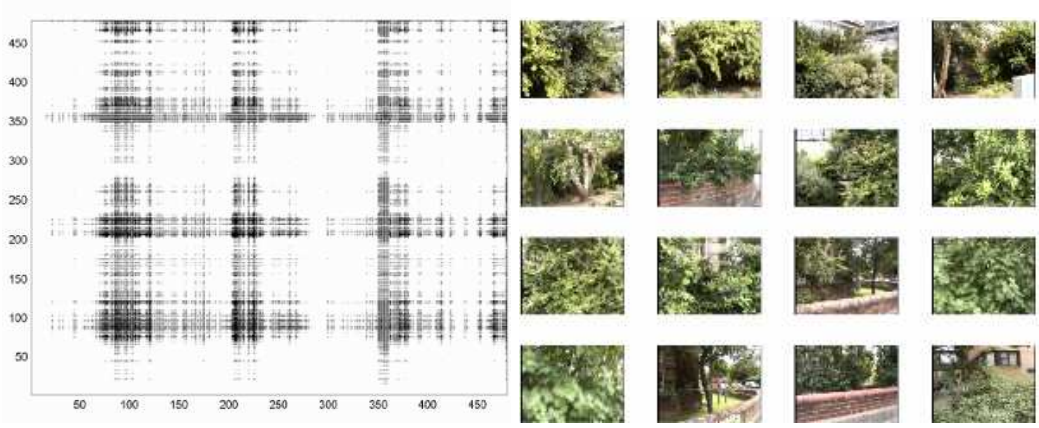


(a) (b)

Figure 7.3: (a) shows a LTM of a subset of the VSM, $M^{Jenkin'}$, shown in Figure (7.1). Loop closing sequence has been removed but significant VARs remain. (b) shows the result of applying the sequence detection algorithm to find a significant sequence. The algorithm erroneously selected a sequence within a VAR.

Figure 7.4: Top: Subset of a query subsequence of images. Bottom: Subset of a best matching subsequence of images.. This is an erroneous sequence that results by applying the sequence detection algorithm directly to $M^{Jenkin'}$. The two sequences are indeed similar; repeating visual entities such as wall patterns, window styles, vegetation result in a Type II error.

7.2.1 Spectral Decomposition of VSM

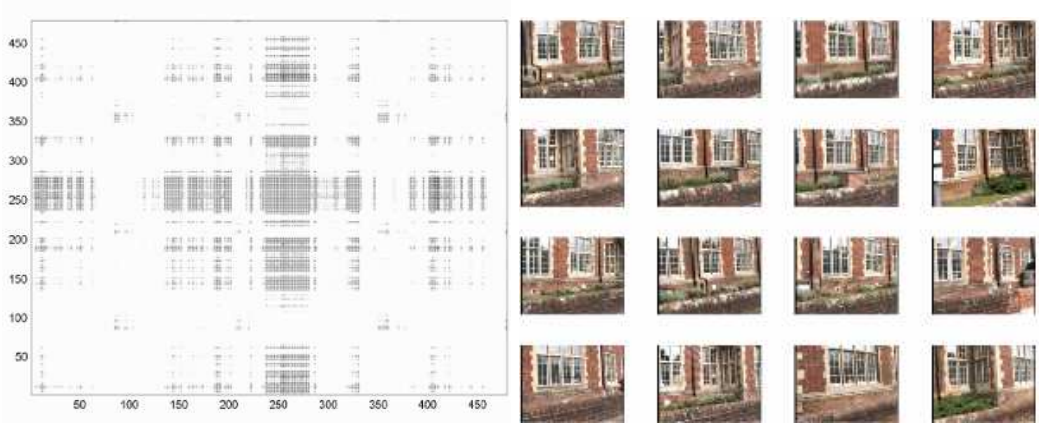


(a) (b)

Figure 7.5: (a) shows rank one matrix ($M_1^{Jenkin} = v_1\lambda_1 v_1^T$) of the VSM shown in Figure 7.1. (b) shows 16 images associated with the highest scoring cells in the matrix. These are mostly images of vegetation. It appears vegetation constitutes a common theme throughout the database as bright coloured cells spread across the matrix.

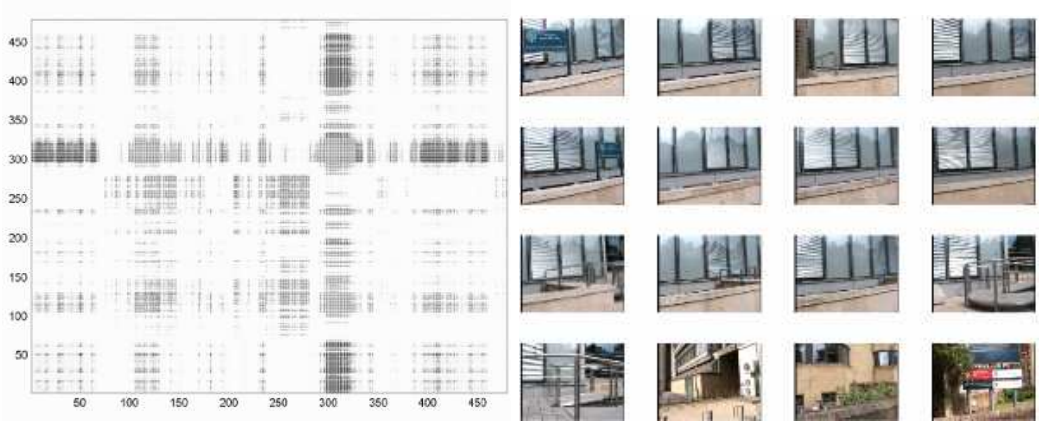
A rank one matrix based on the largest eigenvalue of a similarity matrix is described by $M_1 = v_1\sigma_1 v_1^T$. For example, Figure 7.5(a) is the first[3] rank one matrix, $M_1^{Jenkin} = v_1\lambda_1 v_1^T$, of the VSM, M^{Jenkin}, shown in Figure 7.1(a). Visually, these two matrices are very similar. Figure 7.5(b) shows 16 different images associated with high scoring cells in the rank one matrix. All these images have significant portion of vegetation. The EVD has extracted the dominant 'theme' within this particular environment. Indeed, vegetation can be found throughout the explored environment. The distribution of these vegetation within the environment corresponds with the distribution of high scoring cells along the main diagonal of the matrix, M^{Jenkin}.
Figure 7.6 shows the rank one matrix associated with the second largest eigenvalue, $M_2^{Jenkin} = v_2\lambda_2 v_2^T$, and its associated set of images. High scoring cells in this matrix are concentrated to a small area. These images contain rectangular structures such as bricks and windows. All these images are captured at the front and back of a building, which the robot

[3]The order of rank one matrices is based on the magnitude of the corresponding eigenvalues.



(a) (b)

Figure 7.6: (a) shows rank one matrix ($M_2^{Jenkin} = v_2 \lambda_2 v_2^T$). (b) shows images associated with the highest scoring cells in the matrix. These images contain rectangular shaped entities such as windows and bricks. Given that the robot explored in an urban environment, it is not surprising such features are highly common. High scoring cells in this matrix are concentrated to a small area.



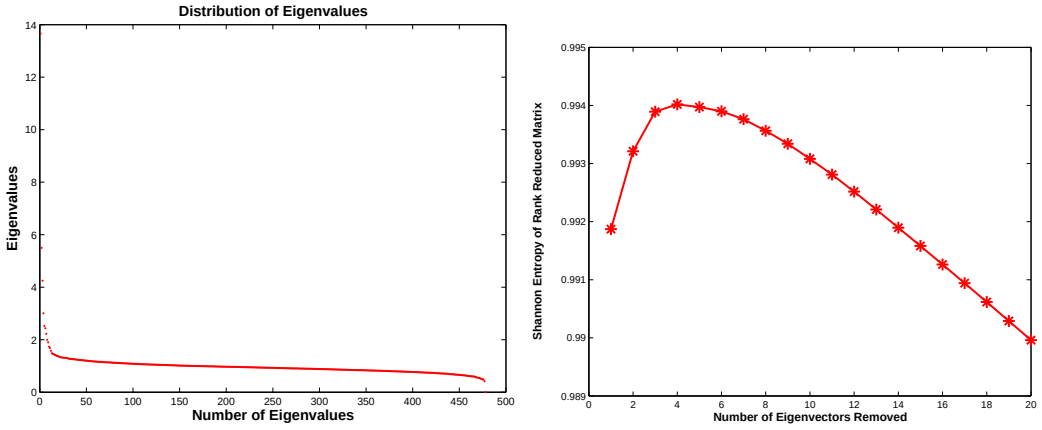
(a) (b)

Figure 7.7: (a) shows rank one matrix ($M_3^{Jenkin} = v_3 \lambda_3 v_3^T$). (b) shows images associated with high scoring cells in the matrix. Shiny surfaces are a common theme in these images. It can be seen that bright coloured cells are mostly confined to a small region within the matrix.

encircled. The building is made up of red brick wall pattern, interlaced with white, Georgian windows. Many buildings have similar facades. A false loop closure could have been easily triggered when the robot circled to the back of the building, which is very similar to the front of the building.

Figure 7.7 shows the rank one matrix associated with the third largest eigenvalue, $M_3^{Jenkin} = v_3 \lambda_3 v_3^T$, and its associated set of images. The high scoring cells in this matrix are concentrated to a small part of the matrix. These are images of shiny surfaces. Most of the images contain the reflective glass walls of a building. Other images also contain shiny glass window or metallic surfaces. In summary, the environment explored has plenty of vegetation, repetitive rectangular structures and artefacts with shiny surfaces. As expected, a typical urban environment is likely to be comprised of such themes.

7.2.2 Rank Reduction of VSM

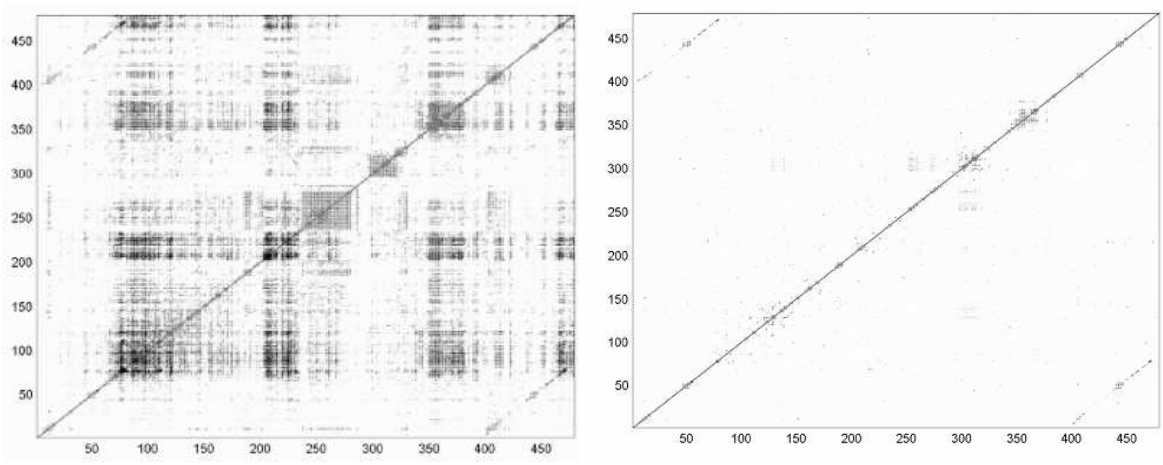


(a) (b)

Figure 7.8: (a) shows a typical distribution of eigenvalues. This set of eigenvalues is obtained from the EVD of the VSM, M^{Jenkin}, shown in Figure 7.9(a). (b) shows a typical distribution of 'Shannon entropy' of successively rank reduced similarity matrices.

Figure 7.8(a) shows a typical distribution of eigenvalues. This distribution of eigenvalues is obtained from the EVD of the VSM, M^{Jenkin}. It is observed that the magnitudes of the

eigenvalues initially drop dramatically before the rate of decline levels off. This suggests that there are a few principal eigenvectors that describe the similarity matrix well. Figure 7.8(b) shows a typical distribution of 'Shannon entropy' of successively rank reduced similarity matrices. For this particular example, the peak in Shannon entropy is reached after removing the top 4 eigenvalues and corresponding eigenvectors.



(a) (b)

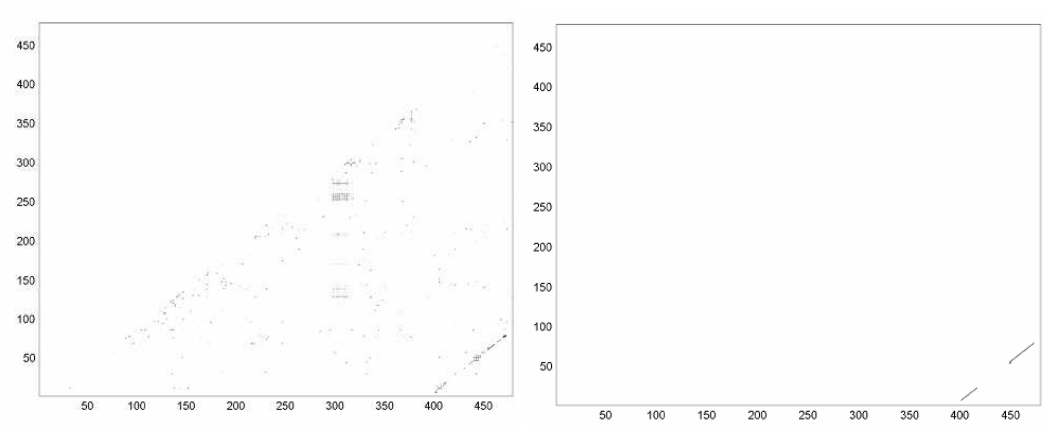
Figure 7.9: (a) shows a VSM, M^{Jenkin}. (b) shows the VSM, $\tilde{M}^{Jenkin}$ after rank reduction. Note the off-diagonal bright line (which signifies the loop closure) has not been affected by the rank reduction whereas VARs within the matrix have been removed.

Figure 7.9(b) shows the rank reduced matrix, $\tilde{M}^{Jenkin}$, using the entropy maximisation criterion method. The VARs has been removed through rank reduction. However, the off-diagonal (which indicates loop closing) remains.

7.2.3 Robust Sequence Detection in Rank Reduced VSM

The next step is to determine if the sequence detection algorithm will be able to pick out the loop closure from this rank reduced similarity matrix and more importantly, if it will erroneously pick out a sequence when loop closure does not exists. Figure 7.10(a) is the LTM of the rank reduced VSM shown in Figure 7.9(b). Figure 7.10(b) shows the result of applying the sequence detection algorithm to find a significant sequence. Figure 7.11 shows a portion of the matching pair of image

subsequences. The top row contains images from one subsequence and the bottom row contains images from another subsequence. It can be observed visually that each column contains a pair of matching images. This means that the robot has completed a loop. The most recent subsequence of images has matched a previously captured subsequence of images.

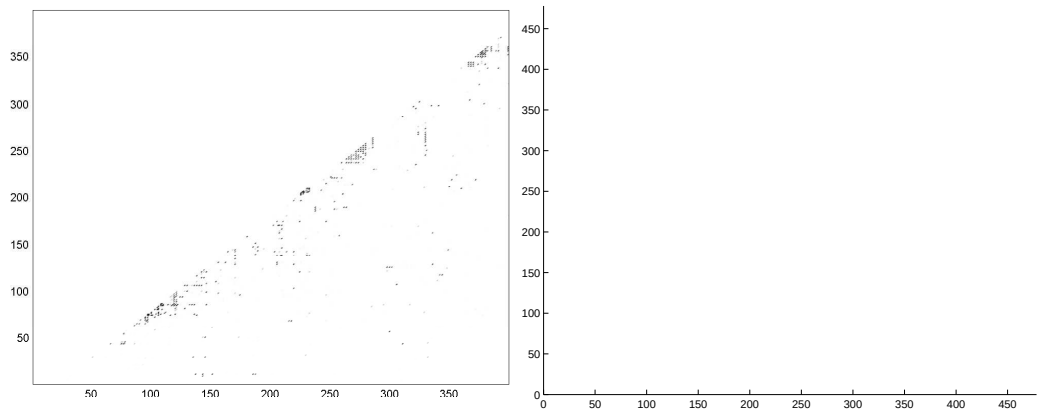


(a) (b)

Figure 7.10: (a) shows a LTM of the rank reduced VSM, $\tilde{M}^{Jenkin}$. (b) shows the result of applying the sequence detection algorithm to find significant sequences.

Figure 7.11: Results from local sequence extracted from rank reduced VSM. Top: Subset of a query subsequence of images. Bottom: Subset of a best matching image subsequence.

The next test is to determine if the sequence detection algorithm will pick out a erroneous loop closure in the rank reduced, truncated VSM, $\tilde{M}^{Jenkin'}$, as it did with the original, truncated VSM, $M^{Jenkin'}$. Figure 7.12(a) is a LTM of the VSM, $\tilde{M}^{Jenkin'}$. Figure 7.12(b) shows the result of applying the algorithm to find a significant sequence. No significant sequence was found.

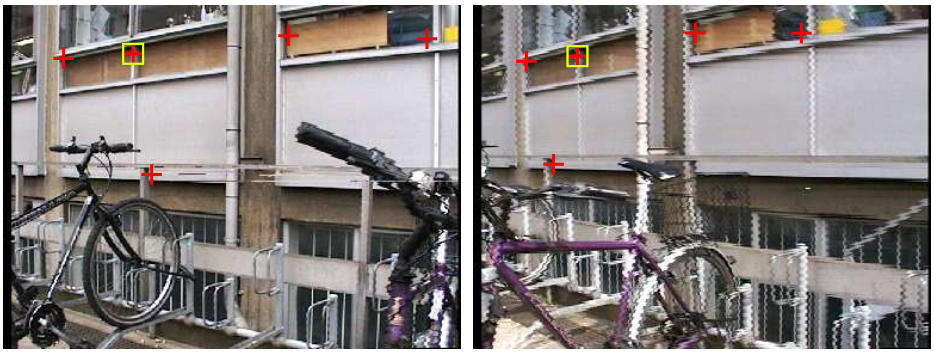


(a) (b)

Figure 7.12: (a) shows a LTM of a VSM, $\tilde{M}^{Jenkin'}$. (b) shows the result of applying the sequence detection algorithm. No significant sequence was found.

Loop Closure Geometry

Given a sequence of time-stamped views, the SLAM system can detect proximity to a previously visited location. In order to execute the loop closure, the geometry of the loop closure has to be determined — the Euclidean transformation between recent and past views that constitutes the loop closure. One option would be to use image time to index into the pose state $\mathbf{X}$ vector, which is, after all, a sequence of past poses, to find which previous pose $\mathbf{x}_{v_i}$ occupied the scene we are now revisiting at time t_k. However, this approach has problems when it comes to undertaking a laser scan match to deduce *precise* estimate of the interpose transformation $\mathbf{z}_{i,j}$ – without a reliable prior or "seed solution" the iterative scan matching method that has been adopted in this work is prone to converge to an incorrect minima. At the same time, exhaustive search in 6D is prohibitively slow. Instead, the sequences $\mathcal{A}$ and $\mathcal{B}$ and laser range data are used to estimate $\mathbf{z}_{i,j}$. Consider the following common projective model of two identical cameras with projection matrices P and P' [47]. (See Appendix B.) A homogenous 3D image scene point $X = [X, Y, Z, 1]^T$ is imaged at $x = PX$ for the first camera and $x' = P'X$ for the second camera. Without loss of generality, the origin can be fixed at the centre of the first camera. If the second camera centre is



(a) (b)

Figure 7.13: Estimating the rigid loop closure transformation. The five point solution is employed [100] using at least five points of correspondence, (red crosses) selected from SIFT descriptors extracted from both images. Scale ambiguity is removed by using the 3D depth of a particular feature (boxed) derived from the onboard laser.

parameterized by a rotation matrix R and a translation t with respect to the origin, then P and P' can be written $K[I|0]$ and $K[R|t]$ respectively. Here, K is the matrix of intrinsic camera parameters. In the case of calibrated cameras (K known) the image points, x and x', are related by the *essential matrix* E such that $x'^T Ex = 0$. The determination of relative camera poses via decomposition of the essential matrix has been used to good effect in robot localization [66, 102] and SLAM navigation [28]. Given two image views of the same scene, five points of correspondence (shown in Figure 7.13) are selected for use in the *five point method* [100] to determine relative pose. The essential matrix has a convenient structure. It can be written in terms of R and t as $[t]_\times R$ where $[t]_\times$ denotes the 3×3 skew symmetric (cross product) matrix constructed from t. Given the elements of E this decomposition yields four possible solutions for R and t. The correct solution is selected by application of a final cheirality constraint; scene points must be in front of the cameras. The two images used in the geometry estimation are those with the greatest similarity score within the image sequence returned from the loop-closure detector. The resulting t is correct only up to scale and range information from the laser data is used to perform the metric upgrade.

Given the instantaneous rigid transformation between the laser scanner and the camera, 3D laser data can be expressed in the 3D coordinate frame of the camera and projected onto the image

(a) (b)

Figure 7.14: The 3D laser data rendered in the image plane. The vehicle employs a nodding mechanism to rotate the laser scanner to produce a 3D scan of the environment. The laser scanner swiftly (0.6 Hz) samples range over +/- 90^o in azimuth and -40^o to 60^o in elevation. Here the camera is looking 60 degrees right.

plane as shown in Figure 7.14. For each of the five visual features, the nearest projected laser-range points are found. Out of the five image features, the image feature which has the closest projected laser range point is selected. The 3D position of this image feature is now known, allowing the final scale ambiguity to be removed and yielding metric t. Given estimates of R and t the iterative laser scan matching can proceed, starting with these estimates as an initial solution to $T_{i,j}$. The scan matcher further aligns the two scans, refining estimates of $T_{i,j}$ to use as a measurement on the SLAM state vector.

7.2.4 Results

In Figure 7.15, the blue points demarcate the poses of the robot in the state estimate maintained by a delayed-state EKF before loop closure. In reality, the robot has closed a loop but the estimated pose placed the robot more than 100 metres away from its actual position. The huge discrepancy in estimated position and actual position means that the loop closing event is not likely to be detected by current techniques, which rely on pose estimates to prompt loop closure. Using our loop closing technique, the point of loop closure is detected as discussed in Subsection 7.2.3. The pose change between the two views are determined using the technique described in

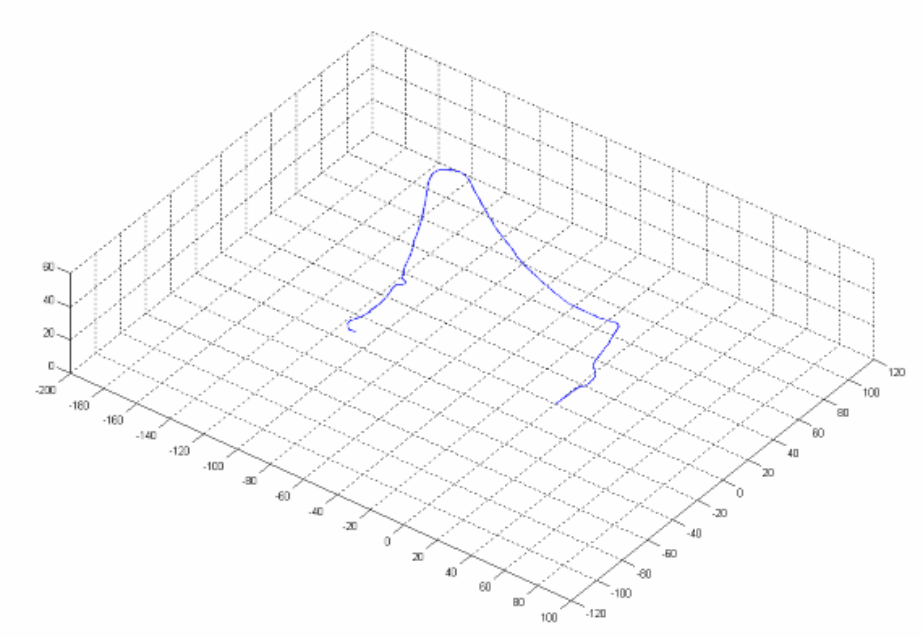


Figure 7.15: The blue points demarcate the robot pose estimates maintained by an EKF just before loop closure. In reality, the robot has closed a loop but the estimated pose placed the robot more than 100 metres away from its actual position.

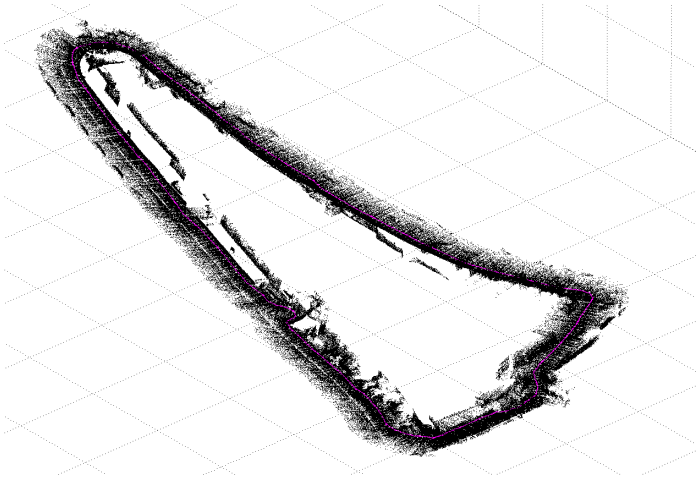


Figure 7.16: A 3D laser map built by an EKF before loop closure. A fine red line which represents the corrected robot trajectory can be seen under close inspection.

Subsubsection 7.2.3. Figure 7.16 shows the estimated map and trajectory of the robot after it traversed the perimeter of a cluster of buildings. Figure 7.17 shows a plan view of the final estimated robot trajectory superimposed over an aerial photograph of the workspace. Additionally, a metric grid has been placed over the area of interest. The astute reader will notice a discrepancy between the map of Figure 7.16 and the plan view of Figure 7.17 (where the trajectory undergoes a semicircular perturbation on the western leg of the circuit). This is because the original buildings present in the photograph have now been replaced with a new Information Engineering Building. The general concept of how the proposed loop closing detection technique work has been explained. The focus is now shifted to testing the performance of the algorithm on different forms of similarity matrix. Various environmental settings are explored to investigate the relevance of the algorithm in different scenarios.

Figure 7.17: An aerial image of the workspace with the final estimated robot trajectory and metric grid (20 metres per division) superimposed on it.

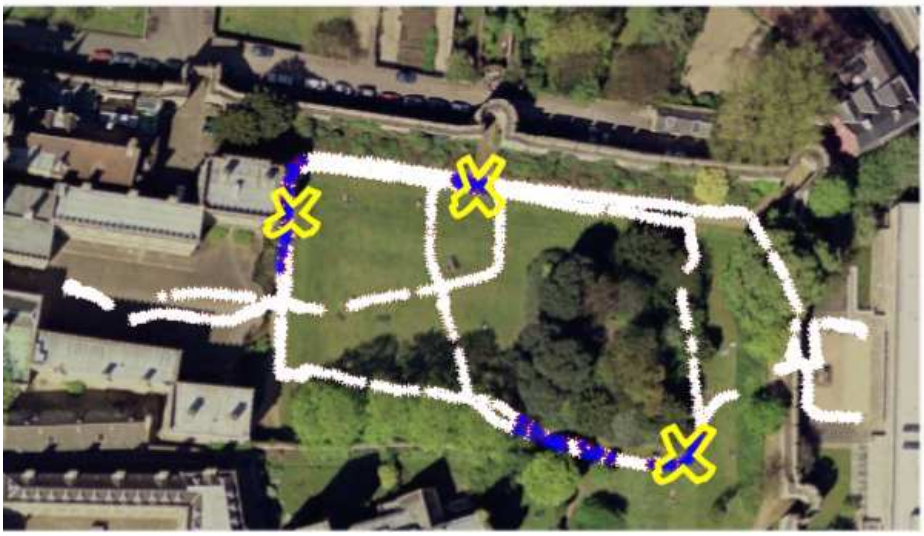


Figure 7.18: An aerial image of the test environment – New College Garden. The position of the robot is measured by a GPS receiver with positional uncertainty of about 5 metres. The path taken by a robot is marked as white crosses. There are various breaks in GPS signal reception. The large yellow crosses mark the positions where loop closure is detected.

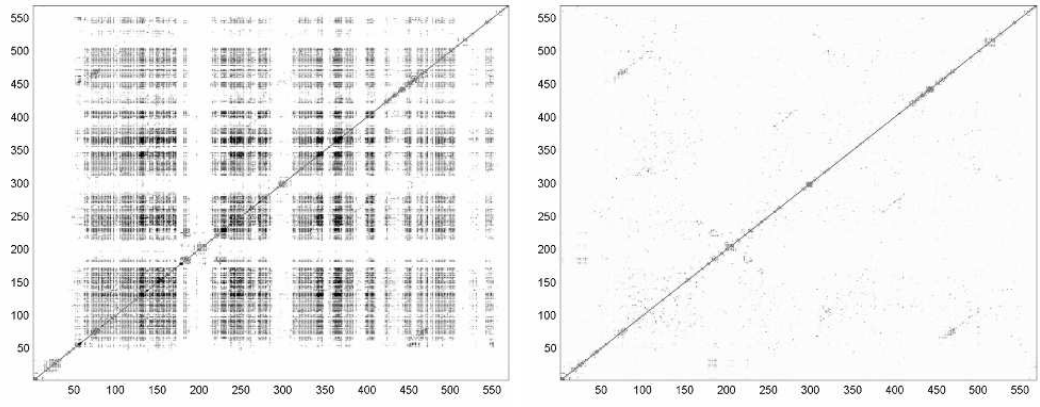
7.3 Scenario II: Outdoor Rugged Terrain Environment

A more challenging scenario is now considered; moving around through both gardens and buildings at different elevations. This setting is beyond the capabilities of the SLAM system itself and so a GPS sensor is used to provide ground truth. The experiment proceeds as before: for every metre travelled and for every 30 degrees change in heading, an image is captured. The camera toggles from 60 degrees left and right. Each image captured is time-stamped. Throughout the experiment, GPS NMEA[4] strings are logged.

An aerial image of the environment where the experiment was conducted is shown in Figure 7.18. GPS estimates of robot's position are plotted onto the image as white crosses. Due to intermittent GPS reception, certain portions of the robot's trajectory are not registered. Following the robot trajectory, it can be observed that multiple loops were made. Robot pose where loop closures are detected are marked as blue crosses. There are particularly revealing results from this experiment,

[4]National Marine Electronics Association

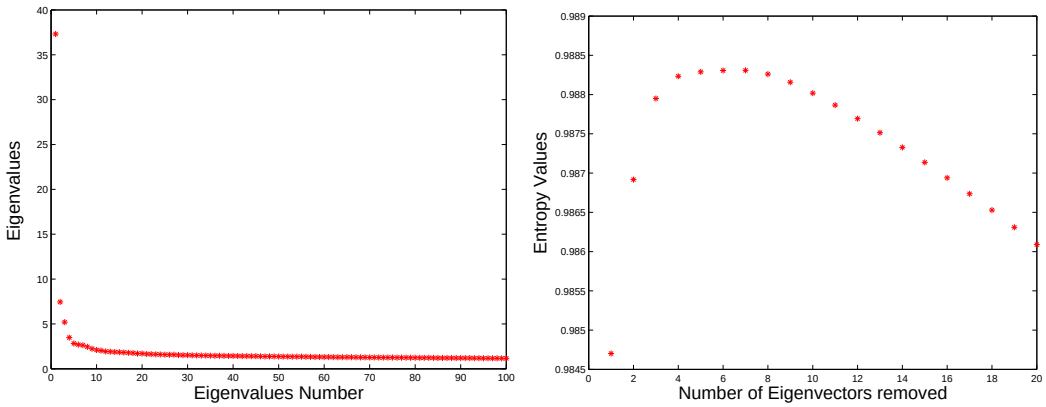
which will be explained in detail in Subsection 7.3.1 during the evaluation of performance of the loop closing detection system.



(a) (b)

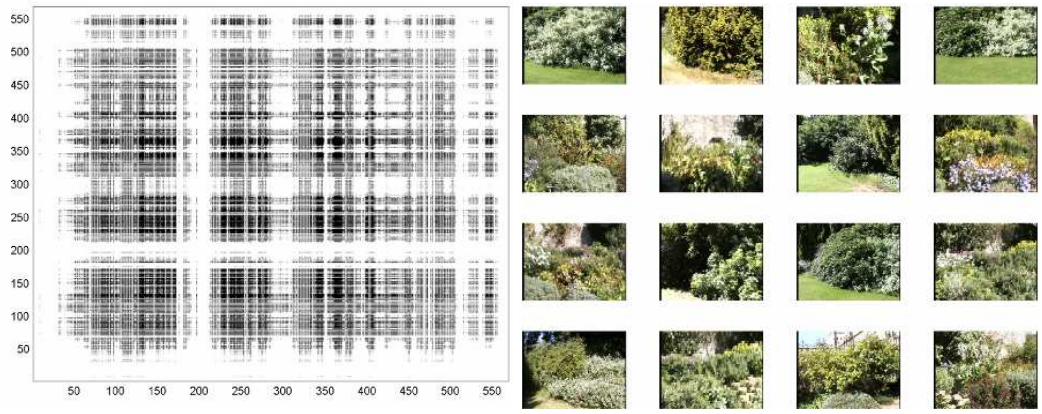
Figure 7.19: (a) shows a VSM, M^{NC}, of the New College Garden. Dark tone represents high similarity score while light tone represents low similarity scores. Note the amount of VARs within the VSM. No distinct off-diagonals can be observed from the original VSM. (b) shows the VSM after rank reduction. Note that off-diagonals (which signifies the loop closure) have become more visible by the rank reduction whereas VARs within the similarity matrix have been removed.

In all, 568 images were collected during this experiment. (See Appendix C-3). The VSM, M^{NC}, for this environment is shown in Figure 7.19(a). There is a significant amount of VARs, marked by the dark squares, within the VSM. No distinct off-diagonals can be observed from the VSM. A rank reduced matrix is shown in Figure 7.19(b) after removing the top six eigenvalues and corresponding outer products. Note that off-diagonals (which signifies the loop closure) have become more visible by the rank reduction process whereas VARs have been removed. Due to multiple loop closure at different locations, there are multiple dark off-diagonals.
In Figure 7.20, a distribution of the eigenvalues obtained from the EVD of the VSM is shown. The eigenvalues drop drastically for the first few initial values before levelling off. Figure 7.20(b) shows a typical distribution of Shannon entropy of successively rank reduced similarity matrices. The Shannon entropy reaches a peak after the reduction of the first six rank one matrices associated with the largest eigenvalues.



(a) (b)

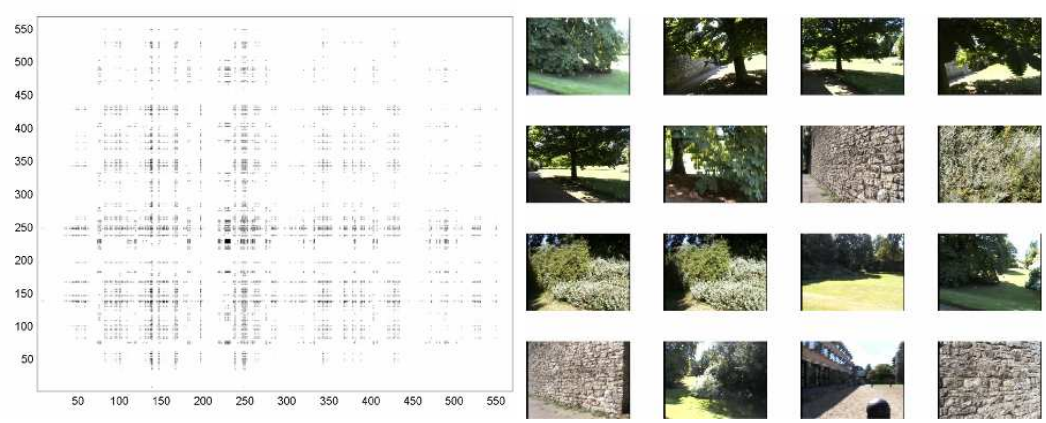
Figure 7.20: (a) shows a typical distribution of eigenvalues. This set of eigenvalues is obtained from the EVD of the VSM in Figure (7.19a). (b) shows a typical distribution of Shannon entropy of successively rank reduced similarity matrices.



(a) (b)

Figure 7.21: (a) shows rank one matrix (M_1^{NC}) of the VSM shown in figure 7.19. (b) shows 16 images associated with high scoring cells in the matrix. These images are mostly images of vegetation. Bright coloured cells are spread across the matrix.

Figure 7.21(a) shows rank one matrix, $M_1^{NC} = v_1 \lambda_1 v_1^T$. Figure 7.21(b) shows sixteen different images associated with high scoring cells in the matrix. These images have significant amount of vegetation. The decomposition has extracted the dominant 'theme' within this particular environment. It can be observed from the aerial image shown in Figure 7.18 that dense vegetation does indeed constitute a major theme within the environment.

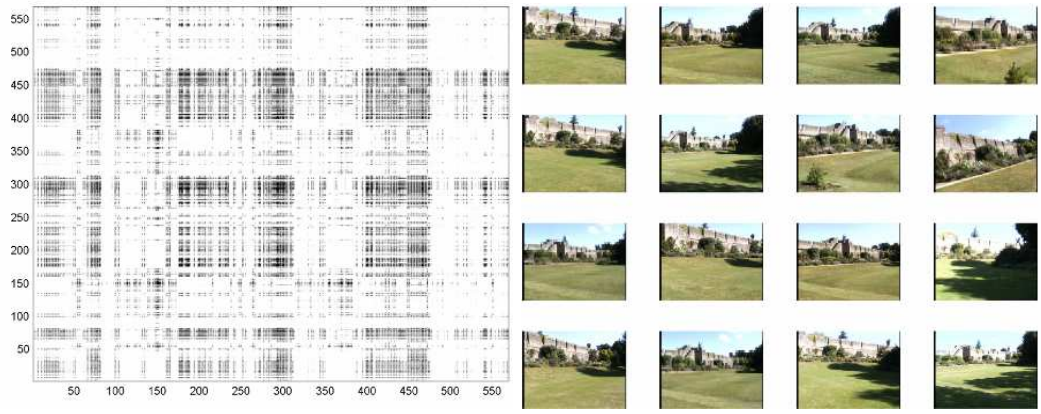


(a) (b)

Figure 7.22: (a) shows rank one matrix ($M_2^{NC} = v_2 \lambda_2 v_2^T$). (b) shows images associated with high scoring cells in the matrix. It is difficult to discern the category of images. However, it can be observed that these images contain textured material like leaves, sand grained walls and pebbles. Bright coloured cells are less spread across the matrix.

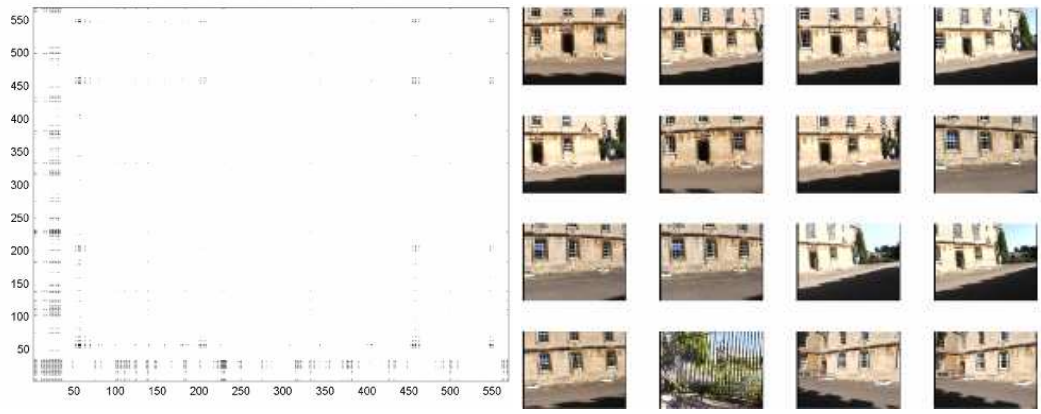
Figure 7.22(a) shows rank one matrix, M_2^{NC}. Figure 7.22(b) shows different images associated with high scoring cells in the matrix. It is harder to discern the category of images — perhaps a bias towards textured material like leaves and grainy walls. Figure 7.23(a) shows rank one matrix, M_3^{NC}. Figure 7.23(b) shows different images associated with high valued cells in the matrix. All of the photos contain images of the wall encircling the park — witnessed by the spread of dark cells throughout the matrix.

Finally, Figure 7.24(a) shows rank one matrix, M_4^{NC}. Figure 7.24(b) shows different images associated with the matrix. These are mostly images of a building seen at the start and end of the experiment. Figure 7.25(a) shows the LTM of $\tilde{M}^{NC}$ and (b) shows the significant sequences



(a) (b)

Figure 7.23: (a) shows rank one matrix (M_3^{NC}). (b) shows images associated with high scoring cells in the matrix. All of the photos contain images of wall encircling the park. Bright coloured cells are spread across the matrix.



(a) (b)

Figure 7.24: (a) shows rank one matrix (M_4^{NC}). (b) shows images associated with high scoring cells in the matrix. These are mostly images of a building. Bright coloured cells are concentrated to a small portion of the matrix.

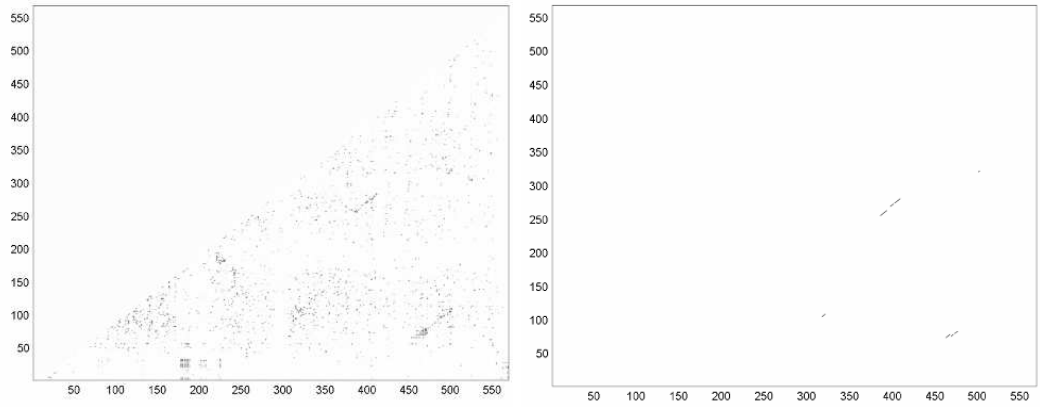


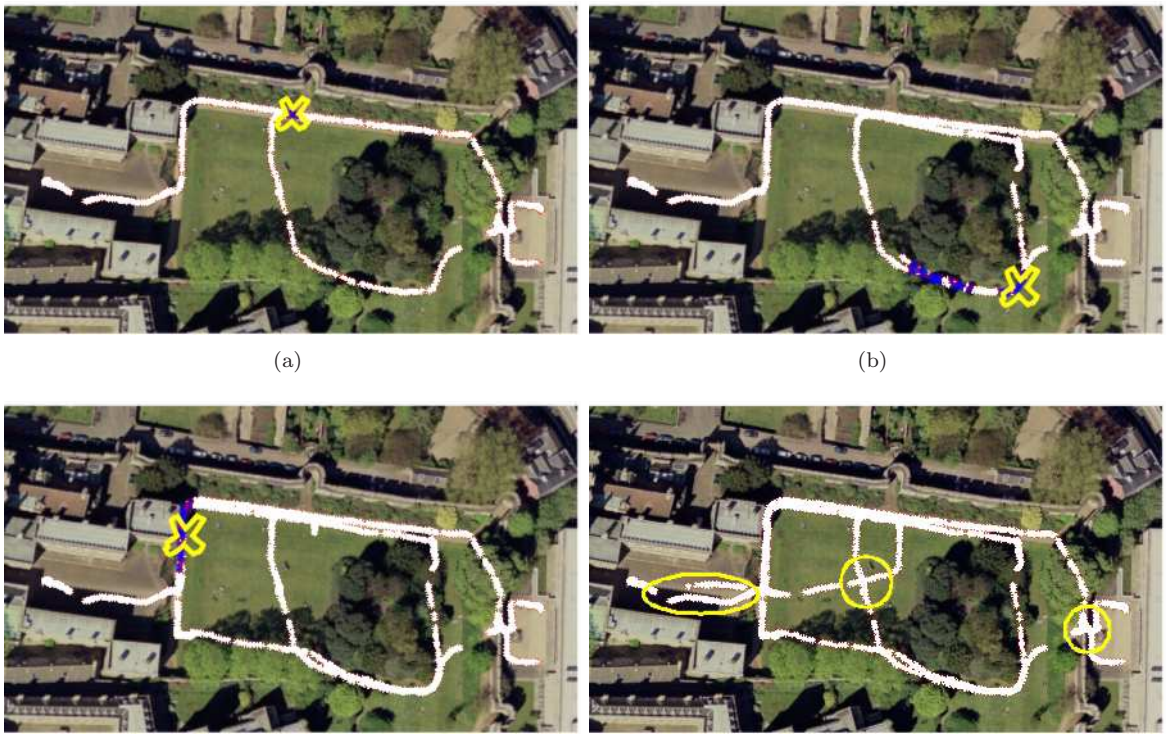
Figure 7.25: (a) is a LTM of a VSM after loop closure has occurred. The VARs has been removed through rank reduction. (b) is the result of applying the sequence detection algorithm to find a significant sequence.

extracted. One of the matching sequences, $\mathcal{A}$ and $\mathcal{B}$, is shown in Figure 7.26. Although this is a particularly challenging environment, the system is able to extract correct loop closure evidence. In the third row along the sequence, the matched images look very different due to a wide difference in viewpoint. Nevertheless the overall scene similarity accumulated along a trajectory has enough statistical significance to imply a loop-closure event.

7.3.1 Analysis

This environment is a particularly challenging one with the structure of M^{NC} being indicative of a general lack of distinctive images. Three main loops were successfully detected as depicted in Figure 7.27(a)(b)(c) as blue crosses. The detection of the second loop (b) is an excellent example of the kind of challenging detection that the method proposed in this thesis enables. Analyzing the Type II errors provides more insight into our approach. Starting with the extreme right of the aerial image shown in Figure 7.27(d), a small loop (marked by a yellow ellipse) was not detected. This is because there is not enough overlap between the second pass and the first pass. In fact, there is only one point of intersection between them at the entrance of the courtyard.

Figure 7.26: Two matching sequences of images $(\mathcal{A}, \mathcal{B})$. These images correspond to the bottom right loop closure in Figure 7.25(b). To a casual observer, it is not immediately apparent that this is indeed a loop-closure. In the third row along the sequence, the two images look very different due to wide difference in viewpoint.



(a) (b)

(c) (d)

Figure 7.27: (a) shows the first loop closure event correctly detected. Loop closure was detected right at the start of the loop. Not the entire second loop was detected. This is because most of the second loop occurred in environment that is visually confusing. Henceforth, the system does not make loop closure detection on the later part of the loop. (b) shows the second loop closure event. More loop closure events were triggered along this loop. However, there are still multiple breaks between the loop closure detection. (c) shows the third loop closure event. The first half of the loop was well detected by our loop closure detection system, with few breaks. The second half of the loop was not detected at all because that part of the environment was visually confusing. (d) highlights loop closure events that were not detected with white ellipses.

Consequently, there should only be one correct image match from that loop closure. However, the single image did not score significantly enough to trigger a loop closure. This is a limitation of our approach — a certain amount of overlap between the first pass and second pass must occur before a statistically significant alignment score, $\eta_{\mathcal{A},\mathcal{B}}$, can accumulate. The precise amount of overlap required to acquire "significance" depends on the environment itself and the numerical similarity between individual images. Naturally, more ambiguous scenes require longer sequences.
The middle yellow ellipse in Figure 7.27(d) marks another potential loop closure that was not detected. Again, there was only a small area of trajectory coincidence. A bigger problem here is that in the centre of the park all the scene diversity (the borders) is in the far field and, coupled with a 90^o difference in heading, this leads to utterly different images [5]. Finally, the yellow ellipse on the extreme left highlights the last potential loop closure that was not detected. The reason for this failure can be seen in Figure 7.24. The subtraction of M_4^{NC} removed the elements of M^{NC} indicating similarity between images of the building facades. Essentially this loop was not detected because of a high likelihood of false loop closure caused by the repetitive architecture of the building. Our policy, to support SLAM, is to strongly prefer Type II errors over Type I errors.

7.4 Scenario III: Indoor Visually Challenging Environment

In the final visual loop closing experiment, the technique is put to test in an environment where, by design, every local scene is visually similar. The question is whether our loop closure detection will fail where there are no obvious globally distinct scenes. This experiment took place in the cloister of a college. The control parameter settings for the camera were the same as the previous experiment. A sequence of 268 images was collected. (See Appendix C-4). Figure 7.28 shows a 2D laser map of the cloister and the robot trajectory. This is a fairly small environment with a loop length of around 160 metres. It can be observed that this environment is geometrically very simple and similar. It consists primarily of long, straight corridors. Figure 7.29 shows a view from inside the cloisters. It shows that the cloisters present a repetitive and ambiguous architectural theme. The VSM, $M^{Cloister}$, is shown in Figure 7.30(a). The rank reduced matrix, $\tilde{M}^{Cloister}$, is shown in

[5] a strong motivator to use an omni-cam instead of a standard pan-tilt-unit

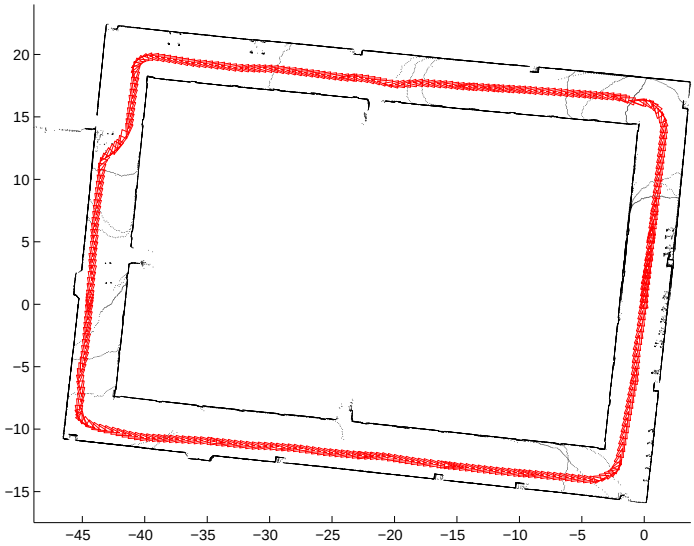
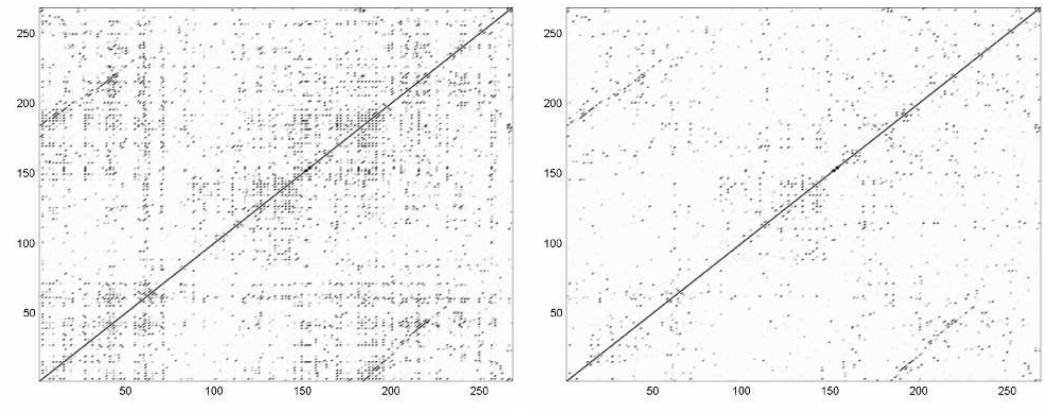


Figure 7.28: 2D laser map of the cloister and the robot trajectory

Figure 7.29: A view from inside the cloisters which by intention present a repetitive and ambiguous architectural theme.



(a) (b)

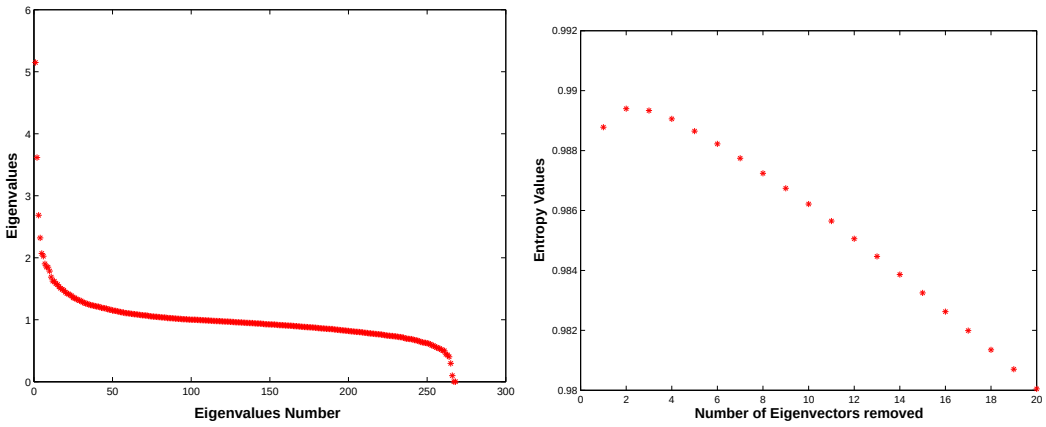
Figure 7.30: (a) shows a VSM, $M^{Cloister}$, constructed from images collected from an exploration run around the New College Cloister. (b) shows the VSM, $\tilde{M}^{Cloister}$, after rank reduction. Note that the off-diagonal bright line (which signifies the loop closure) has not been affected by the rank reduction whereas VARs within the similarity matrix have been removed.

Figure 7.30(b). Note that the off-diagonal bright line (which signifies the loop closure) has not been affected by the rank reduction whereas VARs within the similarity matrix have been removed.

7.4.1 Analysis

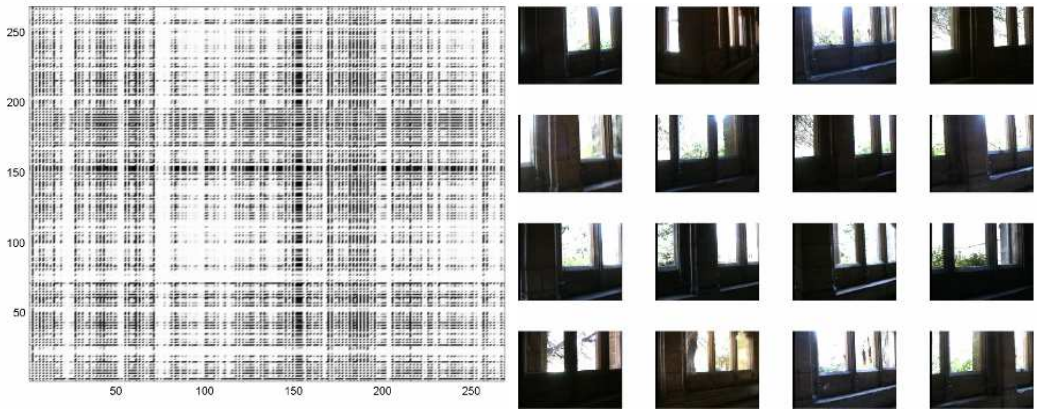
In Figure 7.31(a), a distribution of the eigenvalues obtained from the EVD of the VSM, $M^{Cloister}$, is shown. Figure 7.31(b) shows a typical distribution of Shannon entropy of successively rank reduced similarity matrices. The Shannon entropy reaches a peak after the reduction of the first two rank one matrices. This environment can be summarised as a continuous stretch of arched windows and a continuous stretch of wall. Only two principal eigenvectors were selected to be removed by our entropy maximisation method. The two outer products and corresponding images are shown in Figures 7.32 and 7.33. As expected, these correspond to the two aforementioned dominant themes.

Figure 7.34(a) shows the LTM of the VSM, $M^{Cloister}$ and the results from the sequence detection algorithm are shown in Figure 7.34(b). One of the matching sequences is shown in Figure 7.35. It



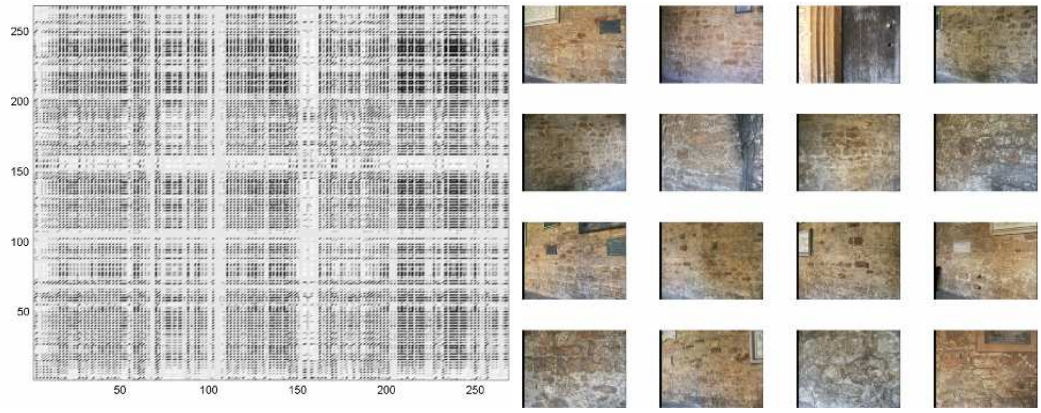
(a) (b)

Figure 7.31: (a) shows a typical distribution of eigenvalues. This distribution of eigenvalues is obtained from the EVD of the VSM in Figure (7.30a). (b) shows a typical distribution of Shannon entropy of successively rank reduced similarity matrices.



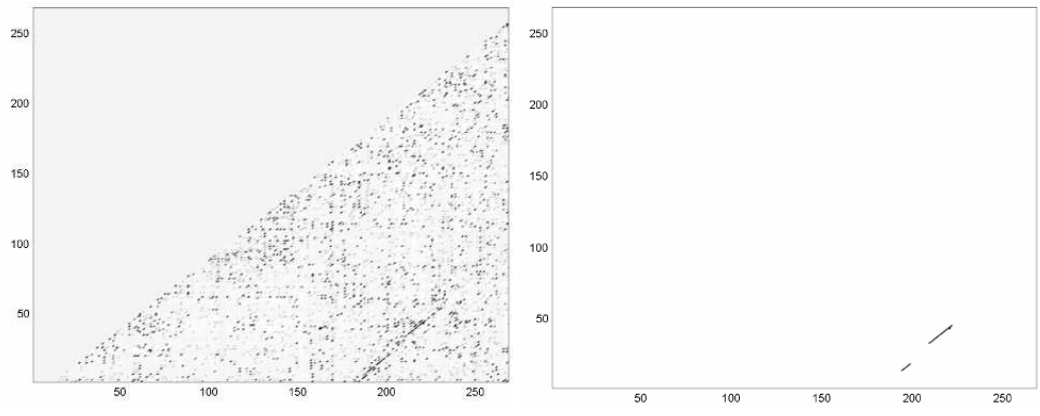
(a) (b)

Figure 7.32: (a) shows rank one matrix ($M_1^{Cloister} = v_1 \lambda_1 v_1^T$) of the VSM shown in figure 7.30. (b) shows 16 images associated with high scoring cells in the matrix. These are images of the windows. Bright coloured cells are spread across the matrix.



(a) (b)

Figure 7.33: (a) shows rank one matrix ($M_2^{Cloister} = v_2 \lambda_2 v_2^T$). (b) shows images associated with high scoring cells in the matrix. These are images of the wall. Bright coloured cells are spread across the matrix.



(a) (b)

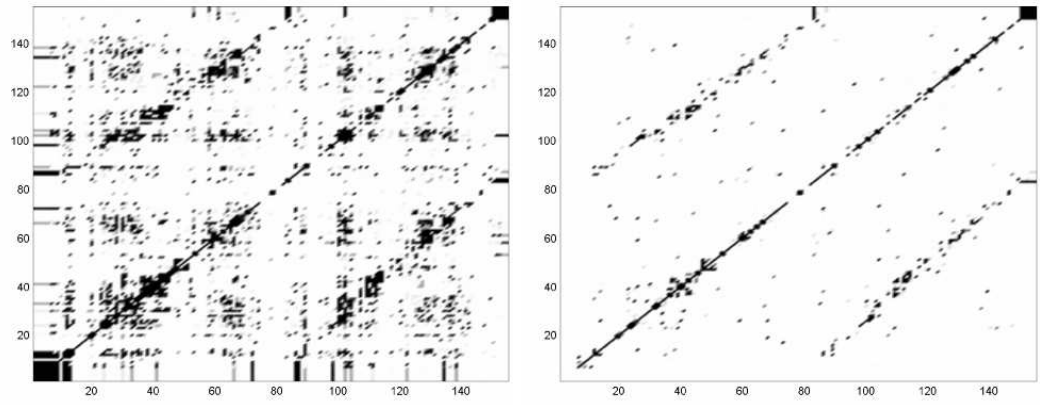
Figure 7.34: (a) is a LTM of a VSM after loop closure has occurred. The VARs has been removed through rank reduction. (b) is the result of applying the sequence detection algorithm to find significant sequences.

Figure 7.35: Top: Subset of a query subsequence of images. Bottom: Subset of a best matching subsequence of images. The degree of accuracy of our sequence matching system can be observed from this result. Only by visually inspecting each image carefully can it be discerned that the image match is correct.

can be confirmed that this is indeed a correct match by observing the presence of plaques and the occasional statue. Note that given the background of common mode similarity between images it took twelve images to accumulate enough evidence to render the alignment score, $\eta_{\mathcal{A},\mathcal{B}}$, "significant" and trigger a loop closure. The corrected map from the loop-closure is shown in Figure 7.28.

7.5 Application with Laser Images

The loop closing technique is not limited to similarity matrix for visual images. It is equally applicable for any similarity matrix which is formed by an appropriate similarity function that compares sensor observations from different local scenes. The applicability of our approach on a similarity matrix that compares similarity between laser scans (spatial images) is demonstrated in this section. Figure 7.36(a) shows a spatial similarity matrix (SSM) constructed from a sequence of laser scans collected from an exploration run around "Acland" Building in Oxford. The details of the similarity function $\mathcal{S}(L_i, L_j)$ which compares two scans L_i and L_j have been previously described in Chapter 3.



(a) (b)

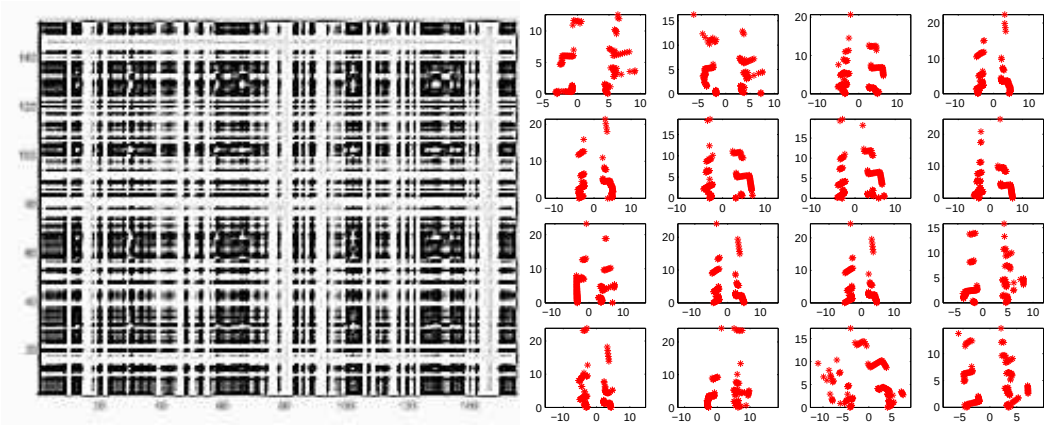
Figure 7.36: (a) shows a SSM constructed from laser scans collected from an exploration run around the Acland Building. The dark bands within the similarity matrix are due to laser scans that do not contain descriptive information. (b) shows the SSM after rank reduction. Note that the off-diagonal dark line (which signifies the loop closure) has not been affected by the rank reduction whereas VARs within the similarity matrix have been removed.

Figure 7.36(a) shows a SSM constructed from laser scans collected from an exploration run around the Acland Building. It can be seen the off-diagonals for the laser similarity matrix are less defined, reflecting the diminished certainty in matches coming from less discriminative (relative to the visual images) data. The dark bands within the similarity matrix are due to laser scans that

do not contain descriptive information. Figure 7.36(b) shows the SSM after rank reduction using our proposed approach. Note that the off-diagonal bright lines (which signify the loop closure) have not been affected much by the rank reduction whereas VARs (bottom left corner and top right corner) within the similarity matrix have been removed.

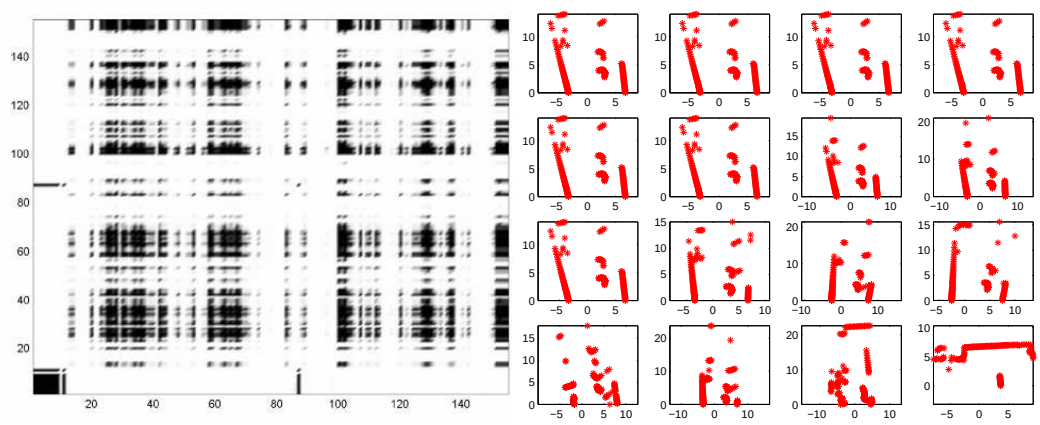
7.5.1 Spectral Decomposition of SSM

Figures (7.37)(7.38)(7.39) illustrate the rank one matrices based on the top three eigenvalues and their associated laser scans. It can be visually inspected that the groupings of laser scans generally share a common theme such as "corridor-like" or "T-shaped".



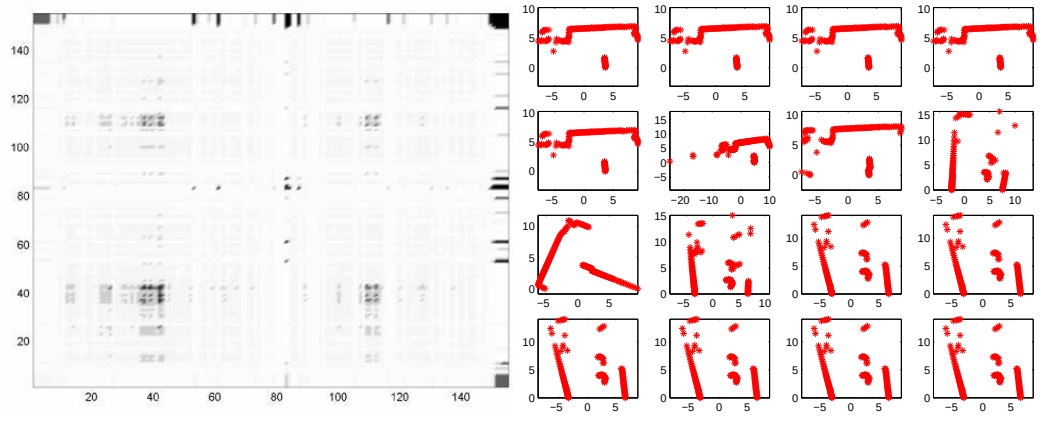
(a) (b)

Figure 7.37: (a) shows rank one matrix ($M_1^{Acland} = v_1 \lambda_1 v_1^T$) of the SSM, M^{Acland}. (b) shows 16 laser scans associated with high scoring cells in the matrix. Bright coloured cells are spread across the matrix.



(a) (b)

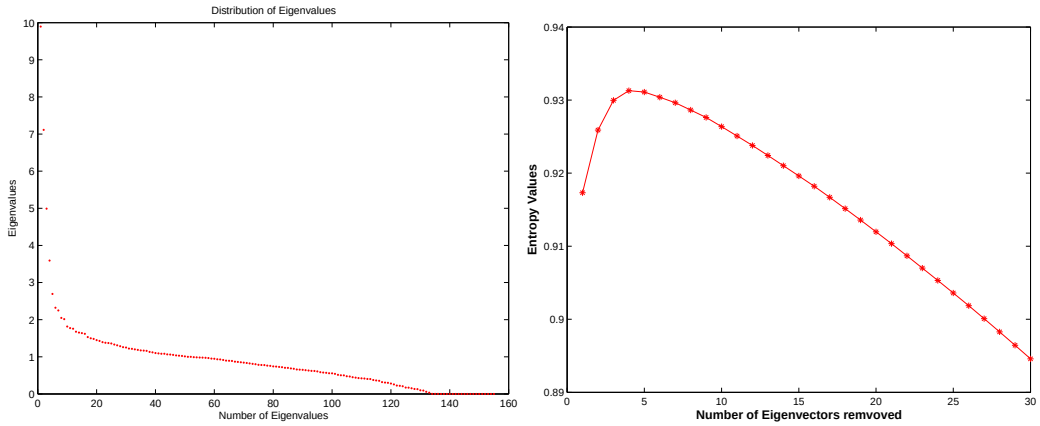
Figure 7.38: (a) shows rank one matrix (M_2^{Acland}) of the SSM, M^{Acland}. (b) shows 16 laser scans associated with high scoring cells in the matrix. These laser scans can be broadly defined as having parallel lines. Bright coloured cells are concentrated at the bottom left corner of the matrix.



(a) (b)

Figure 7.39: (a) shows rank one matrix (M_3^{Acland}) of the SSM, M^{Acland}. (b) shows 16 laser scans associated with high scoring cells in the matrix. These laser scans are broadly T-shaped. Bright coloured cells are concentrated at the top right corner of the matrix

7.5.2 Rank Reduction of SSM



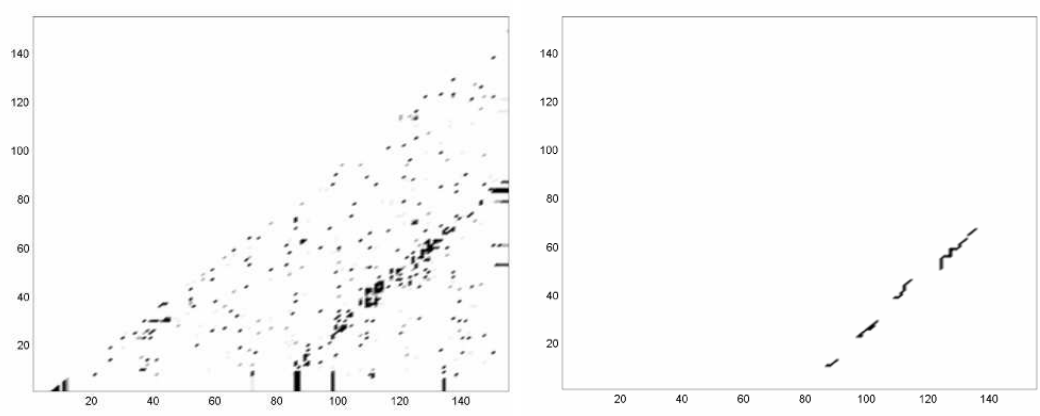
(a) (b)

Figure 7.40: (a) shows a typical distribution of eigenvalues. This distribution of eigenvalues is obtained from the EVD of the SSM in Figure (7.36a). (b) shows a typical distribution of Shannon entropy of successively rank reduced similarity matrices.

Figure 7.40(a) shows a typical distribution of eigenvalues. This distribution of eigenvalues is obtained from the EVD of the SSM in Figure 7.36(a). Figure 7.40(b) shows a typical distribution of Shannon entropy of successively rank reduced similarity matrices.

7.5.3 Robust Sequence Detection in Rank Reduced SSM

Figure 7.41(a) shows a LTM of the SSM shown in Figure 7.36(b). Figure 7.41(b) shows the result of applying the sequence detection algorithm to find significant sequences. Generally, the sequences extracted are relatively short. This is due mainly to the dark bands within the similarity matrix, whereby some scenes contain impoverished descriptors. As such, loop closing does not occur on such scenes.



(a) (b)

Figure 7.41: (a) shows a LTM of a SSM after loop closure has occurred. The VARs has been removed through rank reduction. (b) shows the result of applying the sequence detection algorithm to find significant sequences.

7.6 Summary of Experiments

In this chapter, the loop closing technique has been demonstrated to support a laser scan-matching, delayed-state EKF SLAM system in a variety of settings, using images and laser scans. In each case, the performance of the loop closing detection algorithm is analysed in the context of the SLAM problem. The loop closing technique has also been extended to a multi-robot mapping problem described in Section 5.7.

7.6.1 Timing

Table 7.2 shows the execution times[6] for the various components of the loop closure detection scheme. Results are shown for the four image data sets used throughout the work. The code was run on a 2GHz Centrino Processor (Samsung X50 Laptop) with 1GB of RAM. The nature of the SLAM algorithm employed means that loop closures can be applied between any two poses, past or present, and so the loop closure detection process need not run in tandem with the state

[6]Pass 1 refers to the initial clustering process of descriptors into clusters of visual words. Pass 2 refers to the refinement clustering process to achieve a better grouping of descriptors.

estimation. It is not expected that loop closure events to be common-place and so run-times of minutes is acceptable. However for truly large data sets, there is the risk of falling more and more behind.

		DataSet			
		Cloisters	Thom	New College	Jenkin
# Scenes		212	387	510	485
$\mathcal{V}$ Creation	Pass 1 (s)	7	27	29	134
	Pass 2 (s)	4	8	10	20
	$\lvert\mathcal{V}\rvert$ (No. of visual words)	2603	3822	3138	5982
Detection	M Creation (ms)	6	200	400	900
	Rank Reduction (s)	0	1	3	3
	EVD Estimation (s)	2	10	18	16
	Loop Extraction (ms)	4	80	90	100
	Total (s)	2	11	21	20

Table 7.2: Run times for the four different data sets described in Table 7.1.

Chapter 8

Conclusions, Summary and Future Research

To conclude this thesis, a summary of contributions is provided as well as a discussion of plans for future work.

8.1 Contributions

The main contributions of this thesis can be outlined as follows:

- ***The introduction of an appearance-based technique that detects loop closing*** without recourse to SLAM estimates. This is achieved by encoding similarity relationships between local scenes in a similarity matrix. In this work, every observation captured at a local scene is stored into a database as a set of descriptors. Each newly captured observation is then compared against every other observation stored in the database based on a similarity function (see Chapter 3). The loop closing problem is then posed as the problem of extracting significant sequences within a similarity matrix.

- ***It is observed that loop closing events appears as off-diagonals within a*** similarity matrix when a robot retraces its previous path. The topological links between local scenes within a similarity matrix are exploited to inhibit false positives. Instead of detecting

matching pair of observations, matching sequences are detected. Such sequences are extracted via a dynamic programming algorithm described in Chapter 5. The role of multimodal sensing in tackling the perceptual aliasing problem in loop closing is briefly investigated. Sequences were detected in a combined similarity matrix that is constructed from a combination of similarity comparison of heterogenous observations. In addition, a solution to a map joining application is achieved through the utilisation of the sequence detection algorithm.

- ***It is demonstrated when decisions are made within context of the environment*** explored, loop closure detection is more robust. The common-mode similarities of an environment are extracted through spectral decomposition of a similarity matrix (see Chapter 6). The effects of ambiguous artefacts are subsequently removed through a principled manner of rank reduction based on the entropy maximisation citerion. Sequence detection is then applied on the rank reduced similarity matrix. In various experiments, the approach was demonstrated to work for both visual and spatial similarity matrices. In additional, a probabilistic assessment of such sequences (loop closing events) occurring randomly by chance is provided. Consequently, loop closing is only triggered if the maximal alignment score associated with a sequence is considered statistically significant.

- ***Extensive results were obtained from implementing the loop closing technique*** with a laser scan-matching, delayed-stated EKF SLAM, as described in Chapter 7. Data was collected from various environments (see Appendix C). The validity of loop closing detection results was observed from the comparison of observations within matching sequences and the quality of geometric maps built. In one particular experiment, ground truth was provided from GPS fixes. The sizes of loops varied from 100 metres in length to over 400 metres and multiple loop detections were made in one experiment. The limitations of the technique were highlighted in the analysis of performance for each experiment.

For the first time, appearance techniques have been used to tackle the SLAM loop closing problem while addressing the key issues of:

- perceptual aliasing
- quantitative error analysis
- applicability across sensor modality.

8.2 Future Research and Improvements

Future areas of research that can improve the performance of the loop-closure detection algorithm are listed below:

- One way to enhance the performance of the loop closing algorithm is simply to use panoramic images to represent local scene. A panoramic image will provide rotational invariance to the field of view. Currently, a limited view (48.8^o) of the local scene is captured even though the camera is programmed to capture images from both sides. Similarly, the range of view of the laser scanner is limited to 180^o. This is particularly important when a robot is traversing the same area in the opposite direction. Having a panoramic view will ensure that all available sensory data (photometric and geometric information) is used to represent the local scene and this will enhance matching performance.
- As discussed in Section 7.3, a limitation of our approach is that a certain amount of overlap between the first pass and second pass must occur before a statistically significant alignment score, $\eta_{\mathcal{A},\mathcal{B}}$, can accumulate. Consequently, the proposed loop closing algorithm might not be able to detect loop closing at intersections of robot's trajectories. This is a problem which may possibly be mitigated with clever path planning. At each possible intersections (such as a junction in the hallways), a robot may be programmed to investigate all routes to search for possible loop closures before continuing exploration. Though the proposed loop closing algorithm is independent of robot pose estimate, robot pose estimate can nevertheless be used to help in path planning so as to optimize the chances of loop closing.
- The size of a similarity matrix grows quadratically with the number of local scenes stored. In

landmark-based EKF SLAM, the size of the state vector remains constant when a robot revisits a previously mapped environment and associate observed landmarks with landmarks stored. However, the size of similarity matrix continues to grow. This inadvertently results in heavier computation complexity as a robot continues to revisit previously visited locations. This is especially important since eigenvalue decomposition has a computational complexity of $O(N^3)$. It is important that a method of restructuring the similarity matrix be devised to enable a robot to operate in mapped environments.

- Currently, this work implicitly assumes that the environment does not change significantly during the time interval a robot takes to revisit the same location. This assumption is generally true for most cases of loop closing in environment up to several hundreds metres in size. However in order for mobile robots to operate for long durations (days or weeks), the maps have to be continuously updated to reflect long term changes [8]. Moreover, dynamic objects are sometimes present when sensory observations are registered from a local scene. Effects from such dynamic objects should be removed from the description of a local scene upon revisiting and re-observation at the same location.

Substantial progress in our understanding of the SLAM problem and in the development of efficient and robust SLAM algorithms have been achieved in the past decade. However, successful demonstrations have been limited in the scale and structure of the environments tested. The future trend will be towards more convincing demonstrations of SLAM implementations in more challenging environment of grander scale over a longer duration. The framework put in place in this thesis gives a strong basis in enhancing the robustness of SLAM systems in handling large loop closing, which will eventually enable the mapping of larger environments of increasing complexity.

Appendix A

Probabilistic Formulation of SLAM

The basic statistical framework of the SLAM problem is introduced here. Let $\mathbf{x}$ denote robot pose in $x - y - \theta$ space. A discrete time model of evolution of robot pose and observations is adopted, $k = 1, 2, \cdots$. A robot pose at time t_k is expressed as $\mathbf{x}_k$. Without loss of generality, pose $\mathbf{x}_0$ is defined to be the origin of the coordinate system with a heading direction of 0 degrees such that $\mathbf{x}_0 = [0, 0, 0]^T$. Let $\mathbf{u}_k$ denote a control input applied at time t_{k-1} to drive a robot from $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$. The control input consists of a combination of rotational and translational motion. The probabilistic models of motion and perception are now discussed.

Robot motion

Since robot motion is inaccurate, the effect of a control $\mathbf{u}_k$ on the robot pose $\mathbf{x}_{k-1}$ is modelled by a conditional probability density as follows:

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k) \tag{A.1}$$

which describes the position of the vehicle at $\mathbf{x}_k$ given a previous pose $\mathbf{x}_{k-1}$ and a control input $\mathbf{u}_k$ is executed at time t_k.

The probability distribution for robot pose $p(\mathbf{x}_k)$ after executing control input $\mathbf{u}_k$ is as follows:

$$p(\mathbf{x}_k|\mathbf{u}_k) = \int p(\mathbf{x}_k|\mathbf{u}_k, \mathbf{x}_{k-1})p(\mathbf{x}_{k-1})d_{\mathbf{X}_{k-1}} \tag{A.2}$$

where $p(\mathbf{x}_{k-1})$ is the probability distribution for robot pose before executing $\mathbf{u}_k$ and η is a normaliser.

Robot Perception

Let $\mathbf{z}_k$ denote a measurement at time t_k. Let $\mathbf{m}$ denote a map. The model of robot observation is modelled by a conditional probability

$$p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m}) \tag{A.3}$$

The probability that a robot is at pose $\mathbf{x}_k$ when $\mathbf{z}_k$ is observed is given by:

$$p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{m}) = \eta\ p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m})p(\mathbf{x}_k|\mathbf{m}) \tag{A.4}$$

where $p(\mathbf{x}_k|\mathbf{m})$ measures the probability the robot is at pose $\mathbf{x}_k$ prior to observing $\mathbf{z}_k$ and η is a normaliser.

Let the history of poses be $\mathbf{x}^k = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k\} = \{\mathbf{x}^{k-1}, \mathbf{x}_k\}$. Let the history of control inputs be $\mathbf{u}^k = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\} = \{\mathbf{u}^{k-1}, \mathbf{u}_k\}$. Let the history of measurements be $\mathbf{z}^k = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\} = \{\mathbf{z}^{k-1}, \mathbf{z}_k\}$.

It is reasonable to assume conditional independence:

$$p(\mathbf{z}^k|\mathbf{x}^k, \mathbf{m}) = \prod_{i=1}^{k} p(\mathbf{z}_i|\mathbf{x}^k, \mathbf{m}) = \prod_{i=1}^{k} p(\mathbf{z}_i|\mathbf{x}_i, \mathbf{m}) \tag{A.5}$$

Expand the joint distribution in terms of the state

$$p(\mathbf{x}_k, \mathbf{m}, \mathbf{z}_k|\mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{x}_0) = p(\mathbf{x}_k, \mathbf{m}|\mathbf{z}_k, \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{x}_0)p(\mathbf{z}_k|\mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{x}_0) \tag{A.6}$$

$$= p(\mathbf{x}_k, \mathbf{m}|\mathbf{z}^k, \mathbf{u}^k, \mathbf{x}_0)p(\mathbf{z}_k|\mathbf{z}^{k-1}, \mathbf{u}^k) \tag{A.7}$$

and the observation

$$p(\mathbf{x}_k, \mathbf{m}, \mathbf{z}_k | \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{x}_0) = p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}, \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{x}_0) p(\mathbf{x}_k, \mathbf{m} | \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{x}_0) \tag{A.8}$$

$$= p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) p(\mathbf{x}_k, \mathbf{m} | \mathbf{z}_{k-1}, \mathbf{u}^k, \mathbf{x}_0) \tag{A.9}$$

Rearranging:

$$p(\mathbf{x}_k, \mathbf{m} | \mathbf{z}_k, \mathbf{u}^k, \mathbf{x}_0) = \frac{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) p(\mathbf{x}_k, \mathbf{m} | \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{x}_0)}{p(\mathbf{z}_k | \mathbf{z}^{k-1}, \mathbf{u}^k)} \tag{A.10}$$

Adopting Markov assumption in which current state is conditionally dependent only on the previous state:

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) = p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{x}^{k-2}, \mathbf{u}^{k-1}, \mathbf{m}) \tag{A.11}$$

then the time-update form is as follows:

$$p(\mathbf{x}_k, \mathbf{m} | \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{x}_0) = \int p(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{m} | \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{x}_0) d\mathbf{x}_{k-1} \tag{A.12}$$

$$= \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{m}, \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{x}_0) p(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{z}^{k-1}, \mathbf{u}^k, \mathbf{x}_0) d\mathbf{x}_{k-1} \tag{A.13}$$

$$= \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) p(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{z}^{k-1}, \mathbf{u}^{k-1}, \mathbf{x}_0) d\mathbf{x}_{k-1} \tag{A.14}$$

This leads to the recursive estimator which is is central to virtually all SLAM algorithms:

$$p(\mathbf{x}_k, \mathbf{m} | \mathbf{z}^k, \mathbf{u}^k, \mathbf{x}_0) = \eta. \underbrace{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m})}_{measurement\ model} \int \underbrace{p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k)}_{motion\ model} \underbrace{p(\mathbf{x}_{k-1}, \mathbf{m} | \mathbf{z}_{k-1}, \mathbf{u}^{k-1}, \mathbf{x}_0)}_{previous\ estimate} d\mathbf{x}_{k-1} \tag{A.15}$$

Appendix B

Sensors

The primary sensors equipped onboard the ATRV-JR autonomous vehicle are an EVI-D30 camera and a SICK laser range-finder (See Figure B.1).

(a) Camera and laser scanner mounted on ATRV-Jr robot

(b) Side view of Camera and laser scanner

(c) Front view of Camera and laser scanner

Figure B.1: An EVI-D30 camera is mounted directly above a SICK LMS-200 laser range-finder. The laser scanner is mounted on a nodding mechanism which rotates the laser scanner to produce a 3D laser scan.

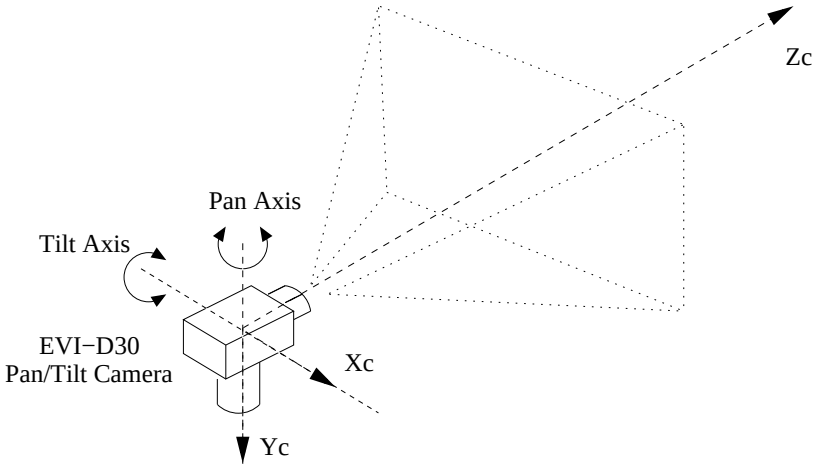


Figure B.2: Coordinate System of Pan/Tilt camera

B.1 EVI-D30 CCD camera

The EVI-D30 CCD camera (shown in Figure B.1) is a pan/tilt/zoom video camera. Commands to set parameters for pan, tilt and zoom can be transmitted via RS232 using Video System Control Architecture (VISCA). The zoom option is not exercised. VISCA is a communications protocol designed to interface video equipment with a PC. The camera produces images of maximum resolution of 768(H)x492(V). It has a horizontal pan angle of ± 100 degrees and a vertical tilt angle of ± 25 degrees. Its horizontal field of view can be adjusted from the range of 4.4^o to 48.8^o. The EVI-D30 camera is mounted above the laser scanner as depicted in Figure (B.1).

B.2 SICK LMS 200 Laser Range-finder

The laser range-finder employed on the ATRV-JR autonomous vehicle is the SICK LMS-200 laser scanner (the blue box in Figure B.1). The range measurements are based on time-of-flight measurements. A short laser (infra-red) pulse is sent out and reflected by an object surface. The elapsed time difference between emission and reception of a laser pulse is used to calculate the distance between the object and the laser scanner. The laser pulse is transmitted via an integrated

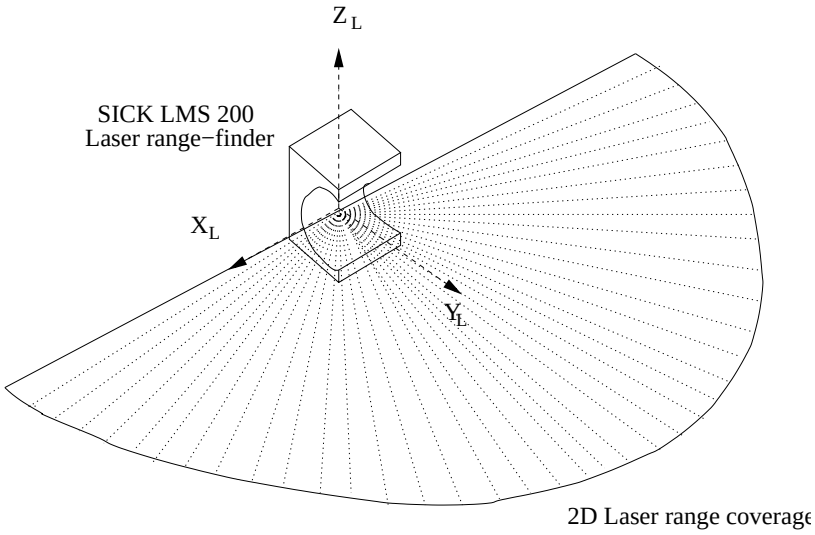


Figure B.3: Coordinate System of laser range-finder

rotating mirror that sweeps a radial range in front of the laser scanner. The laser scanner sweeps from right to left. Each measurement is based on a particular angular position. There are two modes of angular range; 180^o or 100^o. Angular resolution is also given in two modes; 0.5^o or 1^o. For implementation, the angular range is set to 180^o and angular resolution to 1^o. The measurement range is also available to two modes; mm mode or cm mode. For the mm mode, the detection range is 8.191 metres while the for the cm mode, the detection range is 81.91 metres.

B.3 Camera Projection

An image is a $2D$ representation of a $3D$ world. The process of projecting a $3D$ scene point onto a $2D$ image point is modelled by a pinhole camera model, illustrated in Figure B.4. A ray of light is drawn from a $3D$ scene point, $\mathbf{X}$ to the center of projection, $\mathbf{C}$. The line will intersect the image plane. The intersection point on the image plane represents the image of the point.
Let a scene point, $\mathbf{X}$, be modelled in terms of $(X, Y, Z)^T$ and an image point, $\mathbf{x}$, modelled in terms of $(x, y, w)^T$. Let the center of projection be at the origin $(0, 0, 0, 1)^T$. The mapping process of a

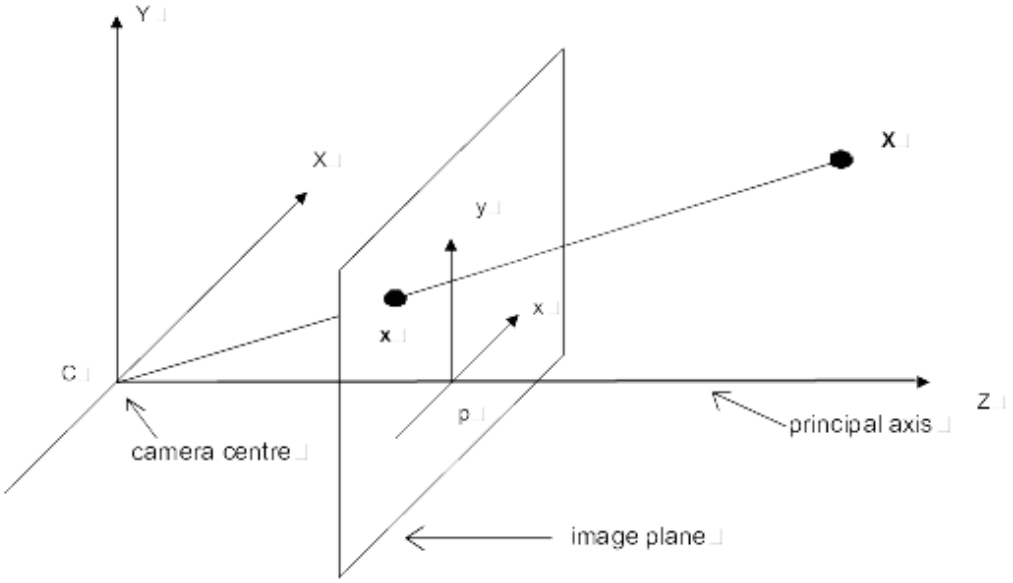


Figure B.4: With respect to the pinhole camera model, a point in space with $\mathbf{X}=(X,Y,Z)^T$ is mapped to the point, $\mathbf{x}=(x,y,w)^T$, on the image plane where w is a scale factor. A line joining the point $\mathbf{X}$ to the center of projection, $\mathbf{C}$, meets the image plane at point $\mathbf{x}$.

3D scene point to a 2D image point can be represented by a projective relationship, $\mathbf{x} = P\mathbf{X}$. P is a 3×4 projective matrix, represented by $P = K[R|t]$. K is the camera calibration matrix. R is the rotation matrix and t is the translation matrix. $t = -RC$ where C is the camera center.

B.3.1 Fundamental matrix

The fundamental matrix is the algebraic representation of epipolar geometry. For each point $\mathbf{x}$ in one image, there exists a corresponding epipolar line $\mathbf{l}'$ in the other image in which point $\mathbf{x}'$ matching point $\mathbf{x}$ must lie on. It is assumed that image point correspondences have been determined.

The fundamental matrix, F, is defined by the following equation:

$$\mathbf{x}'^T F \mathbf{x} = 0 \tag{B.1}$$

The fundamental matrix can be determined given a set of at least seven $2D$ to $2D$ point correspondences, $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. Each point match gives rise to one linear equation in the unknown entries of F. The equation corresponding to a pair of points $(x, y, 1)$ and $(x', y', 1)$ is as follow:

$$x'xf_{11} + x'yf_{12} + x'f_{13} + y'xf_{21} + y'yf_{22} + y'f_{23} + xf_{31} + yf_{32} + f_{33} = 0 \tag{B.2}$$

From a set of n point matches, a set of linear equations of the form is obtained:

$$Af = \begin{bmatrix} x'_1x_1 & x'_1y_1 & x'_1 & y'_1x_1 & y'_1y_1 & y'_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_nx_n & x'_ny_n & x'_n & y'_nx_n & y'_ny_n & y'_n & x_n & y_n & 1 \end{bmatrix} f = 0 \tag{B.3}$$

where f is the 9-vector made up of the entries of F in row-major order. f is solved in the similar fashion as homography in the previous section, using single value decomposition to find the unit singular vector with the smallest singular value. A review of best practices in least square estimation methods can be found in [128].

B.3.2 Essential matrix

The essential matrix is the specialization of the fundamental matrix [52, 129]. If the calibration matrix K is known, then image point $\hat{\mathbf{x}}$ expressed in normalized image coordinates can be obtained through $\hat{\mathbf{x}} = K^{-1}\mathbf{x}$

The essential matrix has the form:

$$E = [t]_x R \tag{B.4}$$

The defining equation for the essential matrix is:

$$\hat{\mathbf{x}}'^T E \hat{\mathbf{x}} = 0 \tag{B.5}$$

$$\hat{\mathbf{x}}'^T K'^{-T} E K^{-1} \hat{\mathbf{x}} = 0 \tag{B.6}$$

The relationship between the essential matrix and fundamental matrix is:

$$E = K'^T F K \tag{B.7}$$

A real non-zero 3×3 matrix E is an essential matrix if and only if it satisfies the equation:

$$EE^T E - \frac{1}{2} trace(EE^T)E = 0 \tag{B.8}$$

This property will help us recover the essential matrix.

B.3.3 Five Point Solution

A "five point" solution [100] to determine the essential matrix is briefly described here. Refer to [100] for a more detailed explanation of the implementation. Each of the five point correspondences give rise to a constraint of the form (B.5). This constraint can also be written as

$$\widetilde{x}^T \widetilde{E} = 0 \tag{B.9}$$

where $\widetilde{x} \equiv [x_1x'_1 \ x_2x'_1 \ x_3x'_1 \ x_1x'_2 \ x_2x'_2 \ x_3x'_2 \ x_1x'_3 \ x_2x'_3 \ x_3x'_3]^T$ and
$\widetilde{E} \equiv [E_{11} \ E_{12} \ E_{13} \ E_{21} \ E_{22} \ E_{23} \ E_{31} \ E_{32} \ E_{33}]^T$
By stacking the vector $\widetilde{x}^T$ for all five points, a 5×9 matrix obtained. Four vectors $\widetilde{X}, \widetilde{Y}, \widetilde{Z}, \widetilde{W}$ that span the right nullspace of this matrix can now be computed by singular value decomposition.
The main computation steps of the algorithm outlined are as follows:

1. Extraction of the nullspace of a 5×9 matrix.
2. Expansion of the cubic constraints
3. Gauss-Jordan elimination with partial pivoting on a 10×20 matrix.
4. Expansion of the determinant polynomial of the 3×3 polynomial matrix B to obtain the tenth degree polynomial

5. Extraction of roots from the tenth degree polynomial

6. Recovery of R and t corresponding to each real root and point triangulation for disambiguation.

Only the relative positions of the points and camera can be recovered. The overall scale of the configuration can never be recovered solely from images. This is where range information from the laser scanner can resolve the overall scale problem.

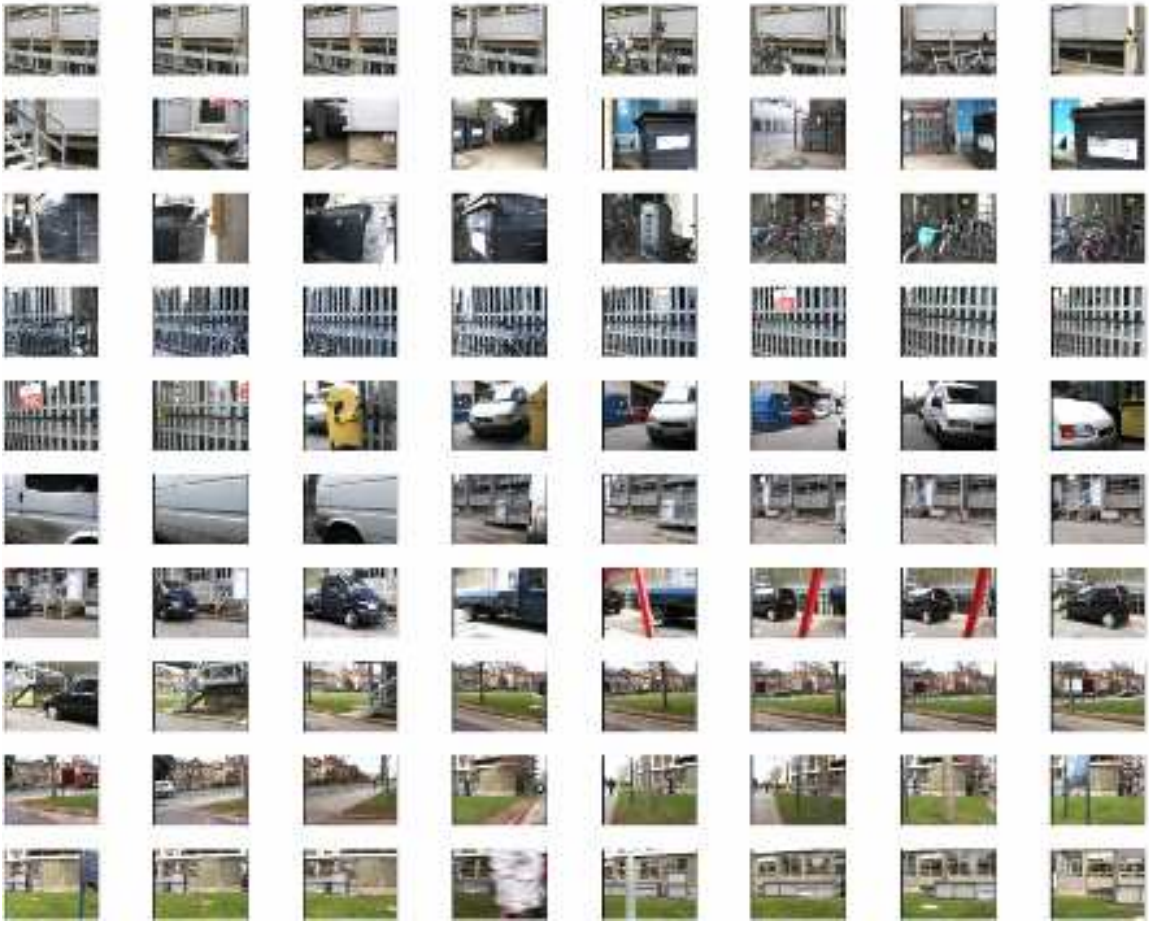
B.3.4 Extraction of relative camera positions from the essential matrix

Once the essential matrix has been determined, the camera matrices and consequently, the rotation and translation matrices, can be retrieved from the essential matrix according to Equation B.4. The essential matrix can be computed directly using the normalized image coordinates from Equation B.5 or it can be computed from the fundamental matrix using Equation B.7. There are four possible solutions, except for overall scale, which cannot be determined. Testing with a single point to determine if it is in front of both cameras (cheirality constraint) is sufficient to decide between the four possible solutions. The problem of determining the overall scale will be solved by the incorporation of range information from the laser scanner and has been described in subsection 7.2.3

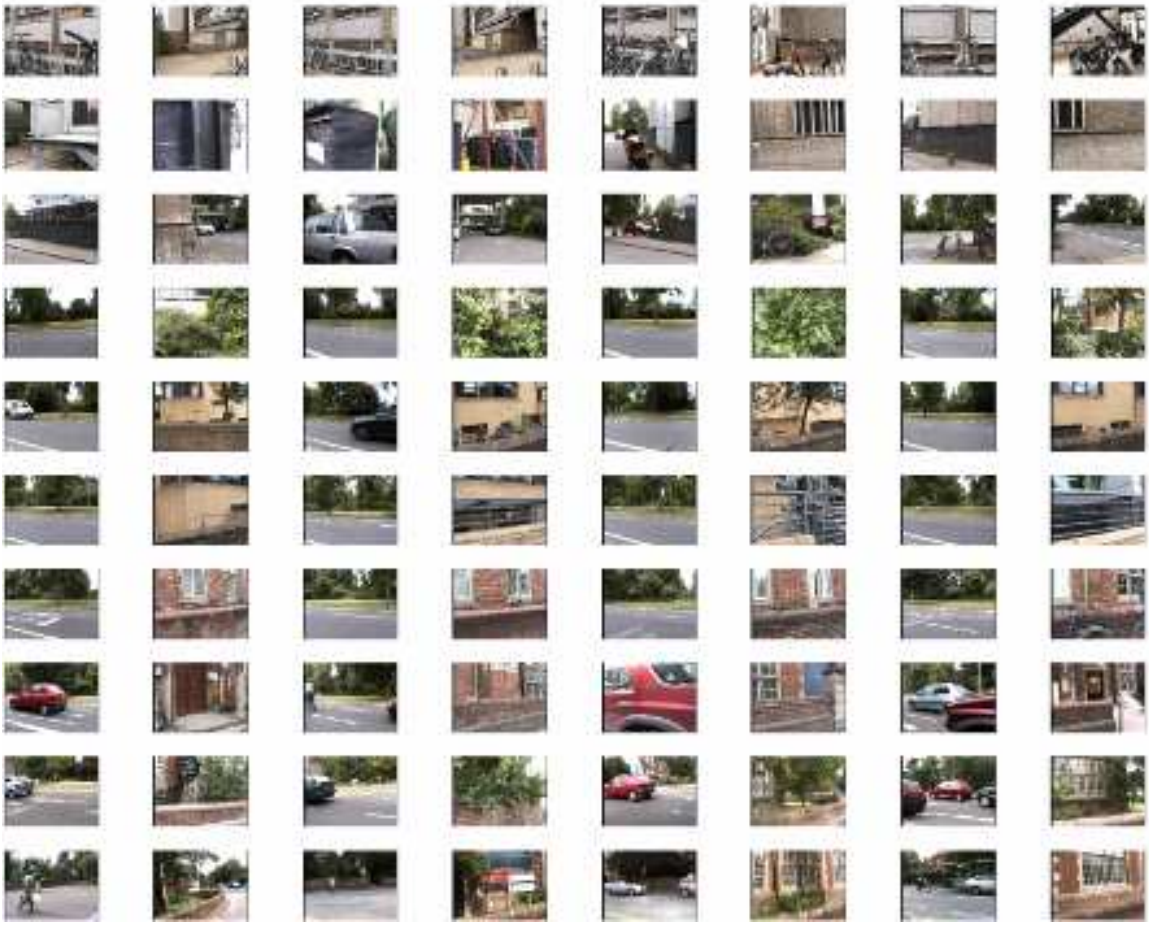
Appendix C

Image Datasets

C.1 Thom Building Sample Dataset



C.2 Jenkin Building Sample Dataset



C.3 New College Park Sample Dataset

C.4 New College Cloister Sample Dataset

Bibliography

[1] S. Altschul and B. Erickson. Significance of Nucleotide Sequence Alignments: A Method for Random Sequence Permutation That Preserves Dinucleotide and Codon Usage. *Molecular Biology and Evolution*, 2:526–532, 1985.

[2] T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, Australian Center for Field Robotics, University of Sydney, 2002.

[3] Y. Bar-Shalom and T. Fortmann. *Tracking and Data association*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.

[4] K. Beevers and W. Huang. Loop Closing in Topological Maps. In *Proceedings of International Conference on Robotics and Automation*, pages 4367–4372, Barcelona, Spain, 2005.

[5] J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *International Conference on Computer Vision and Pattern Recogniton*, pages 1000–1006, San Juan, Puerto Rico, 1997.

[6] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligience*, 24(24):509–521, April 2002.

[7] J. Bentley. Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM*, 18(9):509–517, 1975.

[8] P. Biber and T. Duckett. Dynamic Maps for long-term Operation of Mobile Service Robots. In *Proceedings of Robotics Science and Systems*, Cambridge, USA, 2005.

[9] M. Bosse, P. Newman, J. J. Leonard, and S. Teller. SLAM in Large-scale Cyclic Environments using the Atlas Framework. *International Journal of Robotics Research*, 23(12):1113–1139, 2004.

[10] C. Buckley. *Implementation of the SMART Information Retrieval System*. Cornell University, Ithaca, NY, USA, 1985.

[11] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating absolute position of a mobile robot using position probability grid. *Proceedings of Fourteen National Conference on Artificial Intelligence*, 1996.

[12] Y. Caspi and M. Irani. A step towards Sequence-to-sequence Alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 682–689, Hilton Head, SC, USA, 2000.

[13] Y. Caspi and M. Irani. Spatio-Temporal Alignment of Sequences. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(11):1409–1424, 2002.

[14] J. Castellanos, J. Niera J. Montiel, and J. Tardos. A Probabilistic Framework for Simultaneous Localization and Map Building. *IEEE Transactions on Robotics and Automation*, 15:948–953, 1999.

[15] E. Castillo. *Extreme Value Theory in Engineering*. London Academic Press, 1988.

[16] C. Chen and H. Wang. Appearance-Based Topological Bayesian Inference for Loop-Closing Detection in Cross-Country Environment. *International Conference on Intelligent Robots and Systems*, 2005.

[17] H. Choset and K. Nagatani. Topological Simultaneous Localization and Mapping (SLAM): Toward Exact Localization without Explicit Localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, April 2001.

[18] S. Cohen and L. Guibas. Partial Matching of Planar Polylines Under Similarity Transformations. In *Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 777–786, New Orleans, Louisiana, USA, January 1997.

[19] D. Cole and P. Newman. Using Laser Range Data for 3D SLAM in Outdoor Environments. In *Proceedings of International Conference on Robotics and Automation*, pages 1556–1563, Orlando, Florida, USA, 2006.

[20] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms (Second Edition)*. MIT Press and McGraw-Hill, 2002.

[21] G. Dedeoglu and Sukhatme G. Landmark-based Matching Algorithm for Cooperative Mapping by Autonomous Robots. In *Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems*, pages 251–260, Knoxville, TN, USA, 2000.

[22] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by Latent Semantic Analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

[23] G. Dudek and D. Jugessur. Robust Place Recognition using Local Appearance based Methods. *Proceedings of IEEE International Conference in Robotics and Automation, San Francisco, CA*, pages 466–474, April 2000.

[24] Y. Dufournaud, C. Schmid, and R.Horaud. Matching Images with Different Resolution. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 612–619, Hilton Head, SC, USA, 2000.

[25] A. Eliazar and R. Parr. DP-SLAM 2.0. In *International Conference on Robotics and Automation*, pages 1314–1320, New Orleans, Louisiana, USA, 2004.

[26] S. Engelson. Using Image Signatures for Place Recognition. *Pattern Recognition Letters*, 19:941–951, 1998.

[27] C. Estrada, J. Niera, and J. Tardos. Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments. *International Transactions on Robotics*, 21(4):588–596, 2005.

[28] R. Eustice, O. Pizarro, and H. Singh. Visually augmented navigation in an unstructured environment using a delayed state history. In *Proceedings of International Conference on Robotics and Automation*, pages 25–32, New Orleans, Louisiana, USA, 2004.

[29] R. Eustice, H. Singh, and J. Leonard. Exactly Sparse Delayed-State Filters. In *Proceedings of International Conference on Robotics and Automation*, pages 1100–1114, Barcelona, Spain, 2005.

[30] H. Feder, J. Leonard, and C. Smith. Adaptive Mobile Robot Navigation and Mapping. *International Journal of Robotics Research*, 18(7):650–668, 1999.

[31] J. Fenwick, P. Newman, and J. Leonard. Cooperative Concurrent Mapping and Localization. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 1810–1817, Washington, DC, USA, May 2002.

[32] A. W. Fitzgibbon. Robust Registration of 2D and 3D Point Sets. In *Proceedings of the British Machine Vision Conference*, pages 662–670, Manchester, UK, 2001.

[33] J. Foote and M. Cooper. Media Segmentation using Self-Similarity Decomposition. In *Proceedings of SPIE Storage and Retrieval for Multimedia Databases*, volume 5021, pages 167–175, San Diego, CA, USA, 2003.

[34] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A Probabilistic Approach to Collaborative Multi-robot Localization. *Autonomous Robots*, 8(3):325–344, 2000.

[35] F. Fraundorfer, H. Bischof, and S. Ober. Towards Robot Localization using Natural, Salient Image Patches. In *Proceedings. of the 9th Computer Vision Winter Workshop*, pages 159–166, Piran, Slovenia, 2004.

[36] S. Gilles. *Robust Description and Matching of Images.* PhD thesis, University of Oxford, 1998.

[37] W.B. Goh and K.Y. Chan. A Shape Descriptor for Shapes with Bounday Noise and Texture. In *Proceedings of British Machine Vision Conference*, Norwich, UK, 2003.

[38] G. Golub and F. Van Loan. *Matrix Computation, Third Edition.* The John Hopkins University Press, 2003.

[39] Yihong Gong and Xin Liu. Video summarization and retrieval using singular value decomposition. *Multimedia Systems.*, 9(2):157–168, 2003.

[40] J. Guivant and E. Nebot. Optimization of the Simultaneous Localization and Map Building Algorithm for Real Time Implementation. *IEEE Transactions of Robotics and Automation*, 17(3):242–257, May 2001.

[41] E. J. Gumbel. *Statistics of Extremes.* Columbia University Press, New York, NY., 1958.

[42] J. Gutmann and K. Konolige. Incremental Mapping of Large Cyclic Environment. In *Proceedings of the Conference on Intelligent Robots and Applications*, pages 318–325, Monterey, CA, USA, 1999.

[43] J. Gutmann and K. Konolige. Incremental Mapping of Large Cyclic Environments. In *Proceedings of the Conference on Intelligent Robots and Applications (CIRA)*, pages 318–325, Monterey, California, 1999.

[44] H. Hajjdiab and R. Laganiere. Vision-based Multi-Robot Simultaneous Localization and Mapping. In *Canadian Conference on Computer and Robot Vision*, pages 155–162, Washington, DC, USA, 2004.

[45] J. Hare and P. Lewis. Scale Saliency: Applications in Visual Matching, Tracking and View-Based Object Recognition. In *In Proceedings of Distributed Multimedia Systems / Visual Information Systems*, pages 436–440, Miami, Florida, USA, 2003.

[46] C. G. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proc. 4th Alvey Vision Conference, Manchester*, pages 147–151, 1988.

[47] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.

[48] K. Ho and P. Newman. Combining Visual and Spatial Appearance for Loop Closure Detection. In *Proceedings of European Conference on Mobile Robotics*, Ancona, Italy, September 2005.

[49] K. Ho and P. Newman. Multiple Map Intersection Detection Using Visual Appearance. In *Proceedings of International Conference on Computational Intelligence, Robotics and Autonomous Systems*, Singapore, December 2005.

[50] K. Ho and P. Newman. Loop Closure Detection in SLAM by Combining Visual and Spatial Appearance. *Robotics and Autonomous Systems*, 54(9):740–749, 2006.

[51] K. Ho and P. Newman. Detecting Loop Closure With Scene Sequences. *To be published in Special Joint Issue on Robotics and Vision, International Journal of Robotics Research / International Journal of Computer Vision*, 2007.

[52] B.K.P Horn. Relative orientation revisited. *Journal of the Optical Society of America*, 8:1630–16, October 1991.

[53] W. Huang and K. Beevers. Topological Map Merging. *International Journal of Robotics Research*, 24(8):601–614, 2005.

[54] P. Jensfelt and S. Kristensen. Active global localisation for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17(5):748–760, October 2001.

[55] G.J.F. Jones, J.T. Foote, K. Spärck Jones, and S.J. Young. Retrieving a Spoken Document by Combining Multiple Index Sources. *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, 1996.

[56] Karen Sparck Jones. Exhaustivity and Specificity. *Journal of Documentation*, 28(1):11–21, 1972.

[57] T. Kadir, D. Boukerroui, and M. Brady. An analysis of the scale saliency algorithm. Technical Report OUEL No: 2264/03, Dept. of Engineering Science, University of Oxford, 2003.

[58] T. Kadir and M. Brady. Saliency, Scale and Image Description. *International Journal Computer Vision*, 45(2):83–105, 2001.

[59] T. Kadir, A. Zisserman, and M. Brady. An Affine Invariant Salient Region Detector. In *Proceedings of the 8th European Conference on Computer Vision*, Prague, Czech Republic, May 2004.

[60] M. Kaess and F. Dellaert. A Markov Chain Monte Carlo Approach to Closing the Loop in SLAM. In *Proceedings of International Conference on Robotics and Automation*, pages 643–648, Barcelona, Spain, 2005.

[61] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptor. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 506–513, Washington, DC, USA, 2004.

[62] K. Konolige. Large-Scale Map-Making. In *Proceedings of the National Conference on AI (AAAI)*, pages 457–463, San Jose, CA, 2004.

[63] K. Konolige and K. Chou. Markov Localization using Correlation. In *Proceedings of the International Joint Conferences on Aritificial Intelligence*, pages 1154–1159, Stockholm, Sweden, 1999.

[64] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart. Map Merging for Distributed Robot Navigation. In *Proceedings of International Conference on Intelligent Robots and Systems*, Las Vegas, USA.

[65] J. Kosecka, F. Li, and X Yang. Global Localization and Relative Positioning based on Scale-Invariant Keypoints. *Robotics and Autonomous Systems*, 52(1):27–38, 2005.

[66] J. Kosecka and X. Yang. Global localization and relative pose estimation based on scale-invariant features. In *Proceedings of the International Conference on Pattern Recognition*, pages 319–322, Cambridge, UK, 2004.

[67] B. Kröse and R. Bunschoten. Probabilistic localization by appearance models and active vision. *Proceedings of International Conference on Robotics and Automation*, 1999.

[68] B. Kuipers. Modeling Spatial Knowledge. *Cognitive Science*, 2(2):129–153, 1978.

[69] B. Kuipers and P. Beeson. Bootstrap Learning for Place Recognition. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 174–180, Edmonton, Alberta, Canada, 2002.

[70] B. Kuipers and Y Byun. A Robot Exploration and Mapping Strategy based on a Semantic Hierarchy of Spatial Representations. *Robotics and Autonomous Systems*, 8:47–63, 1991.

[71] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli. Local Metrical and Global Topological Map in the Hybrid Spatial Semantic Hierarchy. In *Proceedings of International Conference on Robotics and Automation*, pages 4845–4851, New Orleans, Louisiana, USA, 2004.

[72] P. Lamon, A. Tapus, E. Glauser, N. Tomatis, and R. Siegwart. Environment Modeling with Fingerprint Sequences for Topological Global Localization. In *Proceedings of International Conference on Robotics and Automation*, pages 3781–3786, Seoul, Korea, 2001.

[73] J. Leonard and H. Durrant-Whyte. Mobile Robot Localization by Tracking Geometric Beacons. *IEEE Transactions. Robotics and Automation*, 7(3):376–382, 1991.

[74] J. Leonard and P. Newman. Consistent Convergent Constant Time SLAM. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, Acapulco, Mexico, 2003.

[75] J.J. Leonard, P.M. Newman, and R.J. Rikoski. Towards Robust Data Association and Feature Modeling for Concurrent Mapping and Localization. In *Proceedings of the Tenth International Symposium on Robotics Research*, Lorne, Victoria, Australia, 2001.

[76] A. Levin and R. Szeliski. Visual Odometry and Map Correlation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 611–618, Washington, DC, USA, 2004.

[77] T. Lindeberg and J. Garding. Shape-adapted Smoothing in Estimation of 3-D Shape Cues from Affine Deformation of Local 2-D Brightness Structure. *Image and Vision Computing*, 15(6):415–434.

[78] D.J. Lipman, W.J. Wilbur, T.F. Smith, and M.S. Waterman. On the statistical significance of nucleic acid similarities. *Nucleic Acids Research*, 12(1):215–226, 1984.

[79] Y. Liu and S. Thrun. Results for outdoor-SLAM using sparse extended information filters. In *Proceedings of International Conference on Robotics and Automation*, Washington, DC, USA, 2002.

[80] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of IEEE Conference on Computer Vision*, pages 1150–1157, Kerkyra, Greece, 1999.

[81] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[82] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.

[83] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275, 1997.

[84] K. Lynch. *The Image of the City*. MIT Press, Cambridge, MA, 1996.

[85] A. Martinelli, N. Tomatis, and R. Siegwart. Some Results on SLAM and the Closing the Loop Problem. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 2917–2922, Edmonton, Alberta, Canada, 2005.

[86] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *Proceedings of British Machine Vision Conference*, pages 384–393, Cardiff, UK, 2002.

[87] P. F. McLauchlan. A batch/recursive algorithm for 3D scene reconstruction. *Proceedings of International Conference on Computer Vision and Pattern Recognition, Hilton Head, SC, USA*, 2:738–743, 2000.

[88] C. Mikolajczyk and C. Schmid. Indexing based on Scale Invariant Interest Points. In *Proceedings of the International Conference on Computer Vision*, pages 525–531, Vancouver, Canada, 2001.

[89] C. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[90] C. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[91] M. Montemerlo and S. Thrun. Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM. In *Proceedings of International Conference of Robotics and Automation*, pages 1985–1991, Taipei, Taiwan, 2003.

[92] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Alberta, Canada, 2002.

[93] K Murphy. Bayesian map learning in dynamic environments. In *Proceedings of Neural Information Processing Systems*, Vancouver, Canada, 1999.

[94] J. Neira and J. D. Tardos. Data Association in Stochastic Mapping using the Joint Compatibility Test. *IEEE Trans. Robotics and Automation*, 17(6):890–897, 2001.

[95] J. Neira, J. D. Tardos, and J. A. Castellanos. Linear time vehicle relocation in SLAM. In *Proceedings of International Conference on Robotics and Automation*, pages 427–433, Taipei, Taiwan, 2003.

[96] Ulric Neisser. Visual Search. *Scientific American*, 210(6):94–102, 1964.

[97] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using Visual Appearance and Laser Ranging. In *Proceedings of International Conference on Robotics and Automation*, pages 1180–1187, Orlando, USA, May 2006.

[98] P. Newman and K. Ho. SLAM - Loop Closing with Visually Salient Features. In *Proceedings of International Conference on Robotics and Automation*, pages 635–642, Barcelona, Spain, 18-22 April 2005.

[99] J. Nieto, J. Guivant, E. Nebot, and S. Thrun. Real Time Data Association for FastSLAM. In *Proceedings of International Conference on Robotics and Automation*, pages 412–418, Taipei, Taiwan, 2003.

[100] D. Nister. An Efficient Solution to the Five-point Relative Pose Problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(6):756–770, June 2004.

[101] C. Papadimitiou and K. Stieglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, 1982.

[102] E. Royer, Lhuiller M., M. Dhome, and T. Chateau. Towards an alternative GPS sensor in dense urban environment from visual memory. In *Proceedings of British Machine Vision Conference*, London, UK, 2004.

[103] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[104] F. Savelli and B. Kuipers. Loop-closing and Planarity in Topological Map-Building. In *Proceedings of International Conference on Intelligent Robots and Systems*, pages 1511–1517, Sendai, Japan, 2004.

[105] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume 1, pages 414–431, 2002.

[106] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[107] S. Se, D. Lowe, and J. Little. Vision-based Mapping with Backward Correction. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 153–158, Lausanne, Switzerland, 2002.

[108] S. Se, D.G. Lowe, and J. Little. Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks. *International Journal of Robotics Research*, 21(8):735–758, 2002.

[109] S. Se, D.G. Lowe, and J. Little. Vision Based Global Localisation and Mapping for Mobile Robots. *IEEE Transactions on Robotics*, 21(3):364–375, June 2005.

[110] C. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[111] R. Sibson. SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[112] C. Silpa-Anan and R. Hartley. Visual Localization and Loop-back Detection with a High Resolution Omnidirectional Camera. In *Workshop on Omnidirectional Vision*, Beijing, China, 2005.

[113] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *Proceedings of the International Conference on Computer Vision*, pages 370–377, Beijing, China, 2005.

[114] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, pages 1470–1477, Nice, France, October 2003.

[115] R. Smith, M. Self, and P. Cheeseman. A Stochastic Map for Uncertain Spatial Relationships. In *4th International Symposium on Robotics Research*, pages 467–474, Santa Clara, California, United States, 1987.

[116] T.F. Smith and M.S. Waterman. Identification of Common Molecular Subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[117] T.F. Smith, M.S. Waterman, and C. Burks. The Statistical Distribution of Nucleic Acid Similarities. *Nucleic Acids Research*, 13(2):645–655, 1985.

[118] C. Stachniss, G. Grisetti, and W. Burgard. Recovering Particle Diversity in a Rao-Blackwellized Particle Filter for SLAM After Actively Closing Loops. In *Proceedings of International Conference on Robotics and Automation*, pages 655–660, Barcelona, Spain, 2005.

[119] C. Stachniss, D. Hahnel, and W. Burgard. Exploration with Active Loop-Closing in SLAM. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1505–1510, Sendai, Japan, 2004.

[120] G. Strang. *Linear Algebra and its Applications, 3rd Edition.* Brooks/Cole, Thomson Learning, 1980.

[121] H. Surmann, K. Lingemann, A. Nchter, and J. Hertzberg. Fast acquiring and analysis of three dimensional laser range data. *Proceedings of the 6th International Fall Workshop Vision, Modelling, and Visualization*, pages 59–66, November 2001.

[122] A. Tapus, S. Heinzer, and R. Siegwart. Bayesian Programming for Topological Global Localization with Fingerprints. In *Proceedings of International Conference on Robotics and Automation*, pages 598–603, New Orleans, Louisana, USA, 2004.

[123] S. Thrun. A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.

[124] S. Thrun, W. Burgard, and D. Fox. A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping. In *Proceedings of International Conference on Robotics and Automation*, San Francisco, California, USA, 2000.

[125] S. Thrun, D. Fox, and W. Burgard. A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. *Machine Learning*, 31(1-3):29–53, 1998.

[126] S. Thrun and Y. Liu. Multi-robot SLAM with sparse extended information filers. In *Proceedings of the 11th International Symposium of Robotics Research*, Sienna, Italy, 2003.

[127] N. Tomatis, I. Nourbaksh, and R. Siegwart. Hybrid Simultaneous Localization and Map Building: Closing the Loop with Multi-Hypotheses Tracking. In *Proceedings of International Conference on Robotics and Automation*, pages 2749–2754, Washington, DC, USA, 2002.

[128] P H S. Torr and D W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997.

[129] P. H. S. Torr and A. Zisserman. Robust computation and parameterization of multiple view relations. In *Proceedings of the 6th International Conference on Computer Vision, Bombay*, pages 727–732, 1998.

[130] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based Vision System for Place and Object Recognition. In *Proceedings of the International Conference on Computer Vision*, pages 273–280, Nice, France, 2003.

[131] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle Adjustment - A Modern Synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–372, London, UK, 2000. Springer-Verlag.

[132] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of International Conference on Robotics and Automation*, pages 1023–1029, San Francisco, California, USA, 2000.

[133] J. Wang, R. Cipolla, and Zha H. Vision-based Global Localization using a Visual Vocabulary. In *Proceedings of International Conference on Robotics and Automation*, pages 4230–4235, Barcelona, Spain, 2005.

[134] A. Webb. *Statistical Pattern Recognition*. John Wiley and Sons, Ltd, 2002.

[135] J. Weber, L. Franken, K. Jörg, and E. Puttkamer. Reference Scan Matching for Global Self-Localization. *Robotics and Autonomous System*, 40(2):99–110, August 2002.

[136] J. Weber, K. Jörg, and E. Puttkamer. APR-Global Scan Matching Using Anchor Point Relationships. *The 6th International Conference on Intelligent Autonomous Systems (IAS-6), Venice, Italy*, pages 471–478, July 2000.

[137] G. Weiβ, C. Wetzler, and E. Puttkamer. Keeping Track of Position and Orientation of Moving Indoor Systems by Correlation of Range-Finder Scans. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 595–601, Munich, Germany, 1994.

[138] I. H. Whitten, A. Moffat, and T. C. Bell. *Managing Gigabytes: compressing and indexing documents and images*. International Thomson Publishing, 1994.

[139] J. Wolf, W. Burgard, and H. Burkhardt. Robust Vision-Based Localization by Combining an Image-Retrieval System with Monte Carlo Localization. *IEEE Transactions on Robotics*, 21(2):208–216, 2005.

[140] B. Yamauchi and P. Langley. Place recognition in dynamic environments. *Journal of Robotic Systems*, 14:107–120, 1997.