Appearance-based hierarchical localization and environment modeling for mobile robot

Chen, Cheng

2007

Chen, C. (2007). Appearance-based hierarchical localization and environment modeling for mobile robot. Doctoral thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/39150

https://doi.org/10.32657/10356/39150

Appearance-based Hierarchical Localization and Environment Modeling for Mobile Robot

Chen Cheng

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2007

Acknowledgements

First and foremost, I am most grateful to my supervisor, Associate Professor Wang Han, for his guidance, comments and supports in the process of my study.

I would like to thank all the SLAM researchers in NTU: Dr. Martin Adams, Dr. W.S.Wijesoma, Lee Kuang Wee, Liu Bingbing, Tang Fan and others for their contributions to NTU SLAM reading group.

Also, I wish to express my gratitude to all the people who have assisted me during the years of my PhD study at Nanyang Technological University: Wang Xiao, Zhu Xiaoling, Liu Nan, Kong Hui, Chen Xujian, Kurt Zheen Zhao, Li Jiang, Leng Jing, Liu Xiaohui for sharing with me their ideas and knowledge.

I am also graceful to Dr.Javier Ibanez Guzmán and all the Ulysses team members: Ng Teck Chew, Shen Jian, Jiang TingYing, Xu Jian, Gong Zhiming for their spirit of cooperation.

Finally, I wish to thank my parents who have given me the strongest supports all the time, without which the work described herein could never be done.

Contents

Summary

Unmanned vehicle navigation is a challenging problem that demands extensive investigations. The localization and environment modeling is one of the most fundamental components in an unmanned vehicle system. Currently, compared with the progresses in vehicle control and vehicle electronics, the insufficient localization and mapping capabilities have become a bottle-neck for unmanned vehicles.

This thesis is dedicated to the sensor data reasoning and environment modeling for unmanned ground vehicles (UGV). More specifically, the objective of this thesis is to derive a probabilistic framework to localize the vehicle and model its surrounding environment. By exploiting appearance information, the proposed techniques are invariant to sensors and target environments. Meanwhile, the presented algorithms can achieve considerable efficiency and accuracy, even for large-scale localization and mapping tasks.

The contribution of this thesis can be summarized as follows.

First, the target environment is modeled as a hierarchy. In this hierarchy, the metric information is encapsulated by a local representation called a 'submap'. The submaps are modeled at the topological level. A known problem is that, this symbolic topological model is not compatible with any numerical inference pro-

cess. The solution in this work is to construct a *map space* that is manifolded in the raw data space. This map space is highly compact, but contains rich information about the appearance of the local environments. By modeling submaps as Gaussian distributions in this manifold, a probabilistic Bayesian inference can be conveniently performed at the topological level. In this thesis, such topological inference is employed to solve the loop-closure detection problem, in which the vehicle is expected to identify the place where it has already been to.

To obtain a consistent map, after a loop-closure is correctly detected, at the point of loop-closure, the localization and mapping algorithm should propagate the estimation error backward to the whole submap topology, so as to correct its previously accumulated errors. Such a bundle adjustment is often referred to as 'loop-closing optimization' in the robotics community.

The second contribution of this work is to employ a 'global appearance prior' to facilitate this adjustment process. The adjustment is modeled as an optimization process. The optimal submap deployment is obtained by solving a *maximum a posteriori* (MAP) estimation problem. The *a priori* information is obtained from both robot mapping and the environment's appearance. Such appearance information can be easily obtained from roadmaps, or if possible, satellite images and aerial photos.

In this work, a two-stage, coarse-to-fine process is introduced to optimize the *a posteriori* map. The first stage is a powerful yet expensive global searching procedure based on a genetic algorithm. It converges when a rough estimate for the submap configuration is achieved. To ensure the efficiency, this estimate is not

necessarily accurate. Subsequently, the second stage is a gradient search algorithm, which is inspired by the active contour algorithm in computer vision domain. The submap configuration from the genetic algorithm is further tuned by an iterative greedy optimization process. By employing this two-stage optimization process, the proposed algorithm offers both efficiency and robustness.

List of Abbreviations and Symbols

Abbreviations

EM	Expectation maximization
GA	Genetic algorithm
HT	Hough transform
IMM	Iterative multiple module
IMU	Inertial Measurement Units
JPDF	Joint probabilistic data association filter
KL	Kalman filter
LAP	linear assignment problem
LC	loop-closing
LCAP	loop-closing with appearance prior
LDA	Linear discriminative analysis
LLE	Locally linear embeding
MAP	Maximum *a posteriori*
MC	Monte Carlo
MCMC	Monte Carlo Markov chain

MHT	Multiple hypotheses tracking
ML	Maximum likelihood
PCA	Principal component analysis
PDAF	Probabilistic data association filter
PDF	Probability density function
PF	Particle filter
SA	Simulated annealing
SIFT	Scale-invariant feature transform
SIR	Sequential importance re-sampling
SLAM	Simultaneous localization and mapping
SVM	Support vector machine
UGV	Unmanned ground vehicles
UKF	Unscented Kalman filter

Symbols

$\mathcal{L}$	the posterior configuration of submap topology using absolute position
$\mathrm{L}(s)$	the location of submap s in the posterior configuration
$\mathcal{M}$	the set of submap representations
M_s	the individual representation for submap s
N_s	the total number of submaps
$\mathcal{P}$	the set of uncertainties in submap transformation prior
P_s	the uncertainty in submap transformation prior $\bar{\mathrm{x}}_s$

Sj_t	the likelihood that vehicle is in submap j at time t
$\mathcal{S}$	the set of submaps
S_t	the probability distribution of the vehicle's position over the submap space
T^*	the ML estimation for the vehicle's position in the topological nodes' space
T_t^m	the topological node m at time t
$\mathcal{T}$	the set of topological nodes
U_t	the set of measured vehicle motions till time t.
$\bar{W}_i$	the mean of submap i's distribution in map space
$\bar{\mathrm{x}}_s$	the prior transformation between submap $s-1$ to submap s
$\mathcal{X}$	the posterior configuration for submap topology using relative transformations
$\bar{\mathcal{X}}$	the prior estimation of topological submap graph from mapping process
Z_t	the set of observations till time t
Φ	the local level mapping process
σ_i	the variance of submap i's distribution in map space
Ω	the projection from measurement space to map space

List of Figures

Chapter 1

Introduction

1.1 Motivation

With fast developing control and artificial intelligence technologies, the research on unmanned vehicles has made tremendous progress. While unmanned aerial vehicles (UAV) have already been commercialized and applied to modern warfare, unmanned ground vehicles (UGV) and unmanned underwater vehicles are still facing several unsolved problems.

One of the major challenges of UGV is the hostile environment. Different from aerial space, which is open and free, the terrain is often occupied by different static or dynamic obstacles. To avoid these unaccessible regions, a highly intelligent perceptional module must be developed. This module should be capable of processing input sensor observations and estimating the interactions between the vehicle and the environment.

The problem of localization and mapping has received considerable attentions in the mobile robot community. The goal of a mobile robot performing localization and mapping is to start from a location in an unknown environment, build a map

of its surroundings incrementally by using the uncertain information received from sensors, whilst simultaneously using the acquired map to localize itself respect to this map. This sensor fusion mechanism is illustrated in Figure 1.1.

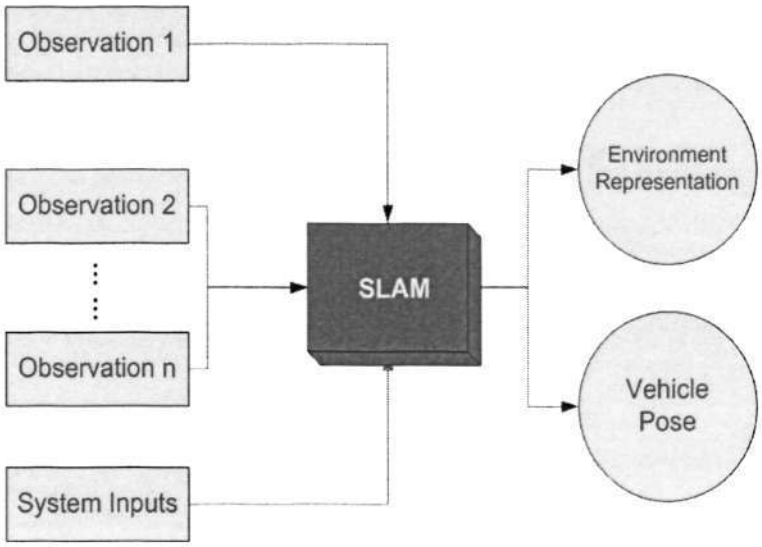


Figure 1.1: Localization and mapping is basically a sensor fusion problem.

It is widely expected that localization and mapping will lead to a self-contained, bootstrap vehicle perception module. This module may estimate both the states of the vehicle and the environment, and therefore support the higher level navigation tasks such as path planning. After years of investigation, tremendous successes have been achieved in various applications, however, the author believes that the mainstream metric localization and mapping can still be improved in three aspects: higher level reasoning, scalability, and environment invariance.

1.1.1 Higher level reasoning

The original strategy of using metric SLAM for autonomous navigation can be briefly explained as follows. When the vehicle obtains a map of its surrounding

environment, together with its location, it can project the map into its own coordinate system. Then the robot can generate a path to avoid potential obstacles. However, in real world applications, it is often desirable that the control commands could be generated from local observations, rather than *a priori* maps. The reason lies in the fact that, no matter how accurate the *a priori* global map could be, it will never be exactly the same as the environment where the vehicle actually moves. There always exist moving obstacles or other kinds of disturbances, which cannot be modeled in advance. Consequently, the mainstream *reactive* path planning strategy has made the robot less dependent on the metric-level positioning.

While the metric localization and mapping have been widely discussed, their topological counterparts have not yet received adequate attentions. In this work, the author argues that the higher level localization and mapping is of great importance with respect to both theory and practice. Although the *reactive* path planners based on local observations work satisfactorily in both indoor and outdoor environments, the control commands sent to such path planners should be nevertheless based on certain higher-level guidance. For example, a *reactive* planning algorithm may lead the vehicle from one room to another. However, some intelligence is still necessary to decide whether the vehicle should travel from Room A to Room B, or from Room A to Room C. Such guidance can only be generated when the robot has a higher-level self-conscious.

1.1.2 Scalable to large mapping tasks

Another desirable application of simultaneous localization and mapping is unmanned reconnaissance and surveillance, where accurate and comprehensive mapping results are expected. Such large scale high resolution mapping tasks generally require the SLAM algorithms to scale well to the size of the map. Although during the past few years, the metric SLAMs based on graphical model have achieved tremendous success, they still suffer considerably from the unified map model that explicitly describes all the map items in one single coordinate system. If the map is organized in a hierarchy, the localization and mapping algorithm may only need to process information on the higher level representations, rather than handling a large quantity of low-level detailed information. By doing so, the efficiency of the algorithm could be significantly improved.

1.1.3 Environment invariance

The performance of metric SLAM algorithms relies critically on the correct associations between two successive frames of data. The robustness of such associations is largely determined by the quality of raw measurements. For example, in a structured environment, there exist rich geometrical patterns, such as lines, corners and compact clusters of points. Therefore, the useful information is dominant. In structured environments, both feature-based and matching-based techniques can work properly. However, in unstructured environments, the reflected laser beams that construct a feature have trivial percentage in the whole scan. The algorithms have to find these trivial landmarks within a huge amount of input data,

which is a challenging task even for humans. Consequently, in these environments, landmark-based algorithms do not work and even matching-based algorithm are prone to fail.

It should be noted that, the robustness problem of metric localization and mapping essentially comes from the complexity of environments, or equivalently, the insufficient perceptual power of 2D laser scanners. In this thesis, the author argues that the information contained within a single laser scan is not adequate for SLAM analysis in a highly unstructured environment, such as cross-country jungles. A possible improvement here is to organize the measurements into small groups, so that a group of measurement frames can be analyzed together and the rich information contained in them can be exploited.

The above limitations of metric SLAM and the advantages of employing a higher level map modeling scheme have motivated the author to conduct this research, in effort to develop an innovative and practical localization and mapping framework.

1.2 Objective

Given above limitations of metric SLAM, it is argued in this thesis that a multi-level data modeling and the corresponding data processing strategy must be developed. Accordingly, the objective of this research is to build an innovative localization and mapping framework based on a hierarchical map representation. This framework must be general, efficient, and extensible.

- **General**

  The localization and mapping algorithm should be invariant under different sensor techniques. For different types of input data, the algorithm should work properly without major modifications. In other words, the algorithm should not exploit heuristics that is only applicable to specific sensors. Equivalently, the algorithm must be environment invariant: it should not be developed for only one specific kind of environment.

- **Efficient**

  Timing is always an important issue in unmanned vehicle applications. The time requirement for a localization and mapping algorithm should be acceptable and controllable for different applications.

- **Extensible**

  The localization and mapping is essentially a sensor fusion problem. While most current techniques are based on data from laser range scanners and dead-reckoning, the algorithm is also expected to be capable of incorporating other sources of information.

1.3 Contributions

1.3.1 A generalized localization and mapping framework

The conventional feature extraction algorithms, together with the scan matching algorithms, can be generally regarded as metric level data representation methods. In addition to these metric level methods, there also exist various statistical

techniques that directly build representations in the raw data space. For example, principal component analysis (PCA), locally linear embedding (LLE), and ISOMAP [91]. Since they are not based on any predefined geometric patterns, these techniques are often referred to as appearance-based model.

The conventional data association techniques, such as multiple hypothesis tracker (MHT), joint capability test, and scan matching, can be generally regarded as recognition processes. These algorithms try to 'recognize' the new observed items from the previously observed measurements. However, correspondences can also be found without decomposing the observations into a set of low dimensional structures (raw scan points or landmarks). In the pattern recognition community, various techniques have been established to perform frame-to-frame recognition, e.g., support vector machine [85], hidden markov model, AdaBoost [97].

Filters are the most popular methods to fuse landmark measurements collected at different time instances. Basically, the objective of a filter (e.g., Kalman filter) is to find an appropriate trade-off between the new observed information and the previously observed information. This can be generally regarded as an optimization process, in which the algorithm tries to obtain an optimal estimate based on the available data. For instance, the Kalman filter is optimal in the sense that it yields minimum mean-square error (MMSE) estimates. In addition to filters, there are various techniques that can also be used for optimization, e.g., graph cut [51], genetic algorithm (GA), and those gradient techniques.

Based on above generalizations, the author argues that the localization and mapping is essentially a combination of three sub-problems, i.e., representation,

finding correspondences, optimization. In this thesis, a general, efficient and extensible mobile robot localization and mapping framework is proposed. This framework is illustrated in Figure 1.2.

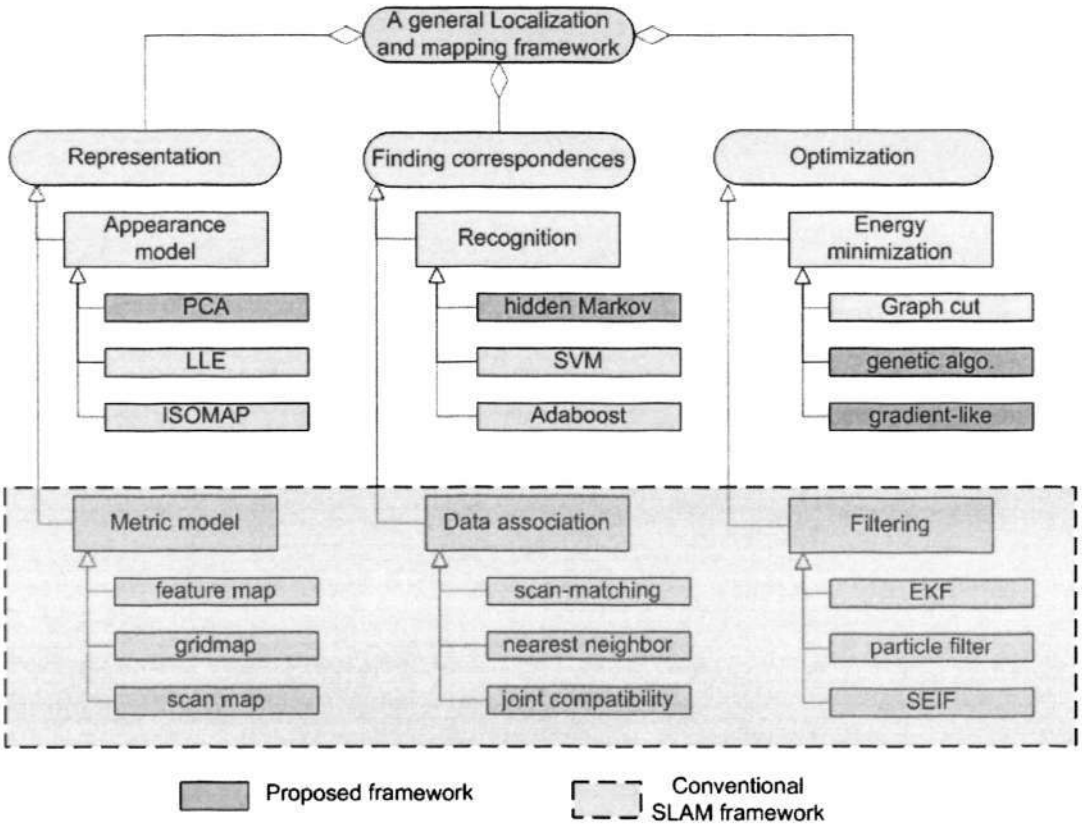


Figure 1.2: The general localization and mapping framework. In this figure, arrows stand for generalization and diamonds represent combinations.

The major innovation in this thesis is to model mobile robot localization and mapping as a general sensor fusion problem based on a hierarchical map model, and to employ appearance information in both localization and mapping processes. Conventional SLAM algorithms try to answer the question 'what is the environment', while the concern of the proposed framework is 'how does the environment look?'.

1.3.2 List of contributions

This thesis presents innovative techniques that cover all the three major components of localization and mapping. The detailed contributions are listed as follows.

- **Hierarchical map representation**

  To compensate the deficiencies of metric localization and mapping, higher level abstraction for the map is developed. The map is organized as a hierarchy. At the first level, it is represented as a collection of local regions called submaps. These submaps encode the appearance characteristics of the local environment. Therefore, the appearances within a submap are homogeneous, while the appearances in different submaps are distinct. Lower than the submap level, the topological node level is constructed by dividing submaps according to a fixed resolution. The purpose of this level is to encode the vehicle's motion at the topological level. The lowest level in this hierarchy is the local metric map, which has been widely investigated. In this map hierarchy, the metric level is employed to store the detailed environment information. By introducing this multi-layer hierarchical map model, the algorithm can handle appearance information, motion information, and metric level detailed information respectively. It can therefore significantly improve the efficiency and flexibility of localization and mapping.

- **Appearance-based data modeling**

  The feature extraction algorithm is designed to model only a pre-defined portion of the input data, however, the goal of the proposed data modeling

algorithm is to capture and model the *dominant* information about an environment. For example, it tries to capture what makes a corridor different from a room, without necessarily detecting doors or walls. Compared with the conventional feature extraction or scan matching algorithms that try to model, say, 10% of the information contained in the unstructured data, the proposed technique can robustly capture more than 90% information, and then model them efficiently.

- **Bayesian inference for map hierarchy**

  Based on the above appearance model, the probabilistic Bayesian inference is integrated with a symbolic topological map to re-localize a mobile robot. The appearances of different submaps are approximated to be Gaussian. Such Gaussian models serve as the 'glue' between the topological map structure and the Bayesian inference. By employing such probabilistic 'glue', the Bayesian inference at the metric level can be conveniently implemented on topological level. One prominent advantage offered by this algorithm is that, it can be applied to a cross-country environment where no features or landmarks are available. Further more, the re-localization can be performed without information from the central SLAM estimation loop.

- **Map optimization with appearance information**

  In this thesis, a hierarchical mapping method is also developed. This algorithm can accurately map a large environment with one robot by visiting the environment for only once. The basic idea is to represent the map by

a collection of submaps arranged in a deformable configuration. A general framework is proposed to compute the *a posteriori* distribution over the space of submap configurations. Such posterior is obtained based on robot mapping and an appearance prior, e.g., a roadmap. By introducing appearance prior as a new source of information, the Maximize *a posteriori* (MAP) estimate can be formulated as an energy minimization problem. The above general energy minimization model essentially encapsulates the localization and mapping into a fixed close-form target energy function. Thereafter, various optimization tools can be applied to minimize the energy according to the specific requirements from different mapping tasks. Both global and local optimization tools are employed in this work. For greedy optimization, a gradient technique is developed. It gradually leads the energy function to a (local) minimum. This algorithm is demonstrated to be highly efficient, though it may be trapped by local minimums. So a global optimization based on genetic algorithms (GA) is applied first to achieve a rough estimate of the submap configuration. Subsequently, this configuration is used as the initial state for the gradient local search. With this two-stage optimization strategy, the proposed algorithm offers both efficiency and robustness.

1.4 Organization of the thesis

This thesis is organized as follows.

Chapter 2 gives a brief introduction to the problem of localization and environment modeling. The localization and mapping is first analyzed as a general estima-

tion problem. Those unique characteristics belonging to SLAM its own are further discussed. Thereafter, the localization and mapping problem is decomposed into three sub-problems: data representation, data association, and data filtering (optimization). A comprehensive literature review is conducted by classifying SLAM literatures into these three classes according to their major contributions.

In Chapter 3, an appearance based mobile robot localization approach is applied to the problem of loop-closure detection. The first section in this chapter explains how to divide the whole map into a hierarchy. After that, the second section formulates how those high dimensional raw measurements are projected into a low dimensional space (map space), which describes the appearance of the environment. Since laser scans from the same region share similar appearance, they are expected to form a distinct cluster in the low dimensional space. This cluster essentially encodes the appearance information of this region. In this work, it is approximated by a Gaussian distribution. In the next section, such Gaussian distributions are employed as the observation model by a Bayesian network. Accordingly, how the robot can localize itself using the Bayesian inference is also presented.

Chapter 4 presents an efficient appearance based loop-closing optimization technique, which can perform large scale loop-closing with one single revisiting. In the first section, the generic probabilistic loop-closing is extended by integrating appearance information as a new prior. The loop-closing problem is then formulated as a *Maximum a posteriori* (MAP) estimation problem. Thereafter, it is demonstrated that this MAP estimation is essentially an energy minimization problem,

which is similar to the active contour problem in computer vision. In the following sections, this energy minimization problem is solved in a coarse-to-fine manner. A two-stage optimization process is developed. The first stage is a global searching process based on Genetic Algorithm (GA), whose objective is to find a rough estimate for the submap configuration. Thereafter, a gradient-based local optimization algorithm is developed to refine the result of global searching.

Chapter 5 provides a summary of the main issues addressed in this thesis and comments on future work.

Chapter 2

Literature review

2.1 Introduction to localization and mapping

The problem of localization and mapping is an active research area [11] [62] [94] in the robotics community. With advanced sensory and computing technologies, today's localization and mapping algorithms have achieved impressive accuracy, robustness and efficiency.

The ultimate goal of localization and mapping is to estimate the state of a vehicle and its environment. This is essentially a stochastic estimation problem [93]. Every estimation algorithm has at least two major components: the observation model and the process model. In a real world application, both of these two models are imperfect and suffer from modeling errors. An estimation algorithm's objective is to manage these errors and to ensure that the output is optimal based on a certain pre-defined criterion.

For the observation model, every sensor measurement comes with error. The amount of error is determined by the characteristics of the sensor involved. Errors are also introduced when raw measurements are further fitted by certain represen-

tation models.

Errors also exist in the process model that describes how a localization and mapping system evolves over time. In the real world, the localization and mapping is essentially a high dimensional and highly non-linear stochastic process that is difficult to formulate. Consequently, in practice, this process can only be approximated with compromises of accuracy, robustness, or generality. The most common error in process modeling is the linearization error, which is introduced when the system's nonlinear state model is linearized for a recursive filter.

If the robot is deployed in an unknown environment, it is desirable that the robot can localize itself when it explores this environment. This is essentially a bootstrap learning process in which the robot 'learns' its own pose and surrounding environment by accumulating information when it travels. Therefore, in a SLAM algorithm, the obtained measurements should not be discarded. These measurements must be formatted and stored for further analysis. This is a unique characteristic that distinguishes SLAM from other localization and mapping problems.

The mechanism of SLAM introduces two further challenges. The first challenge is to handle the redundancy of information. The sensors are working at high speed and the information is collected *incrementally*. Consequently, the observations at different time and places may overlap. As a result, the measurements corresponding to the same entity in the real world could be received more than once. This phenomenon leads to two additional problems. The first is the association problem, which is the problem of finding the correct correspondences between

measurements collected at different times. The second one is the fusion problem: if a certain entity is observed at different times or places, and we could find correct associations between them, then how can these observations be fused together into one single estimate?

The second challenge is about the efficiency of SLAM. If all the previously acquired information must be maintained, the memory consumption of a SLAM algorithm should grow linearly or even faster. Meanwhile, the complexity of fusing new observations may also grow rapidly. If this problem is not solved, a SLAM algorithm would become prohibitively slow after even a short while.

A SLAM algorithm must include techniques to solve above problems. A workable SLAM algorithm must have at least three basic components:

- **Representation**

  The responsibility of this module is to format the input data for further analysis. Data modeling can be regarded as the bridge between sensor observation and map representation. It plays the most important role in a SLAM algorithm. On one hand, data modeling completely determines how the output map is represented. On the other hand, it also imposes strong constraints on how the sensor inputs are further processed.

- **Finding correspondences**

  As discussed above, SLAM algorithms must find correspondences among sensor observations. These observations could be successive (as in a filtering process), or far away from each other. In general, these two kinds of data

association problems propose different challenges and should be handled individually (although a unified solution is also available).

- **Optimization**

  If the association between observations can be obtained, SLAM algorithms are expected to fuse the measurements corresponding to the same object, in effort to achieve an optimal estimate of this object. This can be generally regarded as an optimization process that optimizes a predefined energy function.

These three major components of SLAM are actually highly correlated. In the following sections, a comprehensive review for the related SLAM papers is conducted. According to their major contributions, these papers are classified into above three catalogs. Additionally, some extensions of SLAM in the mobile robot domain are also reviewed.

2.2 Representation

Techniques for data modeling essentially share the same goal: mapping raw measurements into another space. In this new space, it will be easier to process these observations. Usually, this space has a dimensionality much lower than the original measurement space. This is not difficult to understand if we regard such projection as a feature extraction process. However, this may not be always true, for example, raw-scan-based approaches do not modify the raw measurements before further processing.

2.2.1 Metric map models

Given a sequence of sensor measurements, a metric map based algorithm builds a representation for the geometrical structure of the mapped environment. The advantage of constructing this structural model is that, given the sensor's motion, the algorithm can predict, by itself, what the new observations should be. With such a geometrical model, the states of the vehicle, as well as the map, can be recursively estimated by filtering measurements from motion sensors and perceptual sensors. Such a map modeling scheme essentially underlies all current metric SLAM algorithms [20, 36].

2.2.1.1 Feature-based modeling

Using a feature representation, the environment is modeled as a combination of geometrical patterns such as points [61], circles [36], corners [1], openings [100], lines [45, 42], and more recently, polylines [104]. For visual sensors, Zhou [113] developed the multi-dimensional histogram to represent the rich information found within an observed image, such as colors, edges and textures. Lamon [59] introduced a low dimensional representation, called image fingerprint sequences, for measuring the similarity between image frames. Similar comparisons can also be applied to image histogram, as proposed by Ulrich [103].

To extract specific geometric features, *a priori* knowledge of the pattern of features must be available. However, when the environment is unknown, it is often impossible to define the features in advance.

One solutions here is to employ feature definition using less heuristics. The

scale-invariant criterion [63] is widely used to select features. Basically, it assumes that features are prominent enough to be observable at different scales. This can be implemented by conducting convolutions in the input data domain using kennels at different scales. Good features are expected to be prominent in all scales.

Scale-invariant techniques [65] can detect the most plausible features based on one single frame of measurement. However, such features are not necessarily the most plausible in terms of tracking. There could exist prominent features in one frame that vanish after only one or two steps. Consequently, these features may not contribute to the localization process.

The alternative strategy is to locate features from a sequence of measurement frames. Kumar [56] developed the automatic feature detection algorithm based on locally linear embedding [57]. This algorithm uses previously collected image frames to conduct off-line training. When the vehicle is deployed in a similar environment, those regions that are prominent in all observed image frames can be recognized and used as landmarks.

2.2.1.2 Raw-scan-based modeling

The occupancy gridmap (OG) was introduced by Elfes [27]. The basic idea of an occupancy gridmap is to finely divide the map into grids at a fixed resolution. Each grid has a value to represent the likelihood that it is occupied by an object or obstacle. This likelihood can be updated when new sensor data are received. Compared with feature maps, the gridmaps provide more comprehensive 2D descriptions. Furthermore, it can be directly used for navigation: the control

command can be computed solely based on the gridmap. Occupancy gridmaps have gained tremendous popularity during the past few years. Gutmann [37] used correlation to detect possible matches between current observation and the map. Such correlation-based technique is also used by Duckett [25], in whose research the correlation is applied to the histogram of gridmap.

Similar to the gridmap, recently, points or scan matching have also been used in SLAM to serve as the observation model [64, 92]. Since raw range scans are in a high dimensional space, it is impossible to construct one unified state vector for both scans and vehicle poses. Consequently, these two states can only be estimated sequentially, rather than simultaneously.

Nieto [73] recently developed Scan-SLAM that uses a scan-matching technique to solve the data association problem. This algorithm first segments the raw scan frame, thereafter, low-dimensional landmarks can be constructed with Sum of Gaussian (SOG) uncertainties.

As mentioned before, the real-world exists in an extremely high dimensional space (north east coordinates, color, shape, etc). It is very difficult to obtain a compact continuous model for such high dimensionality. Occupancy gridmap solves this problem by assuming that the map components are all independent, so the algorithm only incrementally processes the local environment surrounding the robot. This over-restricted assumption essentially cuts out the correlations among map items. Consequently, when the vehicle comes back to a place it has already been to, the new observations cannot be fused with the old ones.

Another problem is that, the above two techniques are not suitable for long-

distance data association, or so-called loop-closure detection. For loop-closure detection, they often need to work together with a gating strategy. Based on the vehicle's pose prediction, an uncertainty gate is set up first. If landmarks (or other kinds of map items) fall into this gate, the scan-matching routine is triggered to examine potential loop-closure. Apparently, such a scheme could suffer from erroneous vehicle pose estimates from the central SLAM loop. Without the uncertainty gate, the matching algorithm must go through all of the mapped environment to find a possible match, and this 'browsing' process needs to be performed every time new scan arrives. Such a mechanism is prohibitively expensive when the map grows large enough. Additionally, scan-matching itself is not a trivial problem. As to be reviewed in Section 2.3.2, matching two sets of points with distortions and outliers is at least as challenging as feature extraction itself.

Taken the discussion above, it can be observed that, *directly using the raw measurements in SLAM is actually a double-bladed sword.* On one hand, it keeps input measurements intact for the data association and filtering, so that valuable information will not be prejudicedly abandoned by feature extraction. On the other hand, it also keeps all the noise, distracters, or other kinds of disturbances without filtering them away. This poses tremendous challenges to the following data association and filtering process.

2.2.2 Appearance-based map model

Principal component analysis (PCA) is a widely used tool to handle high dimensional measurements spaces. A PCA-based recognition and localization algorithm

was originally introduced in the computer vision community by Turk [102], and then introduced to the mobile robot community by Ishiguro[43]. Also using PCA, Vlassis [107] proposed a robot localization algorithm that used appearance information to localize a mobile robot in the indoor environment. A similar implementation for 2D range data was developed by Crowley [17], in which synthetic range scans were calculated and used to train the appearance model. To the author's knowledge, this work in [17] is the first time that 2D range data are used for appearance-based mobile robot localization. Although innovative, this approach is essentially not completely appearance-based because it still relies on a composite range map to generate synthetic scans, and building such a map can be quite challenging in outdoor environments. A known problem of the original PCA approach is that it could be time-consuming to build the eigenspace as the robot travels. Artac [2] employed an incremental approach to conduct PCA for image data, and demonstrated a significant reduction in complexity. Similar to PCA, linear discriminant analysis (LDA) [26] is also a popular techniques for conducting dimensionality reduction and appearance-based feature extraction. Different from PCA, which deals with the data in its entirety without paying any particular attention to the underlying class structure, LDA deals directly with discrimination between different classes. A more comprehensive comparison between LDA and PCA can be found in [66].

The appearance-based solution that is the most similar to ours is the one proposed by Krose [54], in which a sophisticated algorithm is presented to calculate the probability of observing a certain scene given a robot pose. For this approach,

using a panorama camera, Krose demonstrated attractive localization results in an office environment. However, the performance of this method could be further improved if the topological structure of the environment was incorporated in the localization process.

2.3 Finding correspondences

2.3.1 Tracking based techniques

Since the very beginning of SLAM research, data association techniques based on tracking theories have been employed in SLAM by Simth [87]. The basic idea is to project the stored map items (e.g., landmarks) into the robot's current local coordinate system. A 'gate' surrounding each predicted landmark in local coordinates is further computed. When a new observation falls into this 'gate', a correspondence can be identified. Such a strategy is often referred to as nearest neighbor (NN), which is quite efficient and easy to implement. Unfortunately, it will not be surprising to see that NN is not robust in cluttered environments where distracters or closely placed landmarks can easily confuse the association algorithm. This is because NN is essentially a greedy decision maker that only maintains the best association hypothesis individually for each map item, and for current observation only.

The joint compatibility [72] test was developed to exploit the fact that the landmarks' relative positions to each other are fixed. This is a unique property of SLAM, which does not exist in other data association application such as radar target tracking. Based on this insight, the association for one single landmark

should also be affected by the associations for other landmarks.

Joint Probabilistic Data Association (JPDA) is a similar data association approach that jointly evaluates all the potential association hypotheses. Carine [41] implemented JPDAF using a particle filter (PF), and Hahnel [39] used PF based JPDAF on a mobile robot for tracking multiple people. However, the JPDAF assumes that the number of targets to be tracked is known, and this can become a problem when applied to an unpredictable populated environment. This situation can be handled by including an individual target number estimator in JPDAF, as in Schulz [86].

The limitation of the above techniques is that the association decision is based on the map estimates and vehicle pose estimates. These estimates are computed from the central SLAM loop. Consequently, wrong associations can easily be made when even a small estimation error exists.

This drawback is partly caused by insufficient perceptual information. For conventional 2D range scanners, landmarks are mostly modeled as 2D or 3D vectors that contain no additional information for association. For instance, given a corner in the 2D range scan, it is difficult to tell whether the landmark is a corner of a wall, or a corner of an automobile. Due to such insufficiency, tracking becomes the only criterion available to evaluate possible association candidates. It can thus be observed that, fundamentally, the difficulty with a tracking-based data association strategy comes from the poor sensor input, as discussed in Section 2.2.2.

2.3.2 Matching based techniques

Matching based approaches have become popular in recent years because they do not rely on features and thus can be applied in more unrestricted environment. In other words, they try to bypass the data regression process and pursue directly a high dimensional map model. Therefore, the association problem can be solved in a more informative manner by exploiting the attributes contained in the high dimensional map data.

Hähnel [38] and Wang [109] individually implemented iterative dual correspondence (IDC) [64] for their SLAM algorithms. IDC can be regarded as a modified version of the well-known iterative closest point (ICP) algorithm which was proposed by Besl [6]. It tries to determine the point correspondence and underlying transformation jointly. This leads to a difficult, highly non-convex optimization problem.

A lot of literature has been published in the image processing community on the problem of matching two frames of points. A review of early work can be found in [9]. More recently, Gold [34, 33] and Chui [13] began to solve this joint linear assignment and least squares optimization procedure with a detailed energy function for the matching. The correspondences are not allowed to approach binary values until the transformation begins to converge to a reasonable solution. Similarly, Cross [16] implemented this using expectation/maximization (EM) approach.

Veenman [105, 106] tried to solve the correspondence between dense points in a tracking manner. In this work, several kinds of motion models are generated and

these models are used to build a bipartite graph. The matching is then achieved by finding the minimum cost of the graph using the classic Hungarian algorithm. This work demonstrates a capability to track a sequence of dense moving points, and claims a capability for automatic track initialization and termination. A different probabilistic strategy was proposed by Belongie [4]. A new shape descriptor called shape context was defined for correspondence recovery and shape-based object recognition. Similar to Veenman's work, the correspondence problem is also turned into a bipartite matching problem rather than the more intractable joint linear assignment and least square optimization problem. This problem is solved using the more efficient JVC algorithm [47].

The motion between two frames can also be retrieved without exact point-to-point correspondence. Govindu [35] noticed that certain geometric properties of image contours are invariant under transformation. By comparing these properties (e.g. using cross-correlation), the parameters of the transformation can be retrieved one by one. A similar technique was also used by Roefer [83]. Duckett [24] applied this kind of matching technique to gridmaps, and showed good performance in an indoor environment. However, a compass is required to resolve the rotation. Additionally, the experiments in this work show that although these algorithms can satisfyingly retrieve large motions, they give poor estimats when the motions are small. The smaller the difference between two frames of measurements, the more likely that it is overwhelmed by the noise or other kinds of disturbances.

2.3.3 Non-deterministic techniques

If one frame of measurement is not sufficient to make a reasonable judgment, a straight forward alternative is to accumulate information obtained from a sequence of observations, so that they can be analyzed in a 'batched' manner. Multiple Hypotheses Tracker (MHT) was developed by Reid [81]. It represents the PDF of target positions using multi-Gaussian. Hypotheses are generated according to different possible associations, and then evolve in parallel where only the true one will survive. Cox [15] proposed a track-oriented implementation of MHT, it shows good performance for tracking multiple objects in several image sequences. MHT can also be used to estimate a robot's poses. Jensfelt [45] and Arras [1] implemented MHT-based localization on different robotic systems. Similar localization and re-localization techniques have also been published by Tomatis [99] and Porta [77]. Wang [109] combined the multiple target tracking with SLAM; in this work, the PDF of a robot's pose is regarded to be a uni-model, and the localization and tracking are calculated separately.

MHT is known to be time-consuming because maintaining a huge population of hypotheses uses tremendous computational resources. The Multiple Frame Tracker (MFT) method was developed to make the association algorithm tractable in terms of complexity. It combines probabilistic association with binary programming and has demonstrated impressive efficiency even in large-scale data association applications.

Markov localization proposed by Fox [31] is another implementation of the

above idea to localize a mobile robot. Different from Fox who used only a single beam of laser range data, Gutmann [37, 52] used the correlation between the map and a whole frame of measurement for the observation model.

The dilemma of the original Markov localization method is that, since the robot's *a posteriori* probability distribution is highly non-Gaussian, it is difficult to be continuously modeled. But modeling such a distribution in a discrete manner (at the metric level) is computationally intensive. Currently, there are two classic ways to efficiently model above distribution: the first is to use a discrete model at topological level; the second is metric level modeling using Monte Carlo sampling.

Topological Bayesian inference compress the high dimensional robot pose space into topological node space [70], whose dimensionality is much lower. The combination of map topology and Bayesian inference can be found in [55], in which Kuipers used unsupervised learning to teach the algorithm how to map observations into different topological nodes. Unfortunately, in such a scheme, how serious the perceptual aliasing problem could be is not known yet. Recently, Modayil [67] combined the dynamic Bayesian network with a map topology to build a large scale map.

In the Monte Carlo sampling scheme, the distribution of vehicle's pose within the map is represented as a set of weighted particles [21]. Thrun [95] first introduced the Monte Carlo approach for robot localization, and demonstrated attractive robustness and efficiency. Ranganathan [80] further combined Monte Carlo sampling with a map topology under a Markov localization framework. By doing so, the correct map topology can be learned from the space of all possible

topologies.

Stewart [88] recently developed a hierarchical Bayesian approach for the revisiting problem. This approach divides the environment into a connected set of local map patches. A hidden Markov process is modeled to represents the transitions between these patches.

2.4 Optimization

The goal of SLAM is to estimate the states of both the environment and the robot pose, by fusing information that the robot obtains at different times. For this type sensor filtering, Dissanayake [20] proved three very important theorems based on Gaussian error assumption, i.e.,

(i) The determinant of any sub matrix of the map covariance matrix decreases monotonically as observations are successively made.

(ii) In the limit, as the number of observations increases, the landmark estimates become fully correlated.

(iii) In the limit, the covariance associated with any single landmark location estimate is determined only by the initial covariance in the vehicle location estimate.

Although these theorems provide solid theocratic foundation for SLAM research, they may not all be pertinent in real world applications, as discussed in [32]. The convergence of the map covariance matrix may be too slow for most mapping applications. A lot of algorithms have been developed during the past

few years to improve the accuracy and efficiency of the filters in SLAM. They can be generally classified into three categories. First, there are filters that describe the absolute map states explicitly in a fixed global coordinate system; second, there are filters which analyze the relative relations between map items (these filters resemble graphical models); third, there are filters which represent the map as a hierarchy and try to estimate the deployments of submaps. These three kinds of approaches are reviewed in the following sections.

2.4.1 Filters using absolute pose information

When perceptual information and prediction from interior motion readings conflict with each other, they are fused together under a certain weighting strategy. Filtering techniques are widely used to implement the weighting. Extended Kalman filter (EKF) [3][71] is one of the most popular types of filter in SLAM research: given a nonlinear dynamic system whose state is written as x_k, with observations z_k, where, $k = 1, 2, ...$ EKF can recursively calculate

$$\hat{x}_k = (\textbf{prediction of} \quad x_k) + \mathcal{K}_k \cdot [z_k - (\textbf{prediction of} \quad z_k)] \tag{2.1}$$

Here, $\mathcal{K}_k$ is called the Kalman gain. It determines how much prediction and observation can contribute to the estimation of $\hat{x}_k$. This recursion provides the optimal minimum mean-squared error (MMSE) estimate for x_k, assuming the prior estimate $\hat{x}_{k-1}$ and current observation z_k are Gaussian Random Variables (GRV). In the EKF, the state distribution is approximated by a Gaussian Random Variable (GRV) that is then propagated analytically through the 'first-order' linearization of the nonlinear system. These approximations, however, can introduce large errors

in the true posterior mean and covariance of the transformed (Gaussian) random variable, which may lead to sub-optimal performance and sometimes divergence of the filter. This is partly caused by the fact that we try to use one unified processing scheme (e.g. the Extended Kalman Filter) to perform all the data fusion tasks in SLAM.

Guivant [36] proposed a 'batch' variant of EKF called *Compressed EKF* (CEKF). This approach restricts the online-updating to the local region around the robot. Only when the robot leaves this region is a global updating performed and the accumulated information propagated to the full EKF. Similar to this local region concept, Bosse [8, 7] segmented the map into a graph of interconnected local submaps. Another similar technique called 'map joining' was independently developed by Newman [90].

The Unscented Kalman Filter (UKF) [48] addresses the approximation issues of the EKF. The state distribution is again represented by a GRV, but is now specified using a minimal set of carefully chosen sample points. These sample points completely capture the true mean and covariance of the GRV, and when propagated through the true nonlinear system, then capture the posterior mean and covariance accurately to the third order for any nonlinearity.

However, when there are discrete state and measurement models, without additional restrictions, the performance of KF degrades drastically as the predicted estimate tends to be updated by wrong measurements. Oussalah [75] uses more than one single model to increase the robustness of tracking.

In contrast to the EKF and UKF, particle filters completely abandon the Gaus-

sian uncertainty model and rely completely on samples to represent the PDF of target probability distribution [19, 46].

The complexity and robustness of a particle filter heavily depends on the number of independent particles involved. Although a large number of particles can provide a comprehensive description for the target PDF, they require huge computational resources. Fox [30] developed the KDL sampling algorithm to adaptively select the number of particles, which efficiently removes 'redundant' samples, and thus reduces the complexity of the particle filter.

From the implementation perspective, there may exist situations where the rate of incoming measurements is higher than the update rate of particle filter. Kwok [58] distributes the samples among the different observations that arrive during a filter update. Hence, the target distribution can be approximated using a mixtures of sample sets, which avoids potential filter divergence due to an insufficient quantity independent samples.

Montemerlo [69] developed the FastSLAM technique, which tries to approximate the probability distribution of map in an n dimensional map space using Monte Carlo sampling. This technique is also capable of solving the data association problem simultaneously. However, this algorithm's efficiency largely depends on the number of particles involved, and the number of particles necessary for closing a large loop is not yet clear.

During the past few years, other than the conventional sensors such as range scanner and odometry [36], people have tried to exploit information from different sources. What interests the author most is the fusion of global priors [110, 74].

These techniques incorporate a strong constraint on the local optimization process. This constraint can be regarded as an onboard sensor, whose error is bounded over time and distance, as shown in Figure 2.1.

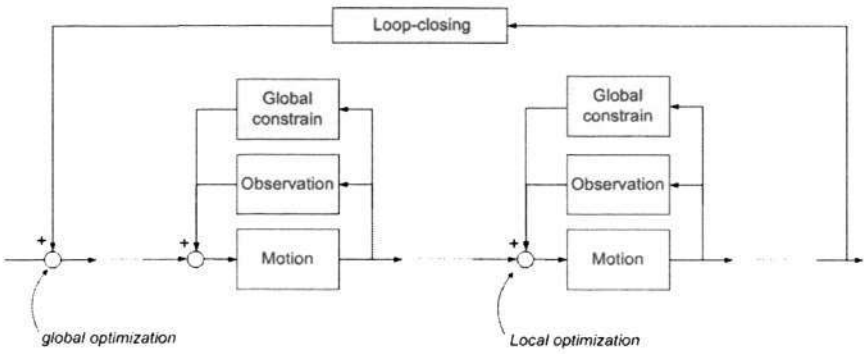


Figure 2.1: Diagram for SLAM with global constraint.

2.4.2 Filters using relative pose information

Processing the uncertainties of SLAM using information form rather than covariance form leads to the development of Sparse Extended Information Filter (SEIF) [96]. SEIF is efficient to incorporate a new landmark in the information matrix, however, it could be time consuming to acquire a useful map representation because a system with n linear equations must be solved. Relaxation is used here to solve the linear system. Also using relaxation, Duckett [23] and Frese [32] developed iterative algorithms for the linear equation system that exists in a maximum likelihood estimation.

Duckett used a genetic algorithm (GA) to solve the large scale optimization problem in SLAM. The basic idea of GA is to represent the a hypothesis of map by a piece of chronometer with a binary item. A population of such genes can be constructed and evolves overtime. A criterion referred to as fitness is used to select

excellent genes and filter out the poor genes.

2.4.3 Filters using submap model

Based on the work in [90], Estrada [28] further introduced the hierarchical SLAM, which only guarantees the consistency at the submap level. This is achieved by ignoring the correlations between map items (local features) from different submaps. Similar to the relaxation-based techniques mentioned previously, the hierarchical method will also result in a linear system which is time-consuming to solve. Estrada employed the iterative extended Kalman filter (IEKF) to calculate the maximum likelihood and demonstrate very efficient loop-closing capability.

2.5 Extensions of localization and mapping

The combination of data modeling, data association, and data filtering constructs the core of SLAM, and is indispensable in every SLAM algorithm. The success of SLAM actually facilitates almost every aspect of unmanned vehicle, from path planning to system design, because it addresses the fundamental problem of 'where am i?'. Since it is based on a solid probabilistic architecture, SLAM can be conveniently integrated with other methodologies through a probabilistic 'interface', such as Bayesian inference. Here, two of the most widely discussed extensions of SLAMs are reviewed: active exploration and object tracking on a moving platform.

2.5.1 Active exploration

The original SLAM framework does not include any navigation strategy. An assumption is widely made in the SLAM community that navigation and control of a

vehicle should be handled by a separate module, based on additional information, e.g., from human instructions. Unfortunately, detaching localization and mapping from navigation essentially turns localization and mapping into an open-loop system.

Recently, active SLAM has gained significant attention. Rather than passively processing the sensor measurements collected by the robot on a given trajectory, active SLAM methods try to actively control the robot to explore the environment in a manner that can facilitate its localization and mapping. Entropy [78, 79] is widely employed as a guidance for selecting actions from all candidates. It provides a quantitative representation for the obtained information known to a SLAM algorithm. Accordingly, the objective of active SLAMs is to increase this information in a more efficient manner.

2.5.2 SLAM with tracking

The original SLAM framework also assumes that the environment where SLAM is conducted is static, which precludes the potential ambiguities in data modeling and association caused by moving objects. Such an assumption is commonly violated in most unmanned vehicle applications in urban or cross-country environments. Wang [108, 109] established a mathematical framework to integrate SLAM and moving object tracking, which provides a solid basis for understanding and solving the whole problem. This framework is called SLAM with DATMO, in which the estimation problem is decomposed into two separate estimators. By maintaining separate posteriors for the stationary objects and the moving objects, the resulting

estimation problems become computational tractable.

2.6 Conclusion

This chapter first gives a brief discussion on the essences of the mobile robot localization and mapping problem. The simultaneous localization and mapping is analyzed as a special case of the general localization and mapping problem, with its own special characteristics. Based on the generalized localization and mapping framework as presented in Chapter 1, the author classifies the SLAM papers into three categories according to their contributions to map representation, data corresponding, and optimization. A comprehensive review is presented while two extensions of SLAM, i.e., active exploration and SLAM with tracking, are also discussed.

Chapter 3

Appearance based Bayesian inference for map hierarchy

3.1 The definition of loop-closing in this work

Loop-closing is an open problem in the SLAM community. However, there does not exist a formal definition of loop-closing today. For instance, for an indoor homogeneous mobile robot, it may take a range scan at one position, then take a right turn at the same position (without any locomotion) and take the second scan, finally it may turn back to the original pose and take the third scan. These three steps actually constitute a loop-closing. Essentially, every time a robot moves, it may match things that it has observed before, and this can, to some extent, be regarded as a loop-closing.

It can thus be noticed that, a loop-closing process comprises two parts: first, detection; second, fusion or optimization. To prevent any confusion, in this thesis, 'loop-closure' refers to the location where the vehicle re-visits a place; 'loop-closure detection' is the process to search for a loop-closure; while 'loop-closing' and 'loop-closing optimization' refer to the optimization process that happens after a loop-

closure is detected.

loop-closure detection is a special case of finding correspondences. As explained in Chapter 2, the objective of finding correct correspondences (associations) is to perform optimization (e.g., filtering). Then it is reasonable to classify correspondences according to the optimization process which it supports.

We may start from the simplest case: given 2 observations Z1, Z2 at time instances T1 and T2 respectively. Normally there will be an overlapped portion in these two observations. Data association (e.g., JPDA) could be employed to find correct correspondences. Thereafter, a Kalman filter may yield an updated estimate X2. If a new Z3 is observed at time T3, we may obtain a new estimate X3 with this new observation. However, It should be noted that, if we assume this is a Markovian process, X1 and Z1 will not be used in the new estimation process. On the other hand, we may find that Z1 and Z3 also share an overlapped portion, and we may use a non-Markovian optimization algorithm to estimate X3. Different from the previous one, this estimation is based on all the states and measurements at time T1, T2 and T3.

Here, the author tries to derive a formal definition for loop-closing in this work.

If the data association's objective is to support a Non-Markovian, bundle-adjustment-like process[101], it should be regarded as a loop-closing. If the data association's objective is to support a Markovian process, e.g., a Markovian filter, then such data association should be regarded as normal maneuver.

In this work, after the vehicle travels a circular trajectory and returns to a point in this circle, it associates its current observation with some observation

it collected long ago. Thereafter, all the observations and states between these two visits are processed concurrently to yield an optimal estimate of the vehicle's trajectory. Since this estimation is a non-Markovian process, according to the previous definition, we regarded as a 'loop-closing'.

However, the author also emphasizes that, for different applications, the loop-closing may have different meanings. The above definition should not be regarded as a general definition for SLAM problem. The definition of loop-closing is still open to further discussions.

In a mathematical manner, the objective of this work is to localize the vehicle when it re-visits a place where it has already been to. Given consecutive measurements Z_t, and the previous readings U_t from inertial sensors, a map representation $\mathbb{P}$ is first constructed for the environment. The mapping is performed in a very unstructured cross-country environment, for which a parametric model is unavailable. For this reason, here $\mathbb{P}$ is represented in a discrete manner:

$$\mathbb{P} = \mathrm{g}^1 \cup \mathrm{g}^2 \cup \cdots \cup \mathrm{g}^{N_g} \tag{3.1}$$

where each g denotes a fragment in the space defined by $\mathbb{P}$. Accordingly, at any given moment, the vehicle's pose v could be either 1) within a certain g^x, $x \in [1, N_g]$, or 2) in an unmapped terrain. This alternative situation can be formulated as:

$$\mathrm{v} \in \mathrm{g}^1 \cup \mathrm{g}^2 \cup \cdots \cup \mathrm{g}^{N_g} \cup \mathrm{g}^0 \tag{3.2}$$

where g^0 is a 'dummy' fragment denoting that the vehicle is not within the current map $\mathbb{P}$. At time t, the goal of loop-closure detection is therefore to identify the g^*,

where

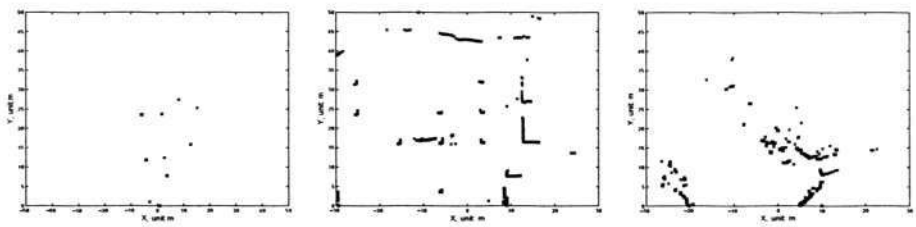
$$g^* = \arg\max_x p(g^x | Z_t, U_t) \qquad x = 0, 1, 2, ..., N_g \tag{3.3}$$

3.2 Introduction to the proposed algorithm

As elaborated in [55], the difficulties of loop-closure detection lie in two-fold: perceptual aliasing, in which different places appear the same; and measurement variability, where the same place appears differently. When the algorithm cannot handle perceptual aliasing, it may take an unexplored place as somewhere already mapped and then generate a false positive report. On the other hand, if the algorithm is too conservative, measurement variability would be difficult to deal with. The algorithm may then report false negative, i.e., the vehicle cannot detect the loop closing although it has already been to the mapped place.

The challenges here also come from the fact that no features or landmarks are available in this cross-country environment. In such environment, it very difficult to model the sensor observations and the environment itself. The testing field is compared with two other types of popular outdoor environments in SLAM, as shown in Figure 3.1. Sub-figure(a), (b), (c) are respectively the photos taken at Victoria Park, Sydney; car park E, NTU; and a cross-country environment. Sub-figure (e),(f),(g) are the raw 2D range scans taken in these environments. It can be observed that, in both Victoria Park, trees cause compact laser reflections that can be accurately and robustly detected; in the car park, there exist rich observable landmarks such as walls and corners. However, in the open jungle environment, no apparent geometrical patterns are available.

(a) Victoria Park (b) Car park E, NTU (c) Open jungle



(d) Vctoria Park (e) Car park E, NTU (f) Open jungle

Figure 3.1: A comparison of environments where SLAMs are performed.

The challenges also come from the scale of the mapping task, as will be detailed in Section 3.6, the total trajectory of the vehicle is as long as 4000 meters. For such large scale mapping, the map representation must be highly compact while the processing routing should also be efficient, otherwise the algorithm will become time-consuming after a short trajectory.

The localization strategy in this work comprises three components. The first one is a hierarchical map representation. In a large scale localization and mapping task, the size of $\mathbb{P}$ can become very large, which makes it impossible to exhaustively examine all the map items. In this work, the map representation $\mathbb{P}$ is organized by a hierarchy. The local regions with homogeneous appearance are grouped together. Such a group is referred to as *submap* in this context. As the vehicle travels, the sequence of submaps extends, and the topological structure of the environment

is also gradually formed. To model the vehicle's motion at the topological level, a lower level representation called *topological node* level is introduced. This level is constructed by segmenting the submaps into smaller fragments at a fixed resolution. By employing topological nodes, the vehicle's metric-level motion can be conveniently transformed to the motion at topological level.

The second component is an appearance model based on raw range scans. All measurements are projected to a low dimensional space using principal component analysis (PCA). For instance, if there exist measurement frames from two submaps A and B, as in Figure 3.2(B). A 3-D coordinate system illustrates this high dimensional measurement space, which is denoted as $x - y - z$. By conducting PCA, these measurement frames are projected to a low dimensional (2-D) space where it is more convenient to segment them. This space is called map space and denoted as $x' - y'$, see Figure 3.2(C). In this map space, projected measurements from the same submap are expected to gather within a compact cluster. This cluster can be approximated by a Gaussian distribution, as in Figure 3.2(D). Compared with the raw range data, the mean/variance representation of Gaussian can significantly reduce the computational complexity, and nevertheless capture the information contained in the raw data.

The third component is to utilize this appearance model to bridge the gap between topological map representation and probabilistic Bayesian inference. It is known that the map topology is inherently symbolic, which is difficult to be processed in a numerical way. The Gaussian distributions discussed in the previous paragraph facilitate the construction of probabilistic observation models for the

map topology.

$$p(z_i|T_x) \equiv p(z_i|S_x) = \exp\left(-\frac{1}{2}\frac{\| \Omega(z_i) - \Omega(S_x) \|^2}{(\sigma_{S_x})^2}\right) \tag{3.4}$$

where S_x is the submap that comprises T_x, and Ω resembles the projection from high dimensional space to the low dimensional space. With this probabilistic appearance model, the popular metric level probabilistic Bayesian inference process can be conveniently transplanted to the higher level in the map hierarchy, see Figure 3.2(E). Using Bayesian network, the algorithm can process a series of measurements sequentially. The estimate based on one frame of measurement will be further examined by its following measurement frames, hence the loop-closure detection will be more informed. By combining these two techniques, the advan-

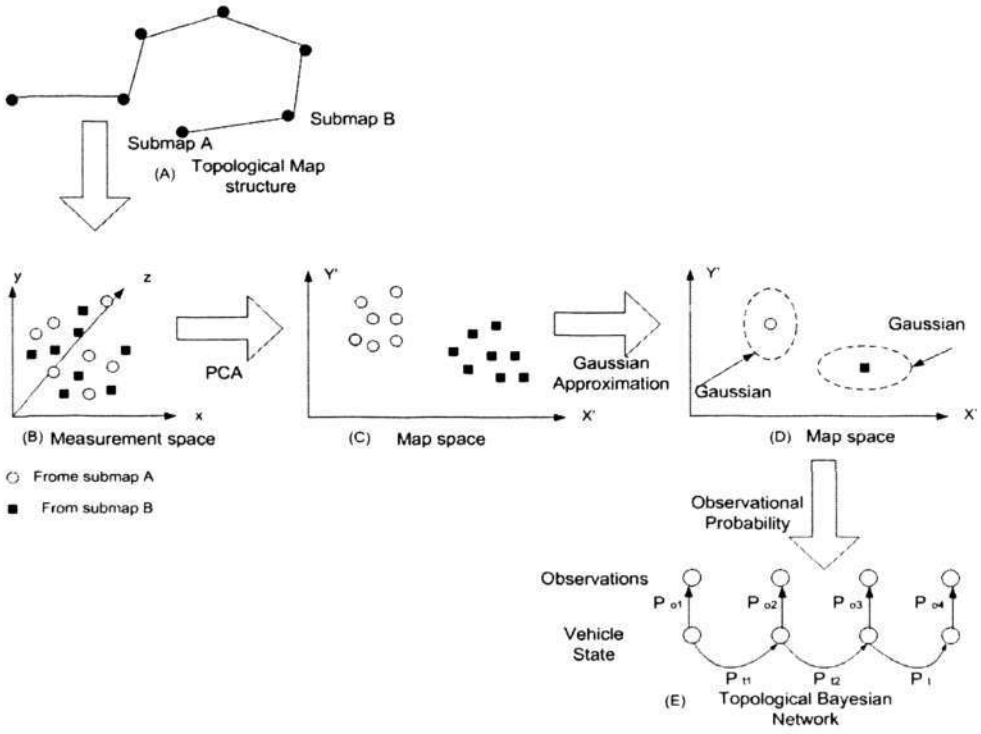


Figure 3.2: Combining symbolic map topology with probabilistic Bayesian inference.

tages of both can be exploited: the presented algorithm is capable of performing Bayesian inference at topological level, while *no* metric-level features are required. This enables the algorithm to detect loop-closure in a cross-country environment, where feature extraction algorithms may fail.

Another important characteristic of the presented algorithm is that, it only uses the sensor information outside the central SLAM estimation loop. Therefore, it does not rely on a potentially erroneous state (vehicle pose) to make a decision regarding the fusion of measurements. So even the location estimated from SLAM is wrong, the loop-closure detection algorithm can still work properly.

This chapter is organized as follows. The next section lays out the strategy of building a map hierarchy; thereafter, Section 3.4 elaborates appearance-based environment modeling based on 2D laser scans; then Section 3.5 explains how to use the appearance-based techniques to perform Bayesian inference; finally, the results and performance analysis are shown in Section 3.6.

3.3 Building the map hierarchy

As explained previously, it is impractical to exhaustively search the whole map $\mathbb{P}$ for a loop-closure. However, it should be bared in mind that, $\mathbb{P}$ is not randomly generated. Instead, the map is highly structured. For instance, $\mathbb{P}$ often embodies a topological structure. This structure provides valuable information for loop-closure detection. To exploit the structural information of the environment, and to process such information by Bayesian inference, the map $\mathbb{P}$ is organized as a 4-level hierarchy. This hierarchy is illustrated in Figure 3.3.

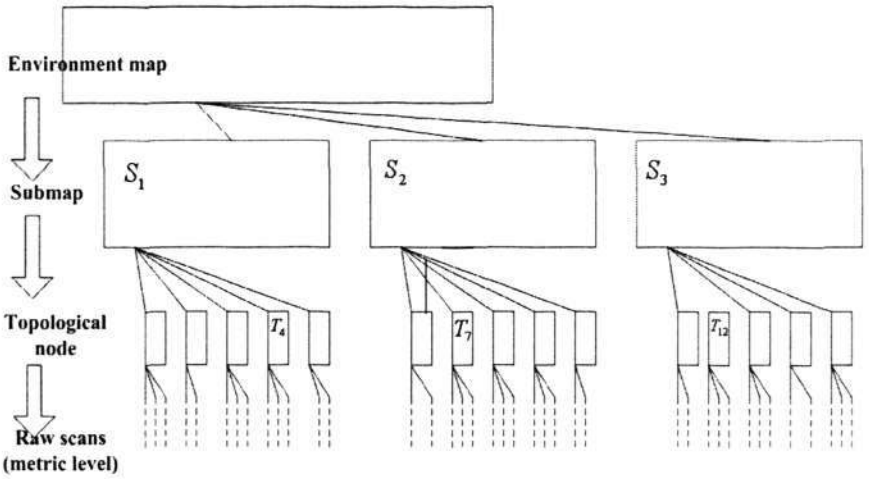


Figure 3.3: The map hierarchy used in this work.

3.3.1 Submap layer

The first layer is called submap layer, which is constructed by dividing the map $\mathbb{P}$ into connected local regions according to the similarities between sensor measurements. At this level, the map is modeled as a collection of submaps S^i:

$$\mathbb{P} = S^1 \cup S^2 \cup \cdots \cup S^{N_s} \cup S^0 \tag{3.5}$$

where S^0 is the dummy submap similar to g^0 and N_s is the total number of submaps, or equivalently, the volume of the submap space.

To build the submap space $\mathbb{S} = [S^1, S^2, \cdots, S^{N_s}]$, the observed measurements (2D range frames) are segmented to different places according to the similarities between them. From the topological perspective, this segmentation process turns continuous sensory experience into a graph of atomic structures, and these structures function as the basic components of a topological map. From the perspective of learning, the loop-closure detection technique in this work is similar to a supervised learning process. It must learn by itself how to distinguish measurement

frames from different regions.

One important characteristic of this supervised learning procedure is that it is performed *online*, there is no off-line training data available. For this reason, the detection algorithm must be self-contained. The sequence of measurement frames is automatically labeled to construct a 'training sample set', from which further classification rules can be learned.

Another important characteristic of this labeling process is that, it is performed online and incrementally, so the map structure is actually encoded in the vehicle's trajectory. If the incoming observations are only labeled according to the similarities between them [55], e.g., using a clustering technique such as K-means [26], the vehicle motion information would not be incorporated in this training process. For example, given two measurement frames similar but far from each other, a completely appearance-based classifier will label them as from the same place. On the other hand, if the map is only segmented according to the vehicle's trajectory, e.g., the volume of the submap [36], observations within the same submap could be distinct. Consequently, it could be difficult to conduct the topological inference.

In this work, the above two kinds of labeling strategies are integrated, as illustrated in Figure 3.4. Either the change of the environment's structure, or the change of the exterior sensor observations, will divide the vehicle's experience into disjoint segments, i.e., initialize a new submap.

Intersections of the road are used to detect the shift between the environment's structures. Since the vehicle is supposed to always navigate itself following the road, such intersections can be indicated by the changes of the heading direction

of the vehicle X^{θ}, which can be easily obtained from onboard sensors such as IMU:

$$\delta_s(t) = \|X_t^{\theta} - X_{t-1}^{\theta}\| \tag{3.6}$$

When its heading changes significantly, a reasonable assertion is that the vehicle has moved from one place to another.

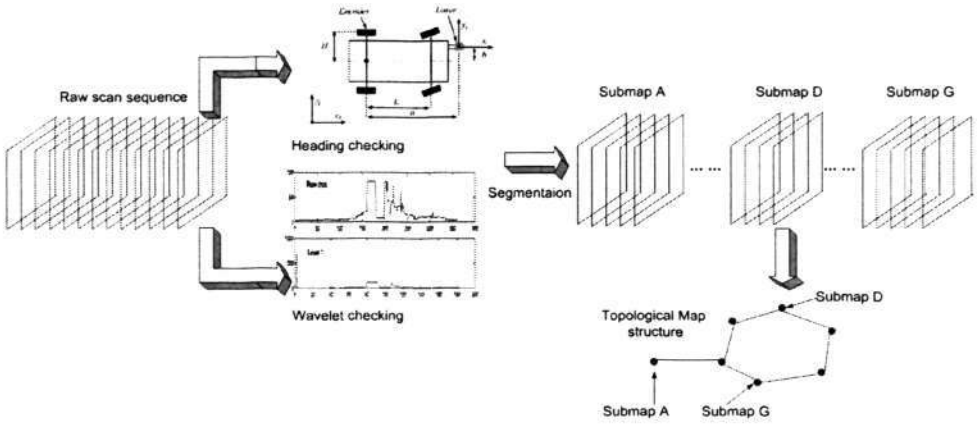


Figure 3.4: The illustrative flow-chat of submap segmentation strategy.

Appearance-based segmentation is not trivial due to the measurement variability. The algorithm needs to capture the major structure of the input range scan, which is often encoded in the low frequency domain. It should also be non-sensitive to the local distracters that exist in the high frequency domain. In this work, wavelet is employed to remove those high frequency details and preserve the structural information contained by the observations. Wavelet is a well-established technique for information compression and noise removal, for instance, here the 3 level db1 wavelet is applied to each frame of 2D range scan, and then a vector whose length is only 1/8 of the original measurement can be acquired, as in Figure 3.5. It can be noticed that the high frequency details is mostly removed in the compressed result.

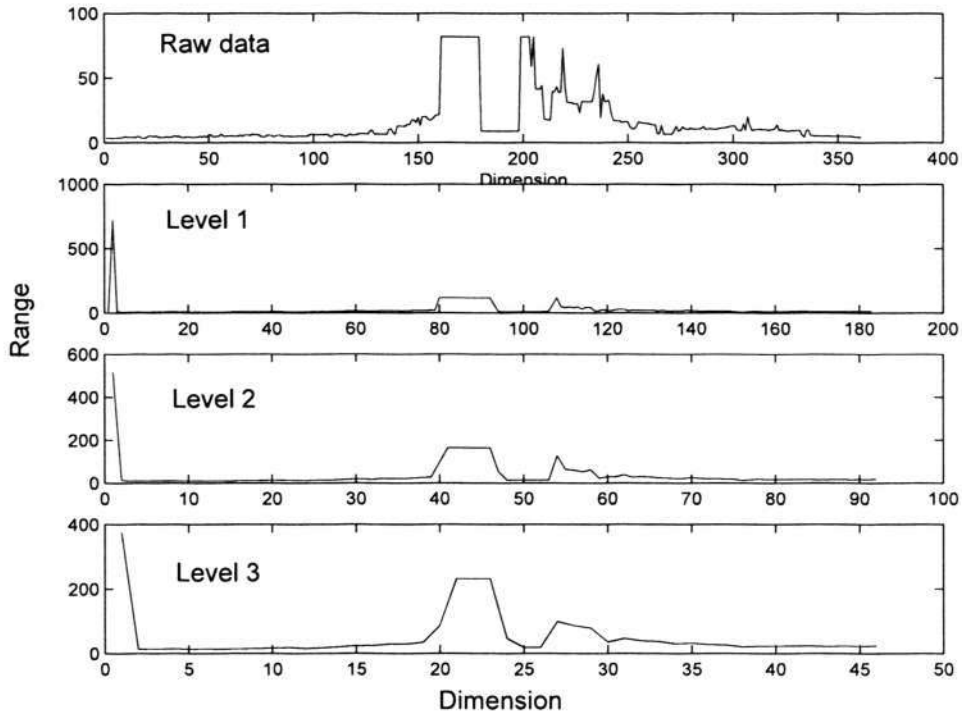


Figure 3.5: The db1 wavelet used in this work.

Mathematically, if the wavelet processing is denoted as $\Theta(\cdot)$, the difference between two consecutive frames z_t and z_{t+1} is calculated as:

$$\delta_{app}(t) = \|\Theta(z_t) - \Theta(z_{t-1})\| \tag{3.7}$$

A submap's shift can therefore be detected at time t if either $\delta_{app}(t)$ or $\delta_s(t)$ is beyond a preset threshold. Readers may be confused to see that two dimensionality reduction techniques, PCA and wavelet, are both employed here. However, please note that the above wavelet segmentation does not have any recognition capability. It is only employed to detect 'new' regions, or equivalently, the shift between submaps. Whether the detected new region has already been mapped or not can only be answered by going through all the previously acquired information, which is performed by PCA.

3.3.2 Topological node layer

Basically, a topological structure comprises a graph of topological nodes, which are connected by edges. In the mapping context, these topological nodes stand for different regions in the environment, while the edges represent how the vehicle can transit from one node to the other. The previous section has elaborated how to 'wrap' the vehicle's metric-level sensor observations to the higher level, similarly, to conduct the Bayesian inference, a topological-level model for vehicle's motion is also necessary.

The transition between submaps alone cannot fully model the motion of the vehicle, for example, a submap can be long enough that given a reading from inertial sensors, it is impossible to predict whether the vehicle is still inside this submap or has left it. Consequently, a lot of valuable motion information is essentially ignored in the inference process at the submap level.

For this reason, in the presented map hierarchy. A lower level representation is further constructed below the submap level to incorporate motion measurements. It is referred to as topological node level, at this level, the map is modeled as a collection of topological nodes that have the same size, let $n(i)$ denote the number of topological nodes contained by the i'th submap, there is

$$\mathbb{P} = T^1 \cup T^2 \cup \cdots \cup T^{N_t} \cup T^0 \qquad N_t = \sum_{i=1}^{N_s} n(i) \tag{3.8}$$

Correspondingly, since a topological node is built from a submap, for each node T^i, $i \in [1, N_t]$, there should always exist a $S^{i'}$ so that: $T^i \in S^{i'}$.

Topological nodes make it very convenient to model vehicle's inertial sensor

measurements in the Bayesian framework. For example, the topological node's length is set to be 10 meters here, so if the vehicle is reported to have moved 12.7 meters, it can be predicted that the vehicle has probably moved to a certain nearby node.

3.3.3 Metric layer

The lowest level in this hierarchy is the metric level where the raw range scans are stored. At the metric level, the map is represented as the collection of raw scans rendered to their corresponding vehicle poses, which is similar to the map used in scan-matching [64]. Given the topological node level representation $T^1, T^2, \ldots, T^{N_t}$, and $m(i)$ that denote the number of scans contained by the i'th topological node, the whole map can be formulated at the metric level as:

$$\mathbb{P} = s^1 \cup s^2 \cup \cdots \cup s^{N_n} \cup s^0 \qquad N_n = \sum_{i=1}^{N_t} m(i) \tag{3.9}$$

3.4 Appearance model for map hierarchy

Different from metric map models that try to register the sensor measurements to a model of the environment, appearance-based models [79, 17, 54] are not designed to capture the relations between observations and the map geometry. Instead, they directly build the environment's representation in the sensor space, i.e., the space spanned by the sensor values themselves. For example, an appearance-based approach can tell what makes the observations in a corridor different from those ones in a square room, but it does not necessarily distinguish the doors or walls. From the perspective of dimensionality reduction, the low dimensional

space built by metric-map-model approaches has definite geometrical meanings, but this may not be true for appearance-based approaches. The 'features' extracted by appearance-based techniques could be uninformative to human eyes. The advantage of appearance-based techniques is that, the *a priori* knowledge required to model the map, such as the definition of lines and corner, is no longer necessary. In other words, appearance-based techniques can work without the conventional 'feature extraction' routing, this is highly desirable when vehicle navigates in cross-country environments where features are difficult to extract. As in Figure 3.6. When it is difficult to extract the feature information which has a tiny percentage in the raw measurements, the appearance model can nevertheless robustly represent the data.

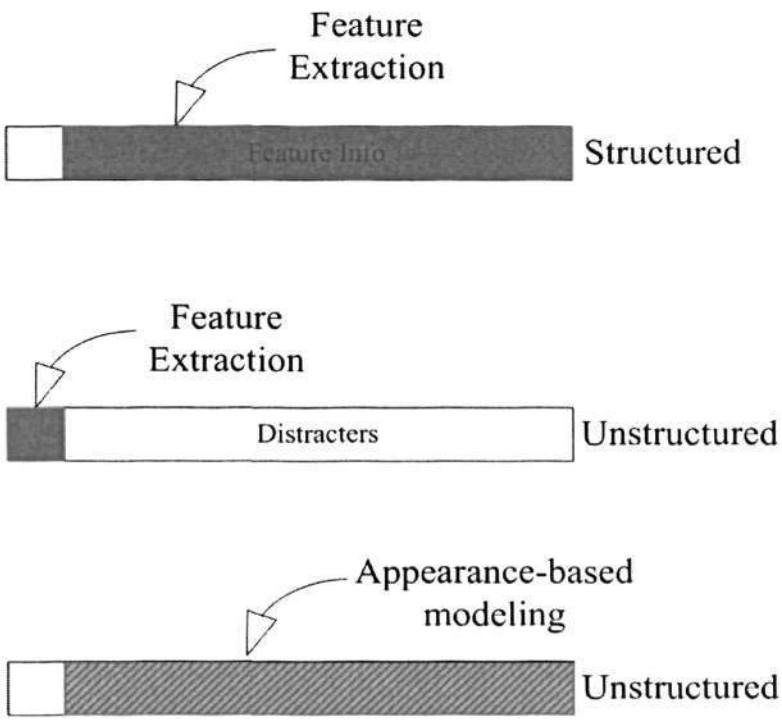
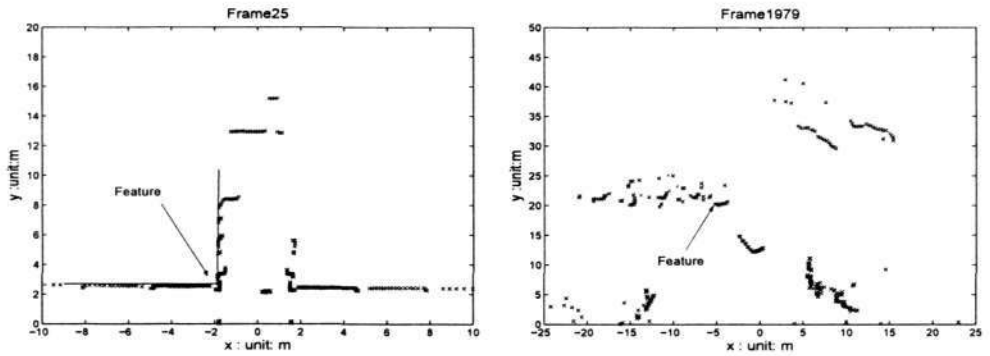


Figure 3.6: Limitation of feature extraction algorithms and the advantage of using appearance model.



(a) Structured environment (b) Unstructured environment

Figure 3.7: Comparison between geometric features in structured and unstuctured environments.

3.4.1 Eigen-representation

Suppose that at time t, the labeling approach discussed in previous subsection has segmented N_s sequentially connected submaps $S^1, S^2, \ldots, S^{N_s}$. For the simplicity of notification, here the number of measurement frames contained by the i'th submap is denoted as f_i, so there is

$$S_i = \left(\mathrm{s}^{i,1}, \mathrm{s}^{i,2}, \ldots, \mathrm{s}^{i,\mathrm{f}_i}\right) \tag{3.10}$$

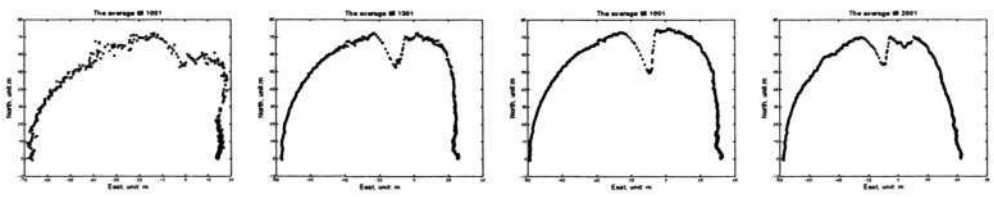
where $i = 1, \ldots, N_s$. To provide a concrete instance for this algorithm, here each frame is assumed to be a typical laser scan including 361 range data. So each frame becomes a vector of dimension 361, or, equivalently, a point in a 361-dimensional space. However, please note that this appearance model is never restricted to laser sensors, all other kinds of input data formats are acceptable for the modeling in (3.10).

Frames of each submap will not be randomly distributed in this huge mea-

surement space and thus can be described by a relative low dimensional manifold. Principal component analysis (PCA) can find the vectors that best account for the distribution of frames within the entire measurement space. These vectors define a subspace of measurement space. For the case of laser range scanner, each vector is of length 361, describes 361 range measurements, and is a linear combination of the original measurements. At time t, the average frame of the whole measurement set is computed by:

$$\Psi = \begin{pmatrix} \psi(1) & \psi(2) & \dots & \psi(361) \end{pmatrix}^T = \sum_{i=1}^{N_s} \sum_{j=1}^{\mathrm{f}_i} s^{i,j} / \sum_{i=1}^{N_s} \mathrm{f}_i$$

As the vehicle moves and new range scans are observed, this average frame evolves over time, as in Figure 3.8.



(a) No. 1 - No. 1001 (b) No. 1 - No. 1301 (c) No. 1 - No. 1601 (d) No. 1 - No. 2001

Figure 3.8: The average of all the collected measurements.

Each frame $\mathrm{s}^{i,j}$ differs from this average by a vector:

$$\phi^{i,j} = \mathrm{s}^{i,j} - \Psi = \begin{pmatrix} \phi^{i,j}(1) & \phi^{i,j}(2) & \dots & \phi^{i,j}(361) \end{pmatrix}^T$$

Then all these $\phi^{i,j}$ are subject to PCA, which seeks a set of normal vectors that can best describe the distribution of the data. In most of the cases, only the dominant part of the distribution is necessary, other details can be ignored [102].

So the normal vector that describes the distribution of the data has a dimensionality much lower. In this implementation, the first λ eigenvectors that correspond to the biggest λ eigenvalues are chosen. These eigenvectors are symbolled as $u^1, u^2, u^3, ..., u^\lambda$, where

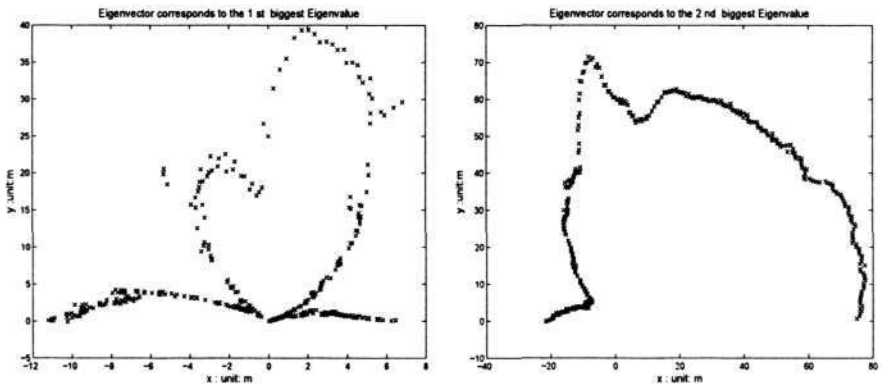
$$u^k = \begin{pmatrix} u^k(1) & u^k(2) & \dots u^k(361) \end{pmatrix}^T \quad k = 1, 2, \dots, \lambda \tag{3.11}$$

These eigenvectors represent the most predominant information about the measurements, and they also define a space with dimensionality λ. Since all the map modeling is conducted in this space, it is referred to here as *map space*. Here the first four of them are shown Figure 3.9. A problem of displaying these eigenvectors is that, they are not supposed to be at the scale of the original measurement frames, so the displayed is the result after normalization. Please note that these figures are expressive: they capture the statistic feature of the measurements, for example, in the first sub-figure, the points in lower part are much denser than the ones in upper part, because this is a common characteristic shared by all frames of laser scans (due to the fixed angular resolution of 2D laser scanner); and all of these 4 sub-figures have the basic shape of a road, because in most of the time, the vehicle travels in a road-like environment.

When a new measurement s^x is available, it is projected into the map-space by a simple operation:

$$w^{x,k} = (u^k)^T (s^x - \Psi) \tag{3.12}$$

where $k = 1, 2, \dots, \lambda$. This describes a set of point-by-point multiplications and summations. These weights (scalar) form a low dimensional vector which can be



(a) Photo 1

(b) Photo 2

(c) Photo 3

(d) Photo 4

Figure 3.9: The eigenvectors corresponding to the biggest 4 eigenvalues.

used to represent a measurement frame:

$$W^x = \begin{pmatrix} w^{x,1} & w^{x,2} & \dots & w^{x,\lambda} \end{pmatrix} \tag{3.13}$$

The vector W^x describes the contribution of each eigenvector in representing the input measurement frame s^x, by treating these eigenvectors as a basis set for measured frames. This low dimensional vector is the core of the appearance-model. It provides a convenient tool to represent measurement frames, as well as local environment (by modeling the distribution of all the measurement frames inside). For convenience, in this context, such projection of a measurement frame in the eigenspace is called eigenframe.

Eigenspace provide a promising way to process the range data. For a typical laser scan at the dimensionality of 361, we can efficiently calculate its representation at dimensionality of less than 20. As can be observed, this appearance-based modeling process is completely independent of any metric-level features or landmarks.

3.4.2 The probabilistic observational models

3.4.2.1 The applicability of Gaussian uncertainty model in eigenspace

The loop-closure detection algorithm proposed in this chapter is based on the assumption that, the whole target environment is constructed by a series of local neighborhoods. Each local neighborhood has a dissimilar appearance, while within one specific local neighborhood, the sensor always observes similar measurements s_n, while n is the number of observations. Based on this assumption, it is reasonable to make a further assumption that s_n can be represented by a Gaussian distribution.

Then the question will be, can we also demonstrate that the projection of d-dimension s_n in the eigenspace also fits a Gaussian distribution.

Since the PCA projection is linear, it can be written as:

$$\mathrm{w} = \mathrm{U}\mathrm{s} + \mu + \epsilon \tag{3.14}$$

while w_n is s_n's projection in the q-dimension eigenspace, and ϵ is the Gaussian noise model of the distribution of s_n, so that $\epsilon \sim N(0, \sigma^2\mathrm{I})$. Based on the Gaussian assumption about s_n, we should have:

$$p(\mathrm{w}|\mathrm{s}) = (2\pi\sigma^2)^{-q/2} \exp\left(-\frac{1}{2\sigma^2}\|\mathrm{w} - \mathrm{U}\mathrm{s} - \mu\|^2 \right) \tag{3.15}$$

With the Gaussian prior:

$$p(\mathrm{s}) = (2\pi)^{-d/2} \exp\left(-\frac{1}{2}\mathrm{s}^T\mathrm{s} \right) \tag{3.16}$$

the marginal distribution of $p(\mathrm{w})$ can thus be calculated as:

$$\begin{aligned} p(\mathrm{w}) &= \int p(\mathrm{w}|\mathrm{s})p(\mathrm{s})d\mathrm{s} \\ &= (2\pi)^{-q/2}|C|^{-1/2} \exp\left(-\frac{1}{2}(\mathrm{w} - \mu)^T C^{-1}(\mathrm{w} - \mu) \right) \end{aligned} \tag{3.17}$$

the model covariance can also be calculated as:

$$C = \sigma^2\mathrm{I} + \mathrm{U}\mathrm{U}^T \tag{3.18}$$

It can thus be observed that, after PCA projection, the distribution of s_n in the eigenspace is still Gaussian. So we can easily retrieve the parameters of this Gaussian distribution. Interested readers are referred to [98] for further discussions.

3.4.2.2 The Gaussian representation in eigenspace

Given a certain submap S^i, where $i \in [1, N_s]$, all measurement frames within this submap can be projected into the map space using (3.12). This operation yields f_i vectors at the length of λ, which are denoted as:

$$\left(W^{i,1}, W^{i,2}, \ldots, W^{i,\mathrm{f}_i}\right) \tag{3.19}$$

The center of this cluster corresponds to the mean of this submap in the map space:

$$\bar{W}^i = \frac{1}{\mathrm{f}_i} \sum_{j=1}^{\mathrm{f}_i} W^{i,j} \tag{3.20}$$

This center's estimate comes with a variance, which is computed by:

$$\sigma^i = \frac{1}{\mathrm{f}_i - 1} \sum_{j=1}^{\mathrm{f}_i} \parallel \bar{W}^i - W^{i,j} \parallel^2 \tag{3.21}$$

the σ_i can also be regarded as the trace of a diagonal covariance matrix, which shows how these $n(i)$ points are distributed in the map space. Here this distribution is approximated as Gaussian:

$$W^i \sim \mathcal{N}[\bar{W}^i, (\sigma^i)^2] \tag{3.22}$$

Given an incoming measurement z_t at time t, the corresponding eigenframe W_t can be obtained by projecting it into the map space using (3.12). The probability of observing z_t conditioned on submap i can be calculated as:

$$p(z_t|S^i) = \exp\left\{ -\frac{1}{2} \frac{\parallel \bar{W}^i - W_t \parallel^2}{(\sigma^i)^2} \right\} \tag{3.23}$$

This formulation is of great importance in vehicle localization. It provides a probabilistic way to model the connection between 2D scanner's observations and the

symbolic topological places in the map, without any knowledge about features or landmarks. By employing this model, Bayesian inference on the map hierarchy can be performed given a sequence of measurements.

3.5 Bayesian inference for map hierarchy

Due to the existence of perceptual aliasing and measurement variability, a deterministic loop-closure detection based on the probability in (3.23) is possibly over-confident. A more reasonable way, therefore, is to fuse the data collected at different time instances to make a 'batch' decision.

In most cases, matching a sequence of measurements with the previously built map is an exhausting task, because the solution space could be exponentially large due to permutation. Markov model offers an efficient way to bypass the permutation. It assumes that only the one step previous action/state can affect the vehicle's current state. This essentially divides the whole expensive matching into a chain of smaller and more tractable testings. Bayes Law then acts as a 'link' to connect these individual testings in a probabilistic manner, so that the local matching results can be propagated through the whole Markov chain to the end. In this work, to improve the robustness of the loop-closure detection, a Bayesian inference process is conducted for the map hierarchy.

3.5.1 Topological Bayesian inference

From the perspective of loop-closure detection, the task of Bayesian inference is to localize the vehicle's *current* position within its *previously* built map (the topo-

logical network). A probability will be assigned to each topological node T^x, the goal of the loop-closure detection is then to find which topological node $\mathcal{T}$ has the highest probability that the vehicle is currently in:

$$\mathcal{T} = \arg\max_x p(T_t^x | Z_t, U_{t-1}, SH_{t-1}) \qquad x = 1, 2, ..., N_t \tag{3.24}$$

where N_t is the total number of the topological nodes, Z_t is the whole set of observations until time t, and SH_{t-1} and U_{t-1} are respectively the set of detected shifts between submaps and transitions between topological nodes till time $t-1$.

At each time instance t, Bayesian inference calculates the vehicle position's distribution over the topological node space, $p(T_t|Z_t, U_{t-1}, SH_{t-1})$. As defined in Section 3.3.1, N_s is used to denote the total number of submaps, so this probability can be further marginalized over submaps S:

$$p(T_t|Z_t, U_{t-1}, SH_{t-1}) = \sum_{j=1}^{N_s} p(T_t|S_t^j, Z_t, U_{t-1}, SH_{t-1}) p(S_t^j|Z_t, U_{t-1}, SH_{t-1}) \tag{3.25}$$

Since each topological node is inside a definite submap, the conditional probability of $p(T_t|S_t^j)$ can be calculated as:

$$p(T_t|S_t^j) = \begin{cases} p(T_t) & \text{if } T_t \in S_t^j \\ 0 & \text{otherwise} \end{cases} \tag{3.26}$$

This probability can be expressed by a function $\nu(T_t, S_t^j)$ that takes value 1 when $T_t \in S_t^j$ and 0 otherwise. Accordingly, the equation (3.25) can be re-written as:

$$p(T_t|Z_t, U_{t-1}, SH_{t-1}) = \sum_{j=1}^{N_s} \nu(T_t, S_t^j) p(T_t|Z_t, U_{t-1}, SH_{t-1}) p(S_t^j|Z_t, U_{t-1}, SH_{t-1}) \tag{3.27}$$

The second item on the right side of (3.27) is the estimate for the vehicle's state in the topological node space. By applying Bayes Rule and assuming that

the estimation problem is Markovian, it can be calculated as:

$$\begin{aligned} p(T_t|Z_t, U_{t-1}, SH_{t-1}) &= p(T_t|z_t, Z_{t-1}, U_{t-1}, SH_{t-1}) \\ &= p(z_t|T_t, Z_{t-1}, u_{t-1}, sh_{t-1})p(T_t|Z_{t-1}, u_{t-1}, sh_{t-1}) \end{aligned} \quad (3.28)$$

Since the observation will not be affected by the vehicle motion and previous observations, theZ_{t-1}, u_{t-1} and sh_{t-1} in the first item on the right side should be omitted. It can be also noticed that, the topological nodes do not have any appearance characteristics, so the observation z_t is actually independent of the topological node and $p(z_t|T_t)$ can be regarded as a constant:

$$p(z_t|T_t, Z_{t-1}, u_{t-1}, sh_{t-1}) = p(z_t|T_t) = c \quad (3.29)$$

The second item on the right hand in (3.28) is for calculating the prior on the likelihood of topological nodes. As explained in Section 3.3, the topological nodes only model the vehicle's motions and do not encode any appearance information. For this reason, they are labeled continuously and the transitions at topological level is completely independent of the shifts at the submap level. Then the item sh_{t-1} can be dropped, and there will be:

$$\begin{aligned} p(T_t|Z_{t-1}, u_{t-1}, sh_{t-1}) &= p(T_t|Z_{t-1}, u_{t-1}) \\ &= \sum_{i=1}^{N_t} p(T_t|T_{t-1}^i, u_{t-1})p(T_{t-1}^i|Z_{t-1}) \end{aligned} \quad (3.30)$$

where $p(T_t|T_{t-1}^i, u_{t-1})$ is the transitional model in the topological node space, and $p(T_{t-1}^i|Z_{t-1})$ is the state of the topological node in previous step.

The second item on the right side of (3.25) is the estimate for the vehicle's state in the submap space. Similar to the estimate for the topological node in (3.28),

this likelihood can be computed using Bayesian inference based on a Markovian assumption:

$$p(S_t^j|Z_t, U_{t-1}, SH_{t-1}) = p(z_t|S_t^j, Z_{t-1}, u_{t-1}, sh_{t-1})p(S_t^j|Z_{t-1}, u_{t-1}, sh_{t-1}) \quad (3.31)$$

where the first item can be simplified into $p(z_t|S_t^j)$ because the observations are apparently independent of the vehicle's movement and previous observations. This probability represents exactly the observational model constructed in Section 3.4.

The second item $p(S_t^j|Z_{t-1}, u_{t-1}, sh_{t-1})$ is the prior probability of the vehicle's state in the submap space. Since the motion between submaps are independent of the motion between topological nodes, the item u_{t-1} can be dropped and then this probability is calculated as:

$$p(S_t^j|Z_{t-1}, u_{t-1}, sh_{t-1}) = \sum_{k=1}^{N_s} p(S_t^j|S_{t-1}^k, sh_{t-1})p(S_{t-1}^k|Z_{t-1}) \quad (3.32)$$

All above inference can be illustrated by the Bayesian inference network in Figure 3.10:

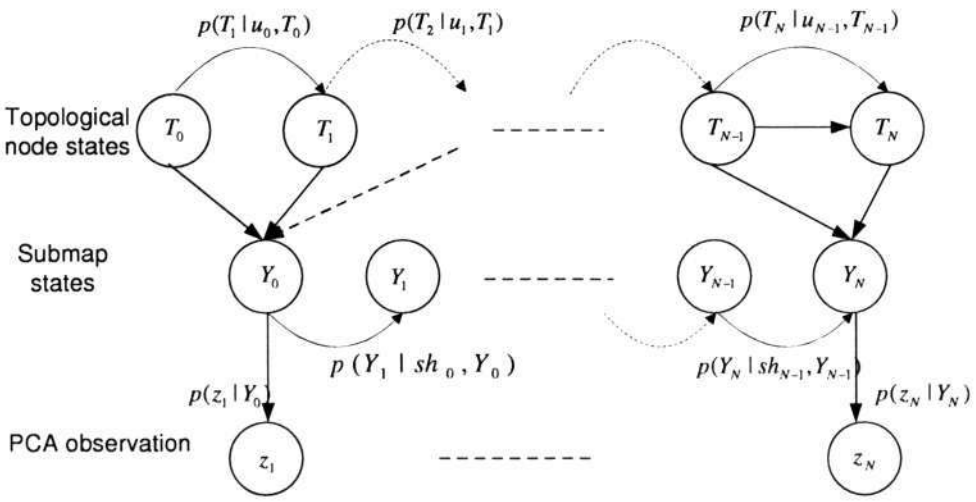


Figure 3.10: The Bayesian inference network in the topological Bayesian inference.

3.5.2 Motion models on submap layer and topological node layer

It can be noticed that, different from conventional Markov localization [31], here two kinds of transitional models (also known as motion model) exist: the transition among submaps, and transition among topological nodes. The transition between topological nodes is formulated as:

$$p(T^i|T^j, u_{t-1}) \tag{3.33}$$

where i and j are the IDs of certain two topological nodes. This probability is used to model the motion predicted from the inertial sensor readings. Given an odometry input v_{t-1}, the number of topological nodes that the vehicle has traveled since last time instance can be computed as:

$$u_{t-1} = \left[\frac{v_{t-1}}{\phi_{node}}\right] \tag{3.34}$$

where ϕ_{node} is the fixed size of the topological node. This equation reveals the advantage of introducing the topological node level in the map representation hierarchy: by doing so, the continuous dead-reckoning process becomes discrete and can thus be integrated with other discrete variables in the Bayesian inference process.

The transitional probability is finally formulated as:

$$p(T^i|T^j, u_{t-1}) = \exp\left(\frac{\left(u_{t-1} - \|i-j\|\right)^2}{\sigma_t^2}\right) \tag{3.35}$$

Obviously, u_{t-1} is not so accurate an estimate because of the round operator [], here a manually set parameter σ_t is introduced to model the confidence of calculating

the number of topological nodes from odometry. For instance, if the vehicle has traveled 50 meters, and the topological node's size ϕ_{node} is 10 meters. The equation (3.34) will tell that the vehicle has traveled 5 nodes, while actually it could be only 4 nodes. σ_t is therefore introduced here to represent such uncertainty.

Basically, the above equation gives bias toward non-loop-closure. It assumes that if one loop-closure has happened, in the following a few steps, the vehicle's trajectory should be consecutive in terms of both time and geography, or in other words, another loop-closure is not so likely to happen again. As observed in the experiment (see Figure 3.16), in the outdoor jungle where the proposed algorithm is designed for, the environment is quite sparse and loop-closing does not happen so frequently. So this assumption will not be over-restricted.

The transitional probability from submap i to submap j is denoted as:

$$p(S^i|S^j, sh_{t-1}) \tag{3.36}$$

where sh_{t-1} represents the report from submap segmentation routing. Apparently, if a submap shift is detected, there could be two possible explanations: first, the detection is correct, the vehicle has moved to the next submap, this situation is given a probability γ; or the detection is a false alarm, the vehicle is still in the current submap, with a probability $1-\gamma$. The transitional probability can be calculated as follows.

$$p(S^i|S^j, sh_{t-1}) = \begin{cases} \gamma & \text{i = j+1} \\ 1-\gamma & \text{i=j} \\ 0 & \text{else} \end{cases}$$

Please note that in above equation, there could exist the situation in which S^i is the first submap and S^j is the last one, here $i = j+1$ still stands.

3.6 Experimental Results

3.6.1 Platform

Two experiments were carried out to test the performances of the appearance-based topological Bayesian inference. The platform for the experiment is a tracked vehicle. For testing purpose, the sensors are also mounted on a pickup in the same layout as they are mounted on that vehicle, see Figure 3.11. More details about the experiments can be found in [12].

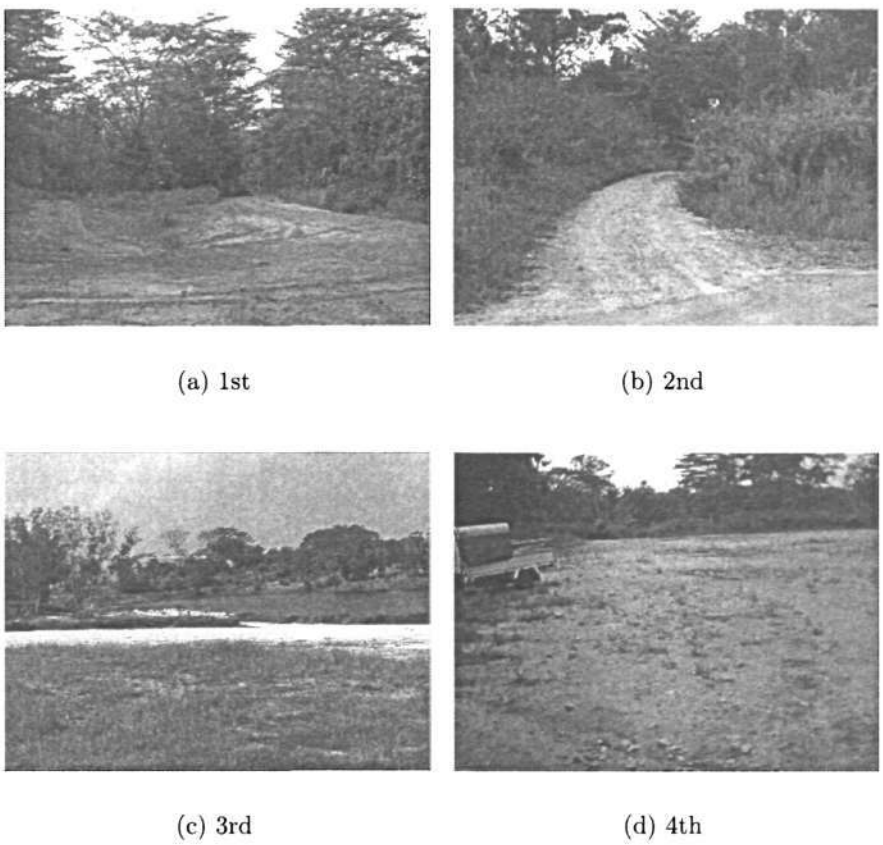
Figure 3.11: The pickup used to simulate the layout of the sensors.

Since the test platform is a tracked vehicle, its motion cannot be read through dead-reckoning. As elaborated in previous sections, the proposed technique utilizes only the local vehicle transformations, i.e., translation and steering. These data can be conveniently calculated from the measurements of speedometer and compass. Although these two sensors' error will accumulate over time, they are already enough for the segmentation purpose.

It must be noted that, the GPS data are used here only for reference and result analysis. During the experiments, no GPS or IMU information is involved.

3.6.2 Testing environment

The experiments were carried out in a tropical jungle environment, some photos are shown in Figure 3.12. It can be observed that, this environment is highly unstructured and no geometric patterns can be identified.



(a) 1st (b) 2nd

(c) 3rd (d) 4th

Figure 3.12: The jungle environment where the experiments were carried out.

The map hierarchy building algorithm is first tested. During the trial, 19,053 frames of 2D scans are collected, the total length of the trajectory is over 3,500 meters. A reference map is built to illustrate the shape the environment, as shown in Figure 3.13. This map is built by rendering the 2D laser scans to the vehicle

poses. These vehicle poses are obtained from a sophisticated GPS/INS module [84], which is very accurate (less than 1 meters). Since the objective here is to examining the loop-closure at topological level, the data with such accuracy can be generally regarded as ground truth. Please note that this map is only used for reference and illustration purpose. It dose not provide any information to the loop closing detection algorithm.

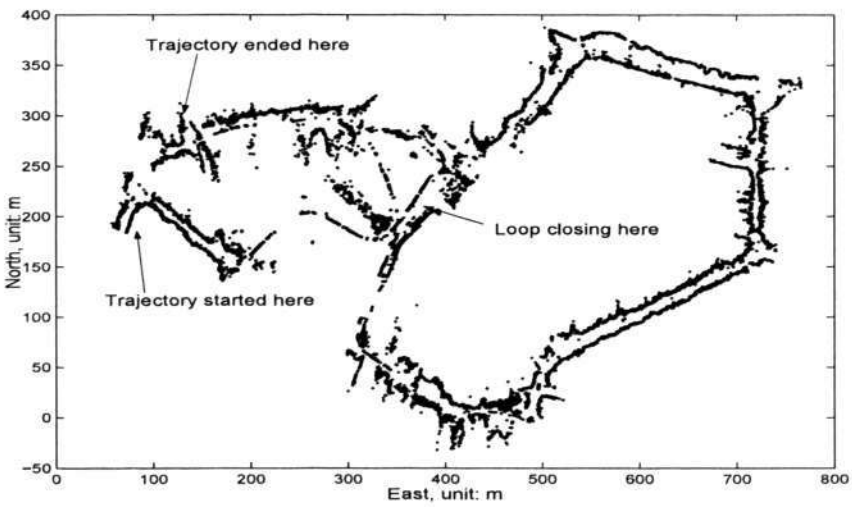


Figure 3.13: The environment where the trial was conducted.

Another experiment is also conducted in which the vehicle traveled a relatively shorter trajectory. It is a square circular environment. During this trial, the vehicle travels around 700 meters, and more than 4000 frames of range measurements are collected.

Photos of the testing field for above two experiments are shown in Figure 3.1. It can be observed that the testing road is not even, the pitch and yaw sometimes cause errors in the mapping result. The the laser beams reflected by the ground can be taken as road boundary, as indicated in Figure 3.14.

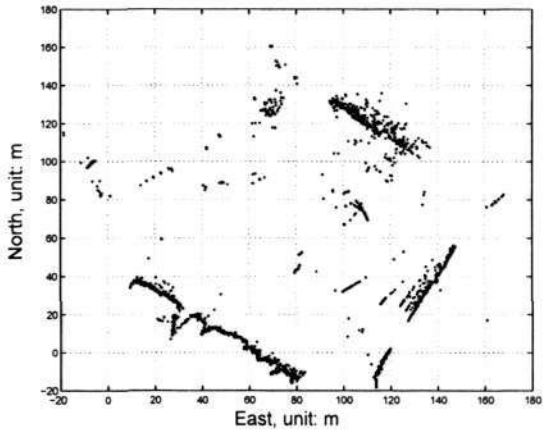


Figure 3.14: The environment with a loop-closure. Just like the map in Figure 3.13, this map is also built from IMU data and for reference use only.

3.6.3 The map hierarchy

For the first environment, the whole map is finally segmented into 35 submaps, based on both the heading direction of the vehicle, and the similarities between continuous measurements, as depicted in Figure 3.15 . The solid dots represent the changes of the vehicle's heading, range from $-\pi$ to π. The thin curve is for the change of the sensor measurements, the peaks of this curve represent the big dissimilarities between successive observations, which indicate the possible transitions from one submap to another. In fact, the dissimilarities computed based on wavelet are much larger. To display them in a single figure with heading's changes, the dissimilarities are all scaled smaller. To keep the figure neat, for every three submaps, only one of them is marked. How the submaps are segmented can be observed from this figure: submap No. 25 is initialized when both the vehicle's heading and the range observations change vastly; for submap No. 19, the heading direction of the vehicle does not change much, but the wavelet checking reports a

high dissimilarity.

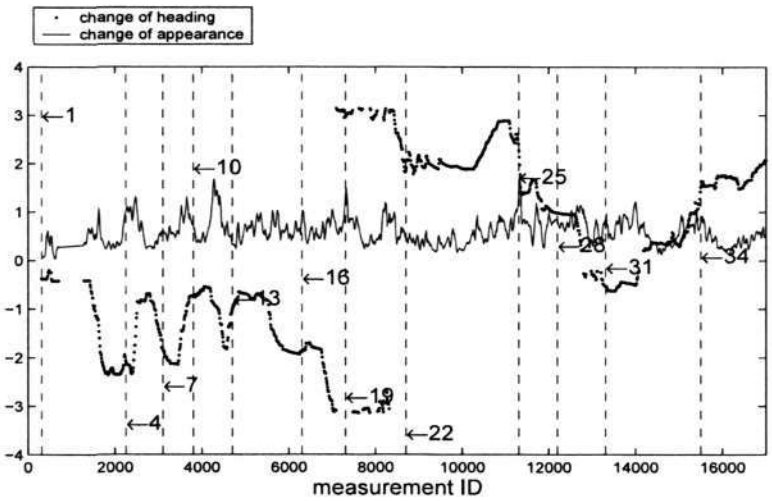


Figure 3.15: The similarities between sequential measurement frames and the changes of vehicle's heading directions.

This online segmentation's results are plotted in Figure 3.16. The submaps are represented as rectangles, the size of rectangle is determined by the length and max width of the road boundary.

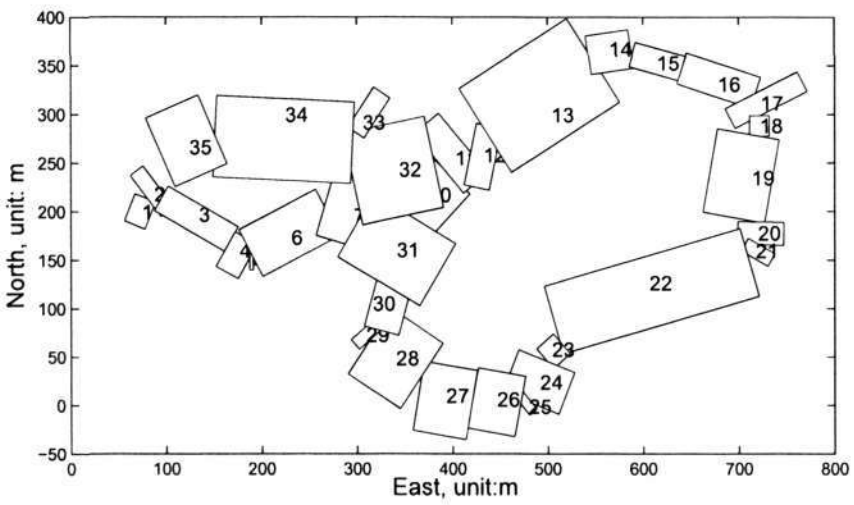


Figure 3.16: The submap segmentation result of the first environment.

For the second environment, the map is segmented into 12 submaps. The first

8 of them are plotted, which correspond to the first loop, as in Figure 3.17. Each submap has its own coordinate system, which is indicated by two arrows. The size of the submap is represented by a dashed rectangle.

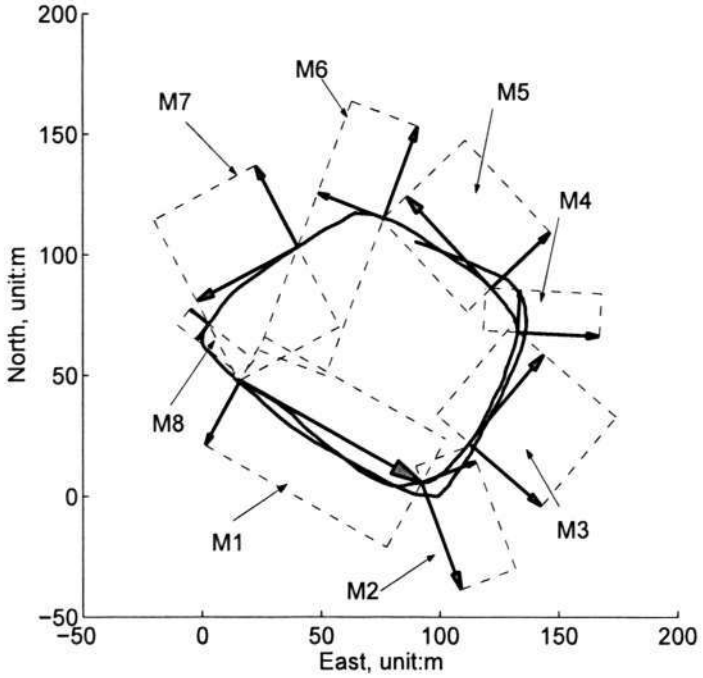


Figure 3.17: The submap segmentation result of the second environment

Topological nodes are obtained by further segmenting the submaps in a fixed resolution, the result is shown in Figure 3.18. Due to the limited space, only one index is plotted for each three topological nodes. As can be observed, the topological nodes are indexed sequentially throughout the map, rather than within each individual submap. This is consistent with the Bayesian inference network in Figure 3.10, which shows that the transitions in topological nodes' space is independent of the submap space.

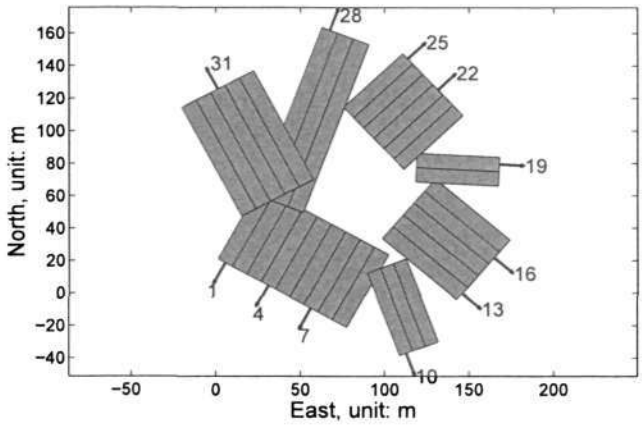


Figure 3.18: Topological node representation of the second environment.

3.6.4 The dimensionality of the map space

To explore the eigenvalue spectrum, Figure 3.19 shows the percentages which represent how much variance the first n eigenvectors account for. The X-axis represents the sorted ID of eigenvectors, the Y-axis represents the normalized accumulated eigenvalues till the corresponding ID. As can be observed, although the data are large in size, the first few of eigenvectors are already sufficient to describe them in the eigenspace. In this thesis, the first 40 of them are used to represent the environment information, or mathematically, $\lambda = 40$. As demonstrated, these 40 eigenvectors can provide about 90 percents of the total observed information.

To examine how λ in (3.13) can affect the performance of the Bayesian inference, the estimation results under different λ are plotted in Figure 3.20. The ground truth is represented by X-axis, coordinates in Y-axis indicate the distances from the estimates to the vehicle's real positions, the unit is topological node. For example, the coordinate $(x = 5, y = 2)$ means at the fifth time instances, the estimate is 2 topological nodes away from the ground truth.

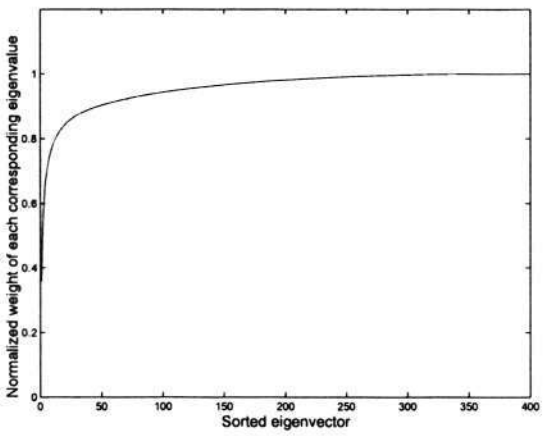


Figure 3.19: The eigenvalue spectrum.

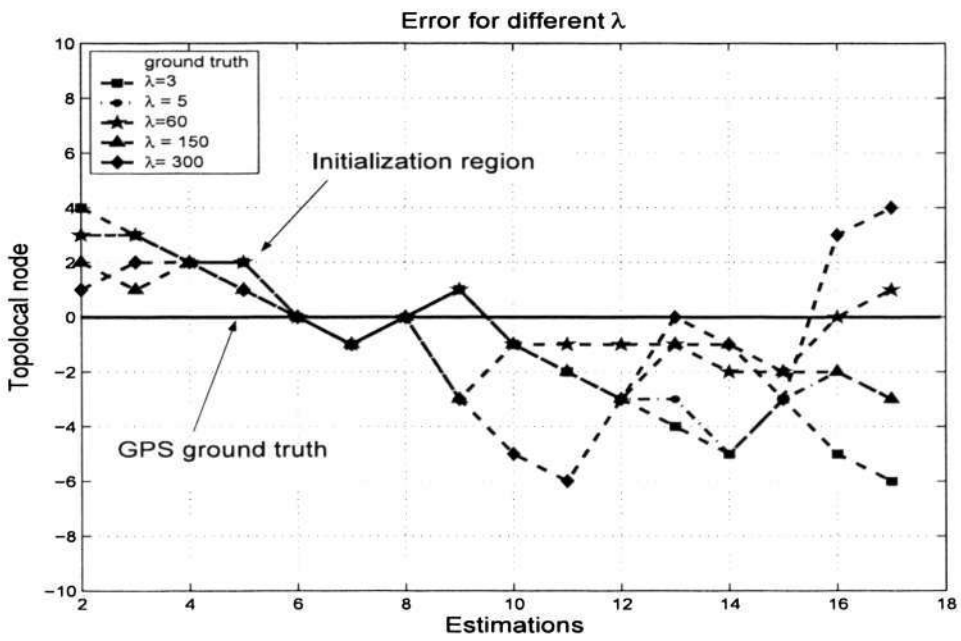


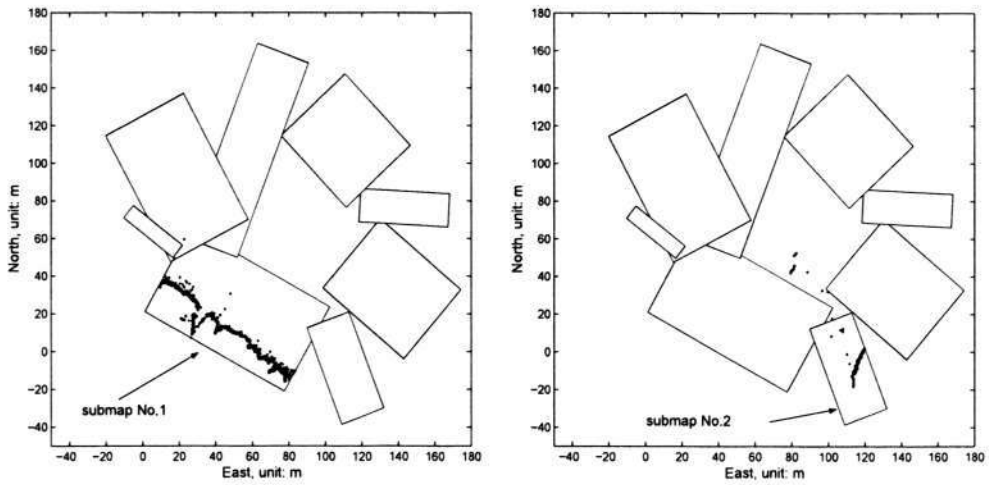
Figure 3.20: The error between the estimates and GPS ground truth.

This figure shows that, the estimation error is big when the λ is either too small(e.g., 3,5) or too big(e.g., 150, 300). This agrees with appearance model's theory: a too small λ means that only very little of the observed information is used in the calculation, in other words, a lot of valuable knowledge is ignored in the PCA process. Bayesian inference then gives poor estimates with insufficient information. To the contrary, a large λ equals to incorporating a lot of redundant information, such as the noises and small distracters. The redundant information can easily confuse the Bayesian inference and then lead to a wrong loop-closure report.

Shortly speaking, the appearance modeling can be regarded as a process to select only the information useful to distinguish different places, and remove those useless. Too small a λ equals to remove too much of the unimportant information, while too large a λ removes not enough unimportant information, and consequently deludes those 'features'. Although such *over-remove* and *under-remove* exist, how to choose the λ is actually not a problem. According to the experiments, for λ ranging from 20 to 80, the inference can all give satisfactory results.

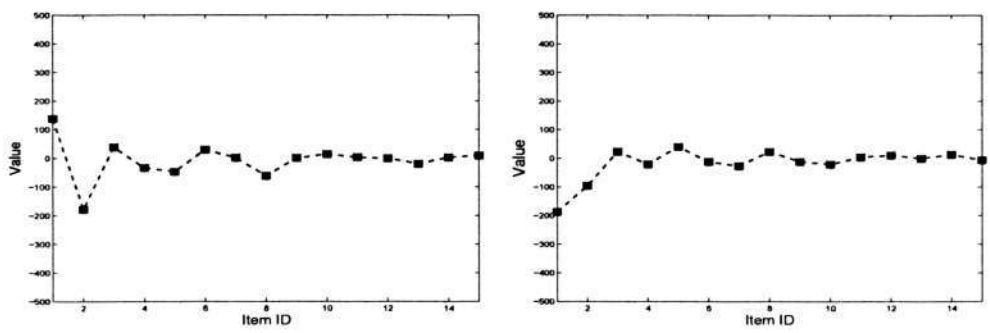
3.6.5 From Euclidean space to mapspace

Being highly compact, the map space modeling can nevertheless catch most of the properties of the environment in the Euclidean space. In Figure 3.21, the submap No. 1 and No. 2 are depicted in the global coordinate, it can be observed that these two submaps are highly different, as a result, their representations in the eigen-space also demonstrate great dissimilarity, as in Figure 3.22.



(a) submap No. 1 (b) submap No. 2

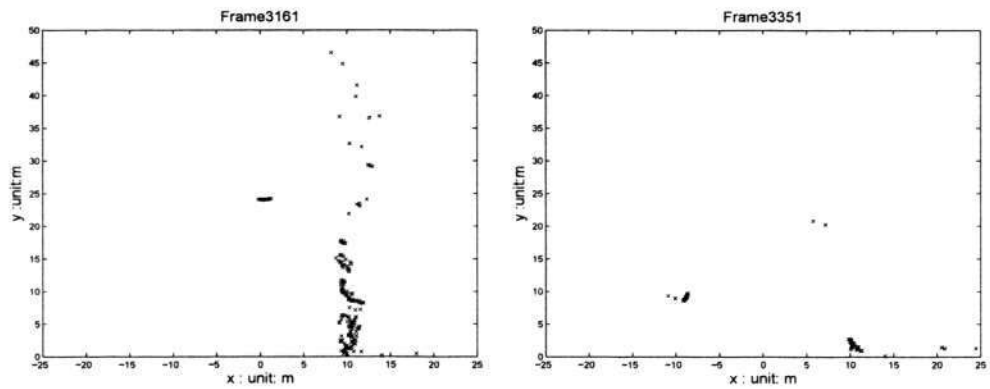
Figure 3.21: The two different submaps in the global coordinate



(a) For submap No. 1 (b) For submap No. 2

Figure 3.22: The averaged Eigenframes used to represent the two submaps in Figure 3.21.

Two measurement frames from submap No. 1 and No. 2 are depicted in Figure 3.23. Their corresponding Eigenframes are shown in Figure 3.24. These two frames are distinct, which agrees with the fact that they are from different submaps. By comparing them with Figure 3.22(a) and (b), the highly similarities between the eigen-representation of submap No. 1 and the Eigenframe No. 3161 can be observed. Such similarity is also found between submap No.2 and No.3351. These facts demonstrate the validity of spotting loop-closure in the map space.

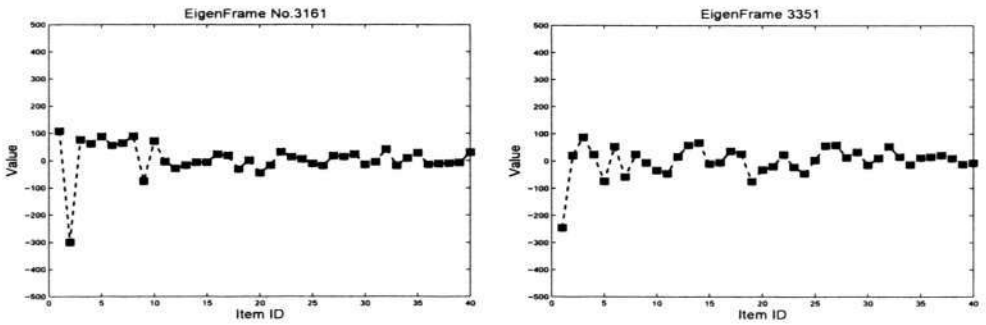


(a) 2D scan No. 3161 (b) 2D scan No. 3351

Figure 3.23: These two frames are from different submaps.

3.6.6 Loop-closure detection results

The observation probability for each measurement frame conditioned on each submap is respectively plotted in Figure 3.25. The flat lines in the center correspond to the time when vehicle stopped (the senor was still working). It can be noticed that there are at least 4 submaps which have similar probabilities at the point of loop-closure (near measurement No. 3000). In this case, the submap No.



(a) Eigenframe No. 3161 (b) Eigenframe No. 3351

Figure 3.24: The Eigenframes computed for the two range scans in Figure 3.23.

1, which has the highest probability, is the correct result. However, it is still dangerous to simply use a Nearest Neighborhood criterion to detect the loop-closure.

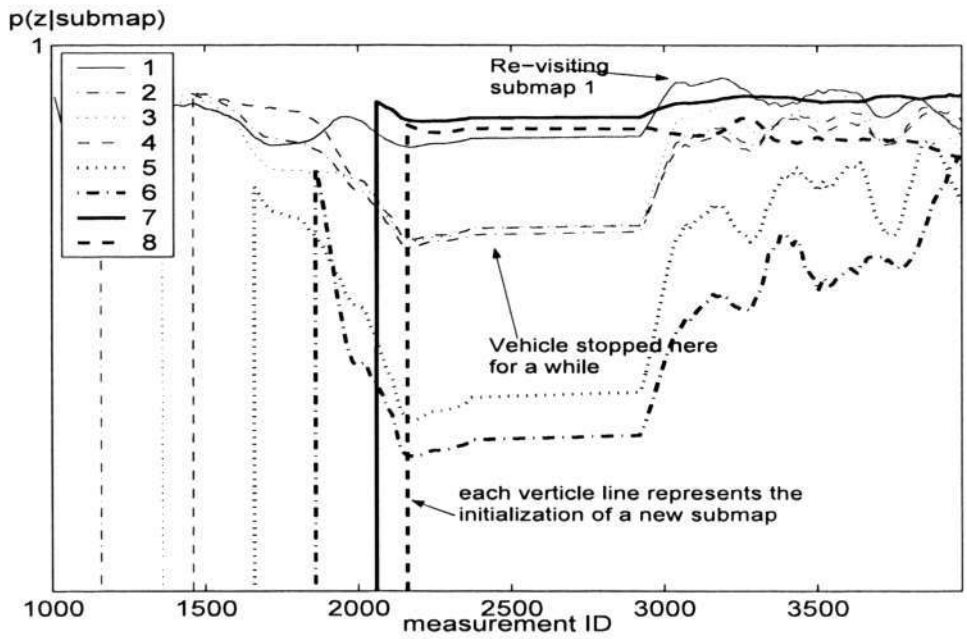


Figure 3.25: The observational probability for measurement z conditioned on different submaps. For the readers' convenience, only the first 8 submaps' curves are plotted. These probabilities are all high because they are not normalized yet. Please note that, no GPS or other positioning sensors' measurements are used to acquire this result.

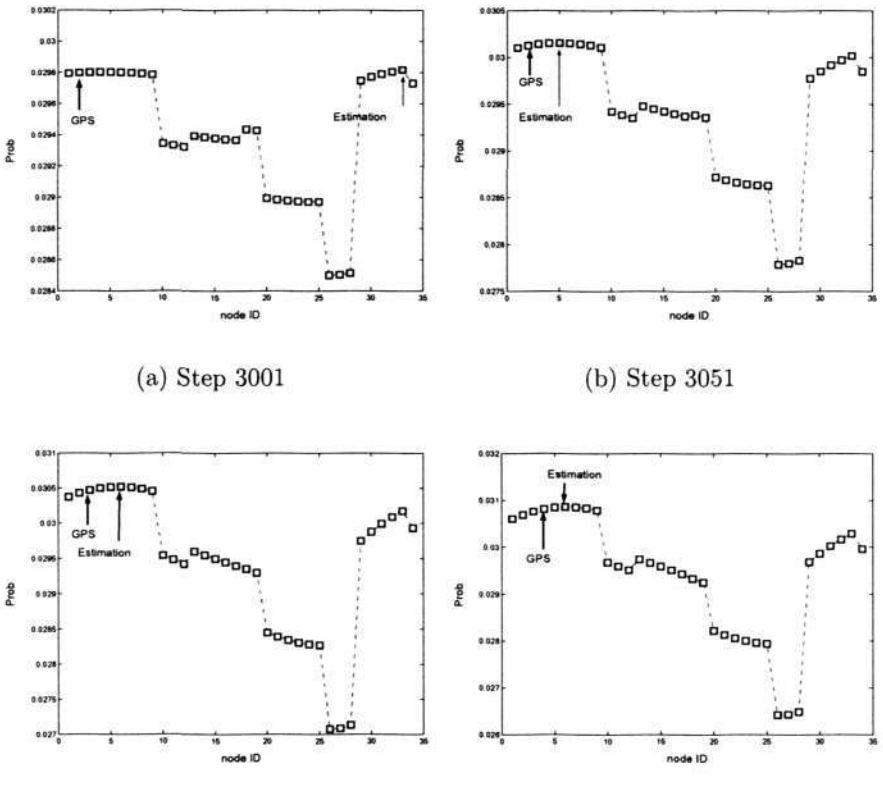
When above possible loop-closures are detected, the topological Bayesian inference procedure is initialized to confirm it. The results of the Bayesian inference are depicted from Figure 3.26 to Figure 3.36. In each figure, the probability distribution over the topological space is plotted. The corresponding geometric information about each topological node can be found in Figure 3.18. The vehicle's actual position read from GPS is marked by an arrow. The proposed algorithm's performance can therefore be observed by comparing the node with the highest probability with the indicated one. Taken that all the sequence of measurements verify the loop-closure hypothesis, a batch decision can thus be made that the vehicle has revisited submap No. 1 at the measurement ID 3000.

As observed in Figure 3.26, the estimates of the proposed algorithm does not match well with the actual position read from GPS, because currently the Markov process is in initialization, current information is not sufficient to correctly compute where the vehicle is.

Figure 3.27 depicts he probability distribution over submaps and the topological network at the position corresponding to measurements No. 3201 - No. 3251. Compared with Figure 3.26, it can be observed that the estimated vehicle position has become much more accurate as new sensor data is received.

A topological shift detected at iteration No. 3301, as in Figure 3.28. The vehicle is supposed to move from its previous submap to a new one. All the submap's observational probability are re-calculated and plotted as above.

Figure 3.29 shows the probability distribution over submaps and the topological network at the position corresponding to No. 3351. Till now, the estimations from



(a) Step 3001 (b) Step 3051

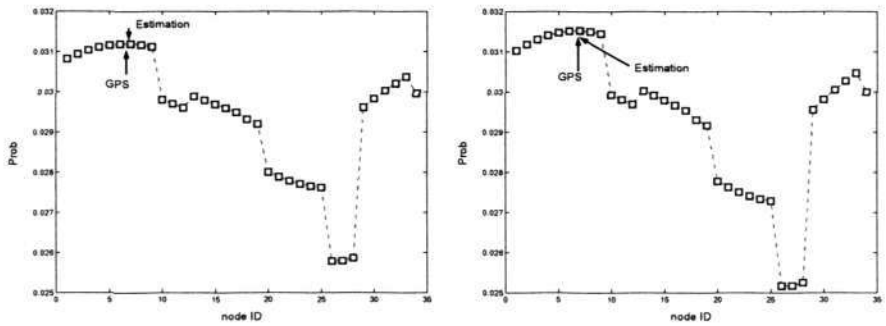
(c) step 3101 (d) 3151

Figure 3.26: The probability distribution over submaps and the topological network at the position corresponding to No. 3001 - No. 3151.

Bayesian inference are satisfactory.

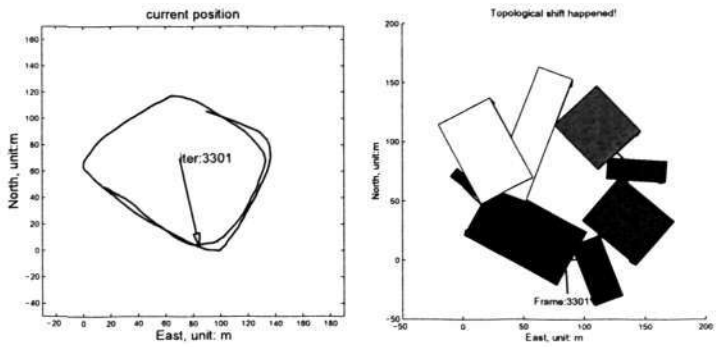
Another topological shift is detected at iteration No. 3401, as depicted in Figure 3.30. According to the information gathered previously, this shift should be detected a short while later. Unfortunately, such a shift is reported due to the fact that the vehicle's current trajectory is different from the one in the first loop. So there is now actually a conflict between the Bayesian estimation and the new observation. If such conflict persists in the following iterations, the Bayesian inference will just report the loop-closure hypothesis at iteration 3000 is false.

Figure 3.31 shows the probability distribution over submaps and the topological



(a) 3201 (b) 3251

Figure 3.27: The probability distribution over submaps and the topological network at the position corresponding to measurements No. 3201 - No. 3251.



(a) GPS position (b) $p(z_t|Y_t^j)$

Figure 3.28: A topological shift is detected at iteration No. 3301.

network at the position corresponding to measurements No. 3451 - No. 3601. The conflict introduced in Figure 3.30 'confused' the Bayesian inference process. Consequently, the error of the proposed algorithm grows during these 4 steps. The question to be answered here is: how should we judge whether such errors are caused by the wrong loop-closure hypothesis, or by the temporary observation error?

Figure 3.32 shows the probability distribution over submaps and the topological

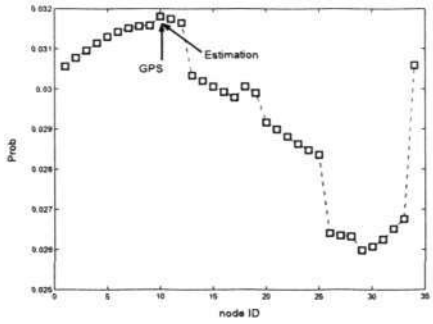
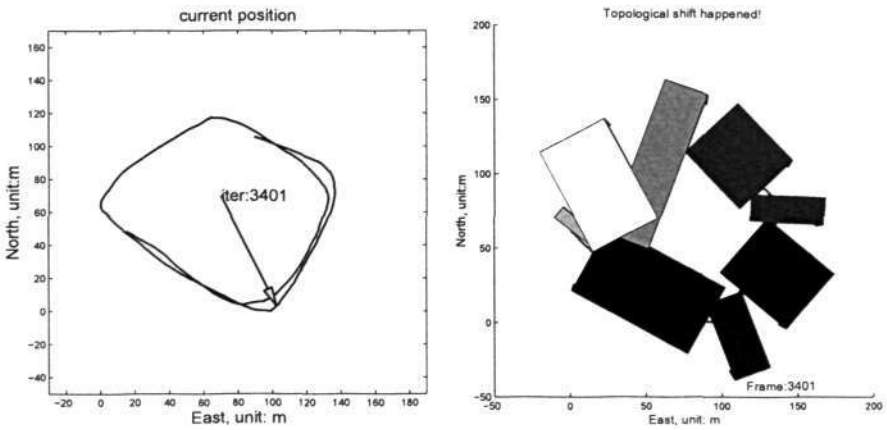


Figure 3.29: At the position corresponding to frame No. 3351.



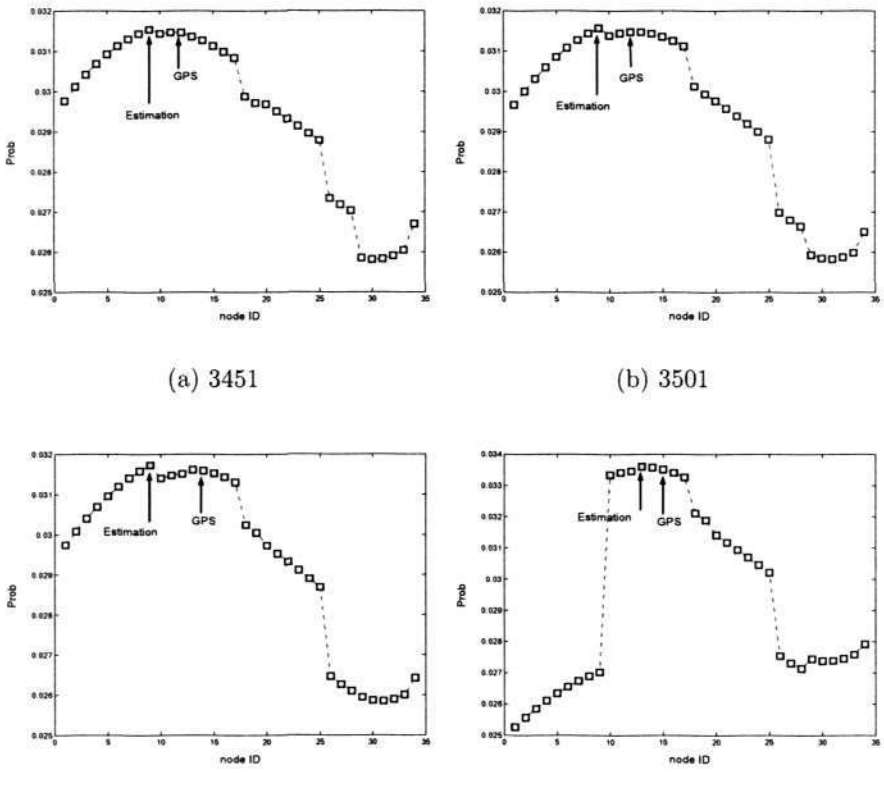
(a) GPS position (b) $p(z_t|Y_t^j)$

Figure 3.30: A topological shift is detected at iteration 3401.

network at the position corresponding to No. 3651. The keep-coming correct observations since iteration No. 3351 finally compensate that error. In this figure, the estimation is quite close to the vehicle's actual position.

A topological shift is detected at iteration 3701, as showed in Figure 3.33. The vehicle's actual position is depicted in (a), the observational probability of each submap is plotted in (b) by gray shading.

Figure 3.34 shows the probability distribution over submaps and the topological network at the position corresponding to No. 3751. The topological Bayesian



(a) 3451 (b) 3501

(c) 3551 (d) 3601

Figure 3.31: Estimates from iteration 3451 to 3601.

inference gives the excellent estimation for the vehicle's pose. Put it in this context, the keep-coming information verify the loop-closure hypothesis. Thereafter, another topological shift is detected, as in Figure 3.35.

The probability distribution over submaps and the topological network at the position corresponding to measurements No. 3851 - No. 3951. The estimation from Bayesian inference matches well with the ground truth, which demonstrates the accuracy and robustness of the proposed algorithm.

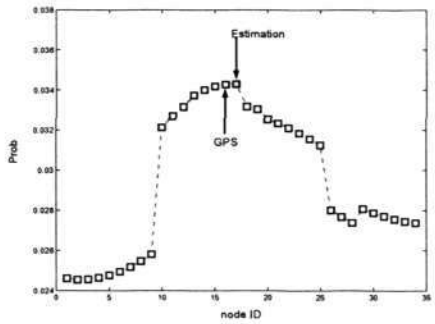
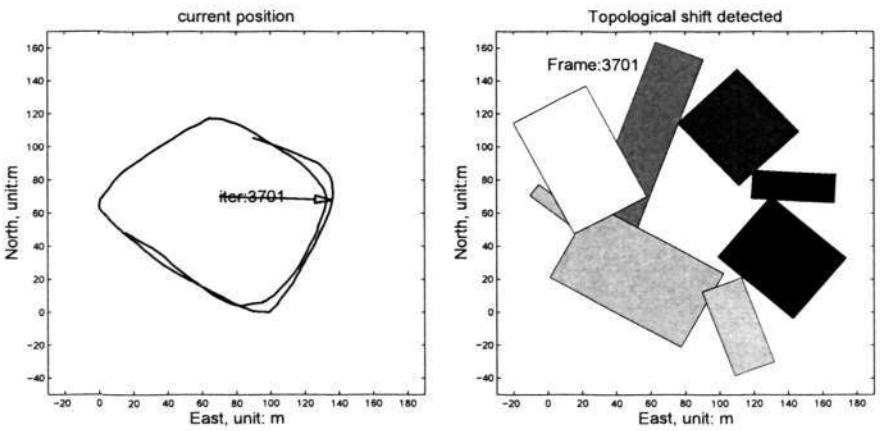


Figure 3.32: Estimate at iteration 3651.



(a) GPS position (b) $p(z_t|Y_t^j)$

Figure 3.33: Topological shift detected at iteration 3701.

3.6.7 Computational efficiency

The algorithms are implemented using Matlab, and run on a computer with one Pentium IV 2.0 G Hz processor. The time requirements of conducting PCA, or equivalently, the parameters of the projection from measurement space to map space, are recorded in Figure 3.37. It can be observed that, as the measurement frames accumulate, the algorithm requires more and more time to calculate the mapping parameters. Within 2,000 frames, the initialization costs less than 10 sec. The processing of raw range scans is handled by the metric level modeling

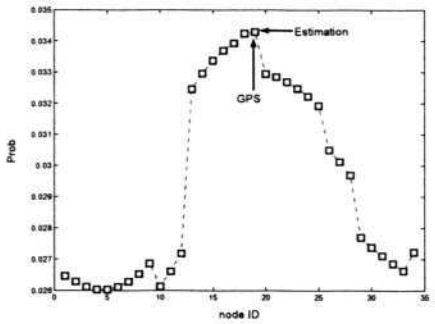
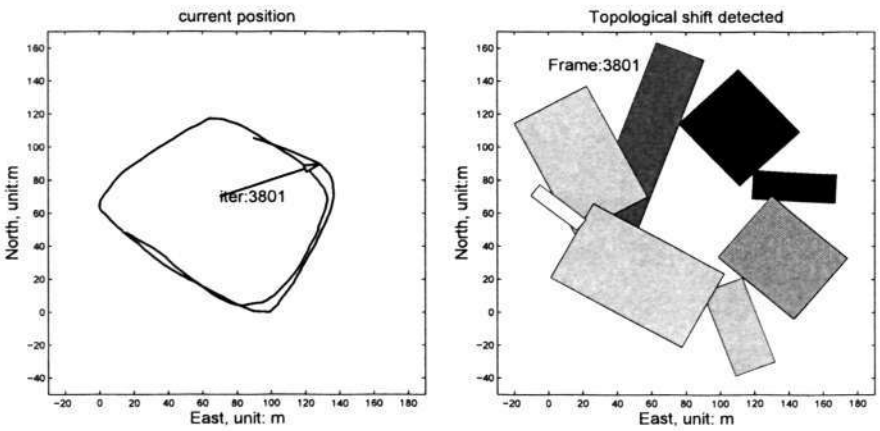


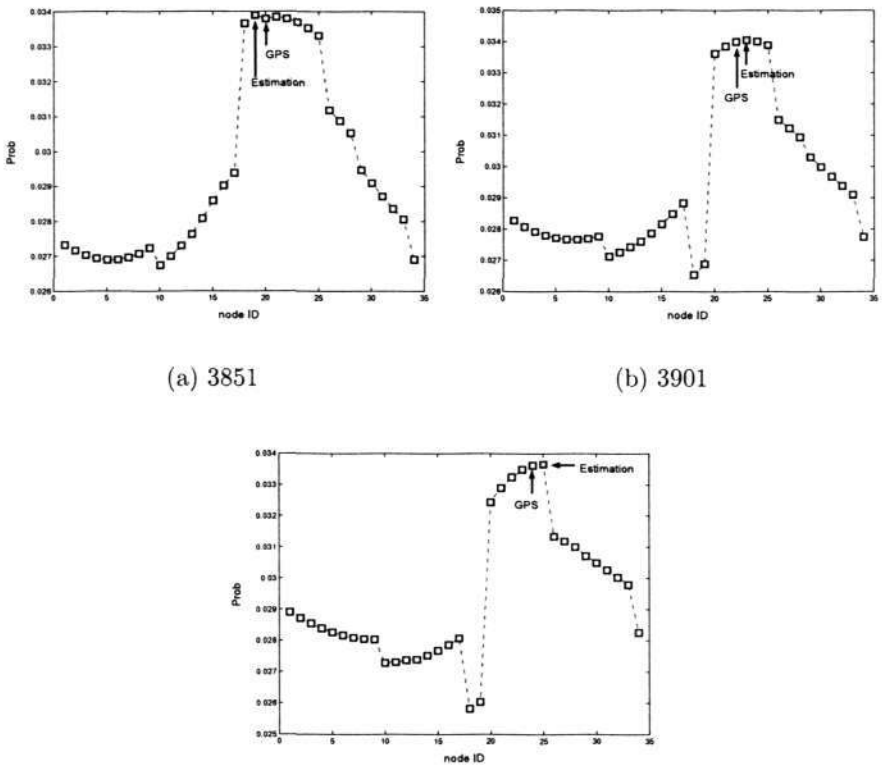
Figure 3.34: Estimate at iteration 3751



(a) GPS position (b) $p(z_t|Y_t^j)$

Figure 3.35: A topological shift is detected at frame No. 3801, then the likelihood of the vehicle's current position is updated.

algorithm which is generally highly efficient, the proposed algorithm here only manages the hierarchical structure of the map, so that it is not required to be embedded into the local mapping process. In fact, the whole Bayesian inference process can be run on an individual computer without sacrificing any performance. Additionally, PCA is conducted only when a submap shift is detected, which does not happen frequently. According to trials, as depicted in Figure 3.14 and Figure 3.13, such shift is detected averagely every 1 minutes.



(a) 3851

(b) 3901

(c) 3951

Figure 3.36: Estimates from iteration 3851 to 3961.

After the projection from measurement space to the map space is computed, the Eigenframe can be computed in constant time (less than 20 ms) for each incoming measurement. Even the submap number scales with traveled distance, in most current SLAM literatures, this number will not surpass 100, and the Bayesian inference can still be run in realtime.

3.6.8 Overlapping at the junction between submaps

It can be noticed that in both Fig.3.16 and Fig.3.17, there exist overlapping regions between submaps. This may lead to the assertion that the summation of the vehicle pose's distribution over the submap space will be more than 1, i.e., the robot could

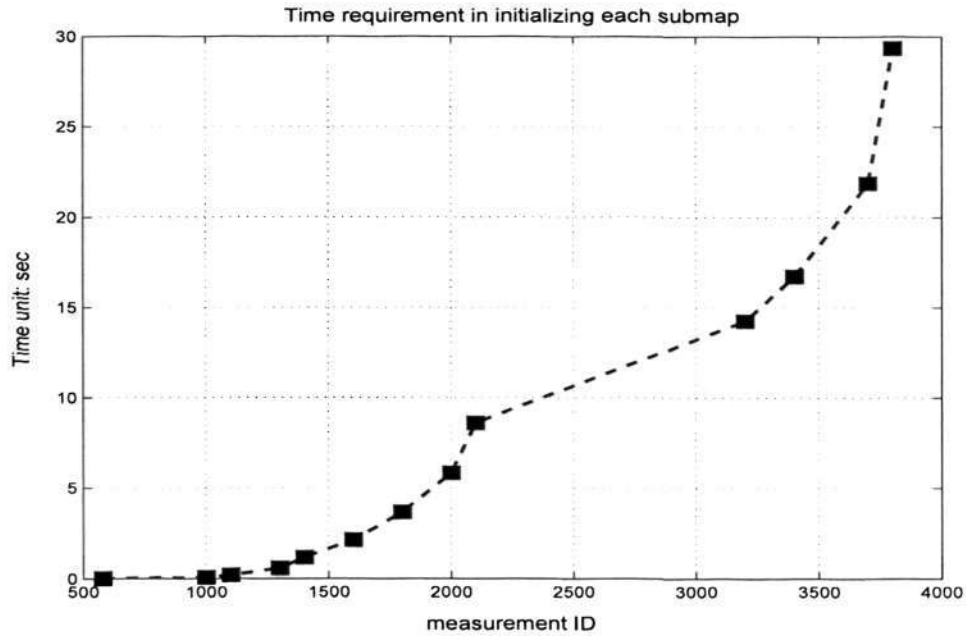


Figure 3.37: The time requirement when a submap is initialized.

be simultaneously within two or more submaps.

However, it should be noted that, the rectangles in Fig.3.16 and Fig.3.17 are only used for illustration, they do *not* represent any geometric shape of the submaps. As can be observed in Fig.3.3, the submaps are constructed by segmenting the sequentially observed range scans. Therefore, 'submap' is a concept existing in the scan space, rather than the Euclidean space. The range scans are observed sequentially. This sequence is determinately segmented by the algorithm elaborated in Section 3.3. For this reason, there will be no overlapping in the submap space. For each new observed scan, either it belongs to one submap, or it belongs to another, there should be no such cases as one scan belongs to both submaps. For this reason, the Equation 3.2 should be justified.

However, the motion of the vehicle is modeled in the Euclidean space, we may

need to project the observations from scan space to Euclidean space for Bayesian inference. Consequently, there could possibly exist overlapping situations, because each scan corresponds to a small region rather than a definite point. So in this work, it is assumed that any two sequential topological nodes are strictly separated.

3.6.9 Viewpoint invariance

As elaborated in previous sections, appearance-based loop-closure detection essentially combines two steps. The first one is a supervised learning process which teaches the algorithm how to distinguish different places. In the second step, it uses the learned knowledge to classify the new observed data.

It is then important to understand that, the appearance-based approaches completely rely on the given samples to understand the environment. If there are no samples from a certain pose of the environment, the appearance-based technique cannot learn from them. Consequently, it cannot recognize such a place in the future, even the vehicle has already been there with a different pose. Shortly speaking, such techniques are originally *not* supposed to be viewpoint-invariant. In the pattern recognition community, one way to solve this problem is to build a more comprehensive training pool that can teach the algorithm the scenes of the same place from different perspectives. This can be implemented by constructing 3D models of the environment and then generating synthetic range scans based on this 3D model. However, a more promising solution is to build not only the 3D shape model, but also the appearance model, as the work in [14]. Meanwhile, it is also possible to solve this problem by employing viewpoint invariant sensors.

3.6.9.1 To calculate rotation by Hough Transform

In addition to the main appearance-based topological inference loop, the author also introduces here a scan alignment technique that can be used to loose the viewpoint invariant constraint. The basic idea is to align the input scan and the submap, thereafter the input scan can be regarded as captured from the same viewpoint as the submap is observed. This alignment method is based on the popular Hough Transform (HT) and the first order moment.

Hough Transform has been widely used in feature detection in images and laser scans [44]. The basic idea of HT is to construct a discrete line parameter space, and let input data points vote for the most suitable parameter.

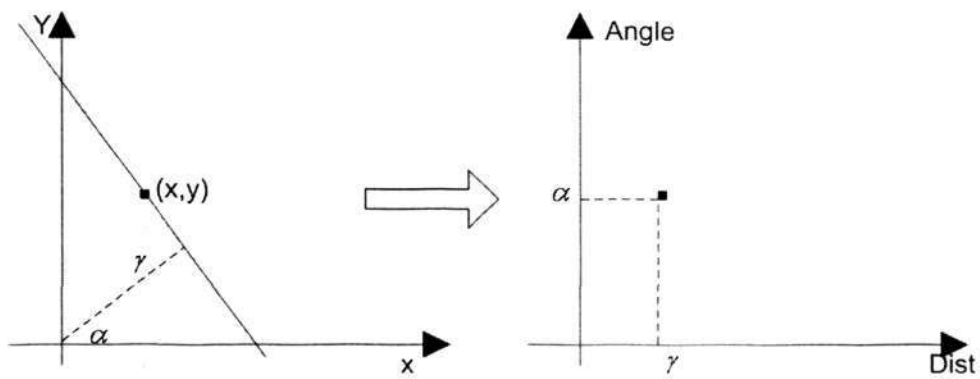
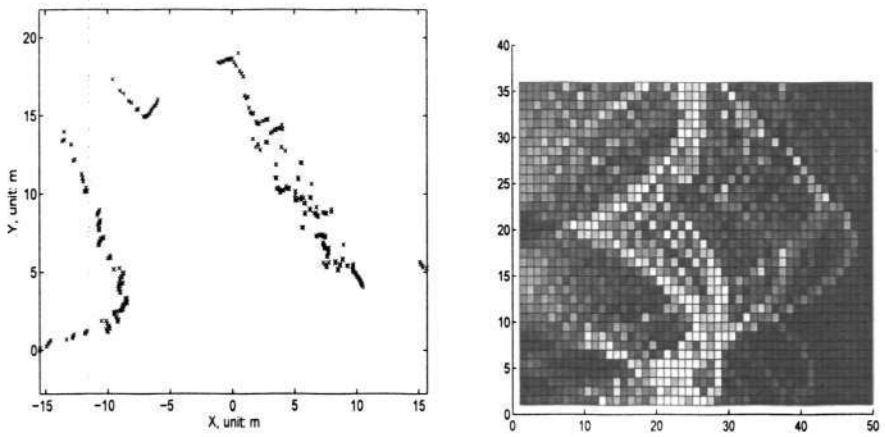


Figure 3.38: Hough Transform projects the points in 2D Euclidean space to a Hough space, which describes the parameters of lines.

For the alignment task, we regard the whole scan as a line. All the 2D points are projected into the Hough space, the heading voted by the most points will be regarded as the heading of the scan.

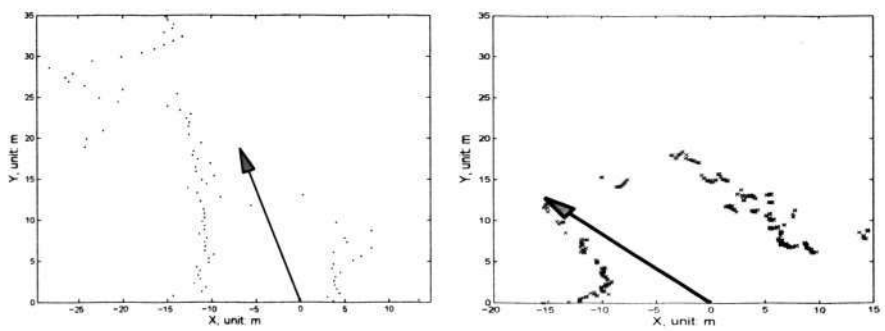
Since a submap is a collection of scans collected at different time instances, the HT can also be applied to submaps. We can easily obtain the heading of a scan



(a) Scan in 2D Euclidean space (b) Voting result in Hough space

Figure 3.39: The projection from 2D Euclidean space to the Hough space. The x-axis in the right figure stands for γ, its resolution is 0.4 meter. The y-axis is α, its resolution is 5 degrees.

and a submap, and then align them. In Fig. 3.40, the heading direction of submap No. 1 and scan No. 2085 is plotted. Correspondingly, their heading directions are also calculated by HT.



(a) Submap No. 1 and its heading (b) Scan No. 2085 and its heading

Figure 3.40: Hough transform is used to calculate the heading of submap and laser scan. The computed heading is marked by an arrow.

The alignment result calculated by Hough Transform is plotted in Fig. 3.41.

By comparing it with Fig. 3.40, it can be observed that, the HT can accurately calculate the rotation between the submap and the scan.

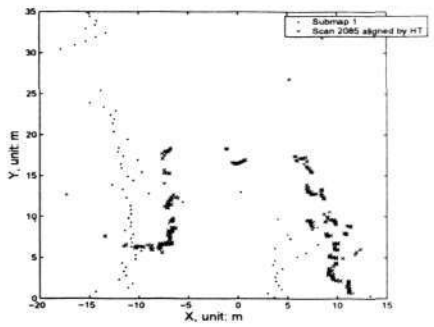


Figure 3.41: The alignment result based on the directions calculated from HT.

However, it should be noted that the Hough Transform in this work is not a feature extraction algorithm, its objective is not to detect any lines in the scan or submap, but rather, to retrieve the most likely heading direction of the distribution.

3.6.9.2 To calculate translation by first order moment

Although HT can handle the rotation of viewpoint, it may not cope well with translations in the x-axis. Therefore, a moment based alignment preprocessing similar to [89] is further introduced. The n'th raw moment $mu_n^{'}$ (i.e., moment about zero) of a distribution $P(x)$ is defined by

$$mu_n^{'} = \langle x^n \rangle \tag{3.37}$$

where

$$\langle f(x) \rangle = \sum f(x) P(x) \tag{3.38}$$

A range scan can be regarded as a distribution of 2-D vectors:

$$s_a = (b^1, b^2, \cdots, b^{361}) = \begin{pmatrix} x^1 & x^2 & \cdots & x^{361} \\ y^2 & y^2 & \cdots & y^{361} \end{pmatrix} \tag{3.39}$$

The first order moment of this 2-D distribution is defined as:

$$\mu_1(s_a) = \frac{\sum_{i=1}^{361} b^i}{361} \tag{3.40}$$

In the same manner, if we regard a submap as a collection of 2D scans, it can also be formed as a distribution of 2-D vectors, similar to the one in (3.39).

By comparing the x value of the first moment of the distribution of a scan and a submap, we can align them in the x-axis. Together with the rotation calculated from Hough Transform, the scan No. 2085 and submap No.1 can be accurately aligned, as demonstrated in Fig. 3.42.

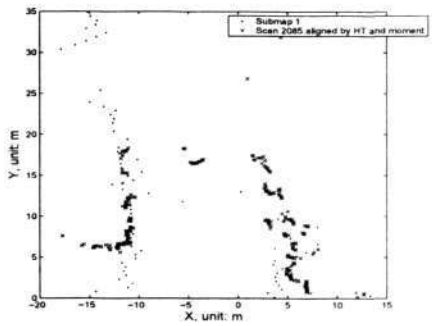


Figure 3.42: The first order moment is used to further align the result from Hough Transform in Fig. 3.41. *It can be observed in this figure, at the position of loop-closure, by using Hough Transform and Moment based technique, a scan from a new viewpoint can be accurately aligned with the previously observed submap.*

3.6.9.3 Results

The scan No. 2085 is observed at the beginning of submap No.1, as illustrated in Fig.3.43.

This scan is very similar to submap No.1. It is observed when the vehicle is turning, i.e., from a different viewpoint. Consequently, its observational probability in Submap No.1 is quite low. With the alignment preprocessing, the observational

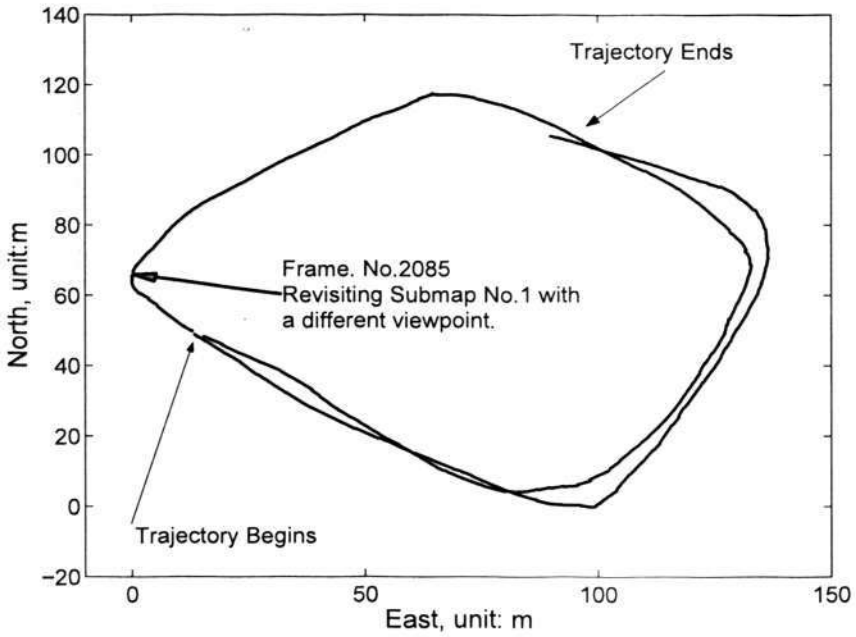


Figure 3.43: The scan frame taken at the position when the vehicle re-visited submap No. 1. This time it entered from a turning road so that the viewpoint is considerably different from the first time.

probability in Submap No.1 is increased. This result demonstrates the feasibility of using HT/moment alignment to loose the viewpoint invariance constraint.

3.6.9.4 Discussion on viewpoint invariance

Fortunately, for a cyclic loop-closure, the viewpoint variances are actually not a problem. As observed in Figure 3.17, when the vehicle visits and re-visits submaps No. 1,2,...,5, its trajectories are not exactly same. This demonstrates the proposed appearance model's capability to handle moderate viewpoint invariance.

It should be noted that, the existence of cyclic environment is never an over-restricted assumption. In most large-scale mapping scenarios such as [109, 38], the involved testing platforms are vehicles whose motion (especially steering) is constrained, then they generally cannot move to off-road terrain. Therefore, for an

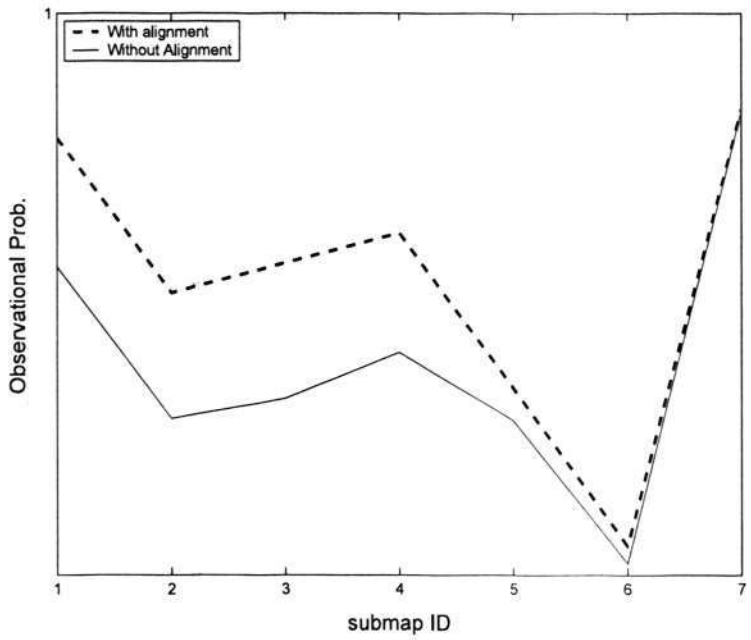


Figure 3.44: In the situation of viewpoint variance, by aligning the scan according to the submap's deployment, the observational probability can be increased.

application in cyclic environment, the vehicle's trajectories, as well as viewpoints, are constrained by nature.

It should also be noted that, the viewpoint invariance is still an outstanding problem either in computer vision domain or in robotics domain, which deserves further investigations.

3.7 Conclusion and Discussion

This chapter explains an innovative approach to detect the loop-closure for SLAM in highly unstructured cross-country environment where no geometrical features are available. It presents how to conduct PCA to model the environment's appearance (not the environment itself) using 2D range scans. After the high dimensional measurements are projected into the low dimensional map, their distribu-

tions are approximated by a series of Gaussian models. These Gaussian models are used to calculate the observation probability for Bayesian inference. By doing so, the appearance model of the environment could be integrated into the topological Bayesian inference process, so that the metrical level feature information is no required. The experimental results demonstrate that the proposed Eigenmap technique can robustly detect loop-closure in cross country environment where no geometrical features are available.

Different from the conventional loop-closure detection techniques which rely on feature tracking, in this inference framework, no vehicle pose prediction is necessary. In other words, it is no longer required that the vehicle's global pose or the features' global position is accurately estimated. So even the vehicle's localization error is huge, the detection algorithm can work properly. This is a highly desirable characteristic for a tracked vehicle moving the cross-country environment, because in this case, the error of the vehicle's self-localization could be quite big after a short distance. With such a huge error, it is impossible to use any 'gating' mechanism to search whether the vehicle has re-visited a certain place.

Chapter 4

Appearance-based loop-closing optimization for map hierarchy

4.1 Introduction

After the detection of a loop-closure, the robot associates its current observation with the map it obtained sometime ago. Due to all kinds of errors in the SLAM process, at the point of loop-closure, there is always an inconsistency between the map and the observation. Accordingly, there need to be an optimization process to propagate the loop-closing error backward to all the map items to obtain a consistent map. In this context, to distinguish it from loop-closure detection as discussed previously, this process is referred to as 'loop-closing optimization'. This chapter will present an innovative loop-closing optimization technique. This technique is based on the same hierarchical framework as discussed in Chapter 3.

Dissanayake proved a very important theorem [20] in 2001. Given enough observations, if all the sensor errors are white Gaussian, the uncertainty of any relation between map items will finally converge to zero. In other words, the map will become fully correlated in the limit, therefore, it can be regarded as a rigid

body.

Going through the same environment again and again until the map converges may be very time-consuming for one single robot in the large-scale outdoor navigation tasks. This has motivated the development of collaborative mapping using multiple robots, as in [53]. However, in a lot of unmanned surveillance/reconnaissance scenarios, it is often desirable to reduce the operational time and number of robots whenever possible.

On the other hand, even a converged map can be obtained, it only provides the relative locations among map items. Such relative map may not be able to support complex outdoor navigation tasks, such as surface-air collaborations, human-robot interactions, multi-sensor multi-platform tracking. In these applications, exchanges of information are often preferred to be performed in a global coordinate system. It is often desirable to accurately register SLAM's mapping result to an existing roadmap.

In this chapter, the loop-closing is modeled as a problem of registering the mapping result to *a priori* appearance information (e.g. a roadmap) in a global coordinate system. This registration is thereafter solved by an optimization algorithm. With the help from roadmap information, the presented algorithm can accurately model large environment with one single robot, by visiting this environment for only once. In the meanwhile, the resulting map provides comprehensive 3D description about the environment in a predefined global coordinate.

The mapping result is modeled as a graph of local submaps arranged in a deformable configuration. The initial configuration is incrementally constructed by

filtering local perceptual data and motion readings till a loop is closed. This initialization process is very similar to other submap-based SLAM algorithms [90][28].

After initialization, global appearance information is introduced as a constraint on the loop-closing optimization process. Appearance information can be obtained from a roadmap. It describes how the mapped environment should look like in a global coordinate system. In the loop-closing process, the appearance information attracts submaps to a configuration that best matches the input image. The process of loop-closing with appearance information (shorted as LCAI in this context) is illustrated in Figure 4.1. The initial submap configuration Φ is built by conventional SLAM techniques. This configuration is deformed according to the appearance information I, to obtain a consistent mapping result.

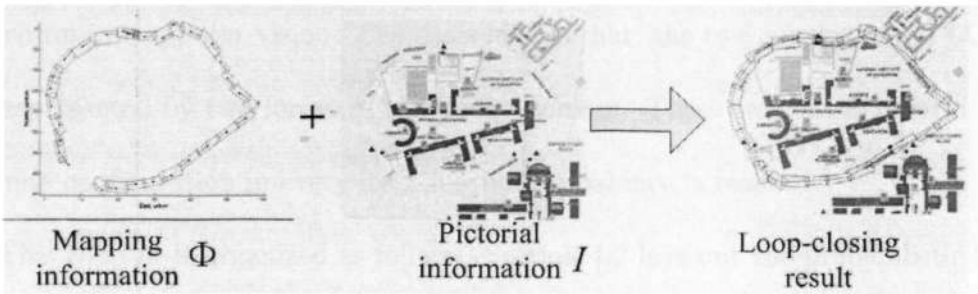


Figure 4.1: The loop-closing with appearance information.

In this work, the LCAI is formulated in a probabilistic form, which is consistent with other generic stochastic SLAM algorithms. The objective of LCAI is to obtain a *maximum a posteriori* (MAP) estimate for the submap configuration, based on both mapping information and pictorial information:

$$\mathcal{X}^* = \arg\max_{\mathcal{X}}(\mathcal{X}|\Phi, I) \tag{4.1}$$

Since the pictorial information I is provided by an image without an analytical

formulation, it is difficult to derive a closed-form solution to match the submap configuration to the image. Another contribution of this work is to develop a two stage optimization algorithm to guide the submap configuration gradually to its desired position. The first stage is a global searching process based on genetic algorithm (GA). The submap configuration space is explored by a set of chromosomes, while each of them encodes a submap configuration. A fitness value is calculated for each gene based on (4.1), so that erroneous submap configurations can be exterminated during the evolution. The evolution stops when a rough configuration is achieved. Using the result of GA as initial configuration, in the second stage, the MAP estimation is solved by an iterative energy minimization process. This technique is inspired by the active contour (also called SNAKE) [49] algorithm in computer vision. The basis idea is that, the two constraints in LCAI are represented by two forces in the image domain. These two forces deform the submap configuration in every iteration until a balance is reached.

This chapter is organized as follows. Section 4.2 lays out the probabilistic formulation of loop-closing with appearance information; thereafter, a genetic algorithm based loop-closing optimization strategy is explained in Section 4.3; Section 4.4 introduces an iterative optimization algorithm that further tunes the result of GA; finally, the experimental results and performance analysis are given in Section 4.5.

4.2 Probabilistic loop-closing with appearance information

The major contribution of this work is to introduce a global appearance information to the loop-closing process, as an additional constraint on the deformable submap configuration. The mapping process is organized as a hierarchy, the lower level mapping fues local sensor data to build the submaps; the higher level mapping is the LCAI which is performed only when the robot re-visits a place and tries to close a loop.

4.2.1 Submap configuration

As elaborated in [28], when the robot navigates and segments the whole environmental map, it can simultaneously estimate the submaps, the transformations between submaps and the uncertainties of these transformations. In this context, they are referred to as *initial submap configuration*, or shorted as *mapping information*. Mathematically, at the time of loop-closing l, given the history of perceptual data Z_l, and interior motion readings U_l, the initial submap configuration Φ can be constructed from a SLAM routing Λ:

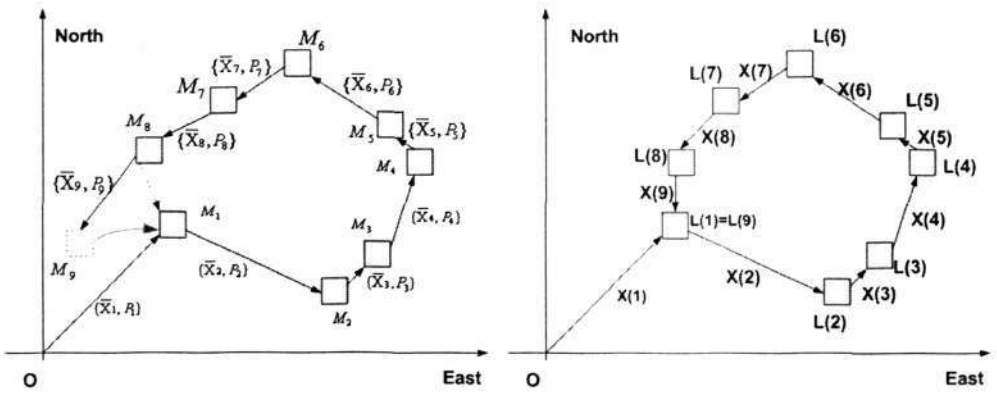
$$\Phi = \{\bar{\mathcal{X}}, \mathcal{P}, \mathcal{M}\} = \Lambda(Z_l, U_l) \tag{4.2}$$

where Φ comprises submap transformations $\bar{\mathcal{X}}$, transformation covariances $\mathcal{P}$, and submaps themselves $\mathcal{M}$:

$$\bar{\mathcal{X}} = \begin{pmatrix} \vdots \\ \bar{\mathrm{x}}_s \\ \vdots \end{pmatrix} \qquad \mathcal{P} = \begin{pmatrix} \ddots & 0 & 0 \\ 0 & \mathrm{P}_s & 0 \\ 0 & 0 & \ddots \end{pmatrix} \qquad \mathcal{M} = \begin{pmatrix} \vdots \\ \mathrm{M}_s \\ \vdots \end{pmatrix}$$

here $s = 1, 2, \cdots, N_m$ and N_m is the total number of submaps.

A typical loop-closing situation can be illustrated by Figure 4.2. The deployment of all the 9 submaps is defined by the sequence of coordinate transformations between submaps, which is denoted as $\bar{\mathcal{X}}$. When a loop-closing situation is confirmed, the submap M_9 should be shifted to the position of submap M_1, as indicated. It can be observed that, the deployment of all the 8 submaps can be defined by either the sequence of *relative* transformations $\mathcal{X}$, or equivalently, the sequence of *absolute* global positions $\mathcal{L}$.



(a) Initial submap configuration (b) *a posteriori* submap configuration

Figure 4.2: (a) The initial submap configuration. (b) The *a posteriori* submap configuration corresponding to (a).

In this framework, each submap M_s is represented as a 3D point cloud $\{x_{s,j}, y_{s,j}, z_{s,j}\}$ while $j = 1, 2, \cdots, N_s$, here N_s is the number of points inside submap s.

$$\mathrm{M}_s = \begin{pmatrix} x_{s,1} & y_{s,1} & z_{s,1} \\ x_{s,2} & y_{s,2} & z_{s,2} \\ \vdots & \vdots & \vdots \\ x_{s,N_s} & y_{s,N_s} & z_{s,N_s} \end{pmatrix} \tag{4.3}$$

The 3D submaps are built from laser range measurements in this work. However,

it should be noted that the proposed algorithm is not restricted to any specific sensor. Actually, these point clouds could be obtained through a variety of sensors and techniques, e.g., 2D range scanner, 3D radar, stereovision and, structure from motion (SFM).

These submaps and their deployments encode the information collected during the mapping process. Based on this information, the *a posteriori* map configuration $\mathcal{X}$ can be estimated. In the mapping context, the target variable to be estimated is the transformation between adjacent submaps, accordingly, $\mathcal{X}$, and all its inside items, are marked with bar.

As depicted in Figure 4.2, at the time when a loop-closure is detected, this configuration resembles a circle, while each arc $s-1 \rightarrow s$ represents a known transformation from submap M_{s-1} to M_s. This arc is denoted as $\bar{\mathrm{x}}_s$, which is a 3-vector. These three items represent the translation of a submap relative to the coordinate of its adjacent submap:

$$\bar{\mathrm{x}}_s = \begin{pmatrix} \Delta\bar{x}_s & \Delta\bar{y}_s & \Delta\bar{\gamma}_s \end{pmatrix} \tag{4.4}$$

The estimated transformations between submaps come with uncertainties. These uncertainties are approximated to be Gaussian, and represented by a covariance matrix $\mathcal{P}$. It can be noticed that a submap M_s is independent of any previous ones, because it is built relative to the vehicle location at the moment it was built. Consequently, any two submaps within $\mathcal{M}$ are statistically uncorrelated. Since P_s is regarded as a fixed parameter, rather than a variable to be estimated, it is not included below the conditioning bar as $\bar{\mathrm{x}}_s$ is.

Similar to the definition of $\bar{\mathcal{X}}$, the *a posteriori* map is denoted as the collection of transformations between local submaps:

$$\mathcal{X} = \begin{pmatrix} \cdots & \mathrm{x}(s) & \cdots \end{pmatrix}^T \tag{4.5}$$

Here the symbol $(\cdot)$ rather than lower-type is used to emphasize that, the final submap graph is not static, but modeled as a function of submaps.

Similarly, the pose of submap s is denoted as $\mathrm{L}(s)$, which comprises the submap's origin $\{\mathrm{n}(s), \mathrm{e}(s)\}$ and direction $\tau(s)$ in the fixed north-east global coordinate frame:

$$\mathrm{L}(s) = \begin{pmatrix} \mathrm{n}(s) & \mathrm{e}(s) & \tau(s) \end{pmatrix}^T \tag{4.6}$$

Equivalent to (4.5), the mapping information can also be formulated in an absolute form:

$$\mathcal{X} \equiv \mathcal{L} = \begin{pmatrix} \cdots & \mathrm{L}(s) & \cdots \end{pmatrix}^T \tag{4.7}$$

A pair of operators $\oplus$ and $\ominus$ are introduced here to represent the coordinate transformations in the Euclidean space. Accordingly, by defining $\mathrm{L}(0)$ as the origin of the fixed global coordinate, there exist following relations between submap locations L and submap transformations x:

$$\mathrm{L}(s) = \mathrm{L}(0) \oplus \mathrm{x}(1) \oplus \mathrm{x}(2) \oplus \mathrm{x}(3) \ldots \oplus \mathrm{x}(s) \tag{4.8}$$

and respectively, the pose of submap s with respect to $s-1$ is calculated as:

$$\begin{aligned} \mathrm{x}(s) &= \ominus \mathrm{L}(s-1) \oplus \mathrm{L}(s) \qquad (4.9) \\ &= \begin{pmatrix} \big(e(s) - e(s-1)\big)\cos\tau(s-1) + \big(n(s) - n(s-1)\big)\sin\tau(s-1) \\ -\big(e(s) - e(s-1)\big)\sin\tau(s-1) + \big(n(s) - n(s-1)\big)\cos\tau(s-1) \\ \tau(s) - \tau(s-1) \end{pmatrix} \end{aligned}$$

4.2.2 Probabilistic loop-closing with appearance information

Based on above mapping information, the task of LCAI is to calculate the *a posteriori* probability density function (PDF) of the submap configuration $\mathcal{X}$. Taken the mapping process denoted in (4.2), the target probability to be estimated is:

$$p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{M}, \mathcal{P}, I) \tag{4.10}$$

According to the Bayes rule, the target probability in (4.10) is marginalized as:

$$p(\mathcal{X}|\bar{\mathcal{X}}, I, \mathcal{M}, \mathcal{P}) \propto p(I|\mathcal{X}, \bar{\mathcal{X}}, \mathcal{M}, \mathcal{P}) p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{M}, \mathcal{P}) \tag{4.11}$$

The image appearance is only determined by the status of the real world, rather than how it appears in the robot's sensor observation. For this reason, the $\bar{\mathcal{X}}$ in the first component on the right-side of (4.11) can be omitted:

$$p(I|\mathcal{X}, \bar{\mathcal{X}}, \mathcal{M}, \mathcal{P}) \propto p(I|\mathcal{X}, \mathcal{M}, \mathcal{P}) \tag{4.12}$$

By substituting (4.12) back to (4.11), the target probability in (4.10) is reformulated as:

$$p(\mathcal{X}|\bar{\mathcal{X}}, I, \mathcal{M}, \mathcal{P}) \propto p(I|\mathcal{X}, \mathcal{M}, \mathcal{P}) p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{M}, \mathcal{P}) \tag{4.13}$$

Since observing I is only dependent on the layout of the submaps, the $\mathcal{P}$ in $p(I|\mathcal{X}, \mathcal{M}, \mathcal{P})$ on the right side can be removed. Similarly, the prior probability of $\mathcal{X}$ is independent of the details of submaps, so the $\mathcal{M}$ in the second item $p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{M}, \mathcal{P})$ is omitted too. The target probability is thus refined as:

$$p(\mathcal{X}|\bar{\mathcal{X}}, I, \mathcal{M}, \mathcal{P}) \propto p(I|\mathcal{X}, \mathcal{M}) p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{P}) \tag{4.14}$$

The probability $p(I|\mathcal{X}, \mathcal{M})$ captures the imaging process, and measures the likelihood of observing appearance information I given the map configuration. It is not difficult to understand that, one submap's appearance will not affect the appearances of others. So this probability can be decomposed into:

$$p(I|\mathcal{X}, \mathcal{M}) \propto \prod_{s=1}^{N_m} p(I|\mathrm{L}_s, \mathrm{M}_s) \tag{4.15}$$

where each $p(I|\mathrm{L}_s, \mathrm{M}_s)$ characterizes the match between submap s and the image.

The distribution $p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{P})$ models the conventional loop-closing optimization process. Its constraint on the whole submap graph can be decomposed into constraints on each individual transformation:

$$\begin{aligned} p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{P}) &= \prod_{s=1}^{N_m} p(\mathrm{x}(s)|\bar{\mathrm{x}}_s, P_s) \\ &\propto \prod_{s=1}^{N_m} \exp\Big(-\frac{1}{2}(\mathrm{x}(s) - \bar{\mathrm{x}}_s)^T P_s^{-1} (\mathrm{x}(s) - \bar{\mathrm{x}}_s)\Big) \end{aligned} \tag{4.16}$$

By substituting (4.15) and (4.16) into (4.14), the final formulation of the posterior probability distribution of the submap graph $\mathcal{X}$ is:

$$\begin{aligned} &p(\mathcal{X}|I, \bar{\mathcal{X}}, \mathcal{M}, \mathcal{P}) \\ &\propto \prod_{s=1}^{N_m} p(I|\mathrm{L}(s), \mathrm{M}_s) \prod_{s=1}^{N_m} \exp\Big(-\frac{1}{2}(\mathrm{x}(s) - \bar{\mathrm{x}}_s)^T P_s^{-1} (\mathrm{x}(s) - \bar{\mathrm{x}}_s)\Big) \end{aligned} \tag{4.17}$$

Taken the objective of LCAI as formulated in (4.1), the MAP loop-closing estimation is detailed as:

$$\begin{aligned} \mathcal{X}^* = \arg\max_{\mathcal{X}} \quad & \prod_{s=1}^{N_m} p(I|\mathrm{L}(s), \mathrm{M}_s) \times \\ & \prod_{s=1}^{N_m} \exp\Big(-\frac{1}{2}(\mathrm{x}(s) - \bar{\mathrm{x}}_s)^T P_s^{-1} (\mathrm{x}(s) - \bar{\mathrm{x}}_s)\Big) \end{aligned} \tag{4.18}$$

4.2.3 An energy-minimization formulation

Taking the negative logarithm of the right side of (4.18) yields:

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} \sum_{s=1}^{N_m} \left(f\left(\mathrm{L}(s), \mathrm{M}_s\right) + \epsilon \times \left(\mathrm{x}(s) - \bar{\mathrm{x}}_s\right)^T P_s^{-1} \left(\mathrm{x}(s) - \bar{\mathrm{x}}_s\right) \right) \quad (4.19)$$

where $f\left(\mathrm{L}(s), \mathrm{M}_s\right) = -\log p(I|\mathrm{L}(s), \mathrm{M}_s)$ is a match cost measuring how well submap M_s matches the image I at the pose $\mathrm{L}(s)$. The symbol ϵ is a weight determining how much the mapping information and appearance information contribute in the minimization process. Therefore, the objective of LCAI is to minimize the target energy function E^{LC}:

$$E^{LC}(\mathcal{L}) = E^{Ext}(\mathcal{L}) + \epsilon \times E^{M}(\mathcal{L}) \quad (4.20)$$

where

$$E^{Ext}(\mathcal{L}) = \sum_{s=1}^{N_m} f\left(\mathrm{L}(s), \mathrm{M}_s\right) \quad (4.21)$$

and

$$E^{M}(\mathcal{L}) = \sum_{s=1}^{N_m} \left(\ominus \mathrm{L}(s-1) \oplus \mathrm{L}(s) - \bar{\mathrm{x}}_s \right)^T P_s^{-1} \left(\ominus \mathrm{L}(s-1) \oplus \mathrm{L}(s) - \bar{\mathrm{x}}_s \right) \quad (4.22)$$

In this energy model, $E^{M}(\mathcal{L})$ characterizes the cost of deforming from the stochastic submap structure built from (4.2). Minimizing $E^{M}(\mathcal{L})$ is essentially trying to reach a Maximum Likelihood estimate for the submap configuration based solely on initial mapping information, while $E^{Ext}(\mathcal{L})$ gives penalty to $\mathcal{L}$ which does not match the appearance information.

The distribution of image pixels' grey values is difficult to be modeled analytically, which leads to a high dimensional, non-convex energy function. Consequently, it is very difficult, if not impossible, to reach a close-form solution for

LCAI in (4.29). A lot of literature has been published addressing such optimization problem, they generally fall into two categories. Global optimization algorithms, e.g., genetic algorithm [22], try to find a solution in the solution space for which the objective function (4.29) obtains its global minimum. On the other hand, greedy algorithms, such as ICM [5], follow the problem solving meta-heuristic of making the locally optimum choice at each stage with the hope of finding the global optimum. They do not consistently find the globally optimal solution, because they usually do not operate exhaustively on all the data. For this reason, greedy algorithms can, in principle, achieve higher efficiency than global techniques.

In this work, the global search and greedy search are integrated into a two-stage optimization process to minimize the energy in (4.29). The first stage is a genetic algorithm process that terminates when a rough estimate is obtained. Thereafter, a more efficient gradient search is performed to refine the result of GA. By employing this two-stage optimization process, the advantages of both methods can be exploited: the presented technique can achieve both robustness against local minimum and efficiency.

4.3 Global energy minimization using genetic algorithm

4.3.1 Unconstrained loop-closing optimization

Before loop-closure is detected, say, at time -1, the last component in the submap configuration is:

$$\mathrm{L}(m,-1) = \bar{\mathrm{L}}(0) \oplus \bar{\mathrm{x}}_1 \oplus \bar{\mathrm{x}}_2 \ldots \oplus \bar{\mathrm{x}}_m. \tag{4.23}$$

When a loop-closure is detected at time 0, the last submap is associated with submap 1, so the new pose of the last submap will be:

$$\mathrm{L}(m,0) = \bar{\mathrm{L}}(0) \oplus \bar{\mathrm{x}}_1. \tag{4.24}$$

Apparently, due to the existence of linearization error and Gaussian approximation, $\mathrm{L}(m,-1)$ will not be equal to $\mathrm{L}(m,0)$. The difference between them is often referred to as loop-closing error. This loop-closing error can introduce a big energy to the system:

$$E^M\Big(\mathrm{x}(m,0)\Big) = \Big(\mathrm{L}(m,0) \ominus \mathrm{L}(m-1,0) - \bar{\mathrm{x}}_m\Big)^T P_m^{-1} \Big(\mathrm{L}(m,0) \ominus \mathrm{L}(m-1,0) - \bar{\mathrm{x}}_m\Big). \tag{4.25}$$

The loop-closing optimization's objective is thus to re-distribute above error to each component in the map and try to achieve a minimum energy of 4.20 with an additional loop-closing constraint:

$$\mathrm{L}(m) = \mathrm{L}(1) \tag{4.26}$$

Taken the coordinate transformation defined in 4.8, this constraint is equivalent to:

$$\mathrm{L}(m) = \mathrm{L}(1) \oplus \mathrm{x}(2) \oplus \mathrm{x}(3) \oplus \mathrm{x}(4) \ldots \oplus \mathrm{x}(m). \tag{4.27}$$

This formulation essentially embodies a more general case of the constrained optimization problem in the work of Estrada [28]. This constrained problem is turned to an unconstrained one here by replacing (4.26) with a weighted penalty function:

$$\mathcal{C}(\mathcal{X}) = w\|\mathrm{L}(0) \oplus \mathrm{x}(1) - \mathrm{L}(0) \oplus \mathrm{x}(1) \oplus \mathrm{x}(1) \cdots \oplus \mathrm{x}(m)\|, \tag{4.28}$$

when the weight w approaches infinity, the solution of the unconstrained problem converges to the solution to the constrained one.

So the objective energy function to be minimized is:

$$E(\mathcal{X}) = E^M(\mathcal{X}) + E^{LC}(\mathcal{X}) + \mathcal{C}(\mathcal{X}) \tag{4.29}$$

which is a complex function with potential local minima over the submap transformation space.

4.3.2 Introduction to genetic algorithm

Genetic Algorithm (GA) is an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomized, GA is by no means random, instead it exploits historical information to direct the search into the region of better performance within the search space. GA is designed to simulate processes in natural systems necessary for evolution, which follow the principles first laid down by Charles Darwin of 'survival of the fittest'. Since in nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.

At each generation k, the GA maintains a population of N_c chromosomes:

$$\mathbb{POP}(k) = \left(\mathcal{X}^1(k), \mathcal{X}^2(k), \mathcal{X}^3(k), \cdots, \mathcal{X}^{N_c}(k)\right) \tag{4.30}$$

GA simulates the survival of the fittest among individuals over consecutive generation for solving a problem. Each generation consists of a population N_c of character strings that are analogous to the chromosome that we see in our DNA. Each indi-

vidual represents a point in the submap configuration space and a possible submap deployment:

$$\begin{aligned}\mathcal{X}^i &= \Big(\mathrm{L}^i(0), \mathrm{x}^i(1), \ldots, \mathrm{x}^i(m)\Big)^T \\ &= \Big(\underbrace{n_0^i, e_0^i, \tau_0^i}, \underbrace{\psi_1^i, \phi_1^i, \gamma_1^i}, \cdots, \underbrace{\psi_m^i, \phi_m^i, \gamma_m^i}\Big)^T\end{aligned}$$

The individuals in the population are then made to go through a process of evolution. A fitness measure $\mathcal{F}$ can be easily derived from the energy function in (4.29):

$$\mathcal{F}(\mathcal{X}^i) = \exp\Big(-E(\mathcal{X}^i)\Big) \tag{4.31}$$

After evaluating each chromosome, a new generation can be established by selecting chromosomes from the old ones in a probabilistic manner. The likelihood that a chromosome should be selected is:

$$p(\mathcal{X}_k^i) = \frac{\mathcal{F}(\mathcal{X}_k^i)}{\sum_{j=1}^{N_c} \mathcal{F}(\mathcal{X}_k^j)} \tag{4.32}$$

The pair of chromosomes (called parents) are selected at random and the single-point crossover operator is applied according to a fixed crossover rate p_c. For this operation, a random number in the range of 0 the length L_q of the string is generated. This is called the crossover point. the portions of the two strings lying to the right of the crossover point are interchanged to yield two new strings (called offspring).

The mutation operation is introduced to prevent premature convergence to local optima by randomly sampling new points in the search space. In a mutation event, some mutation points are randomly selected based on a mutation likelihood p_m, as in Figure 4.4. The gene at each mutation point is changed, so that the

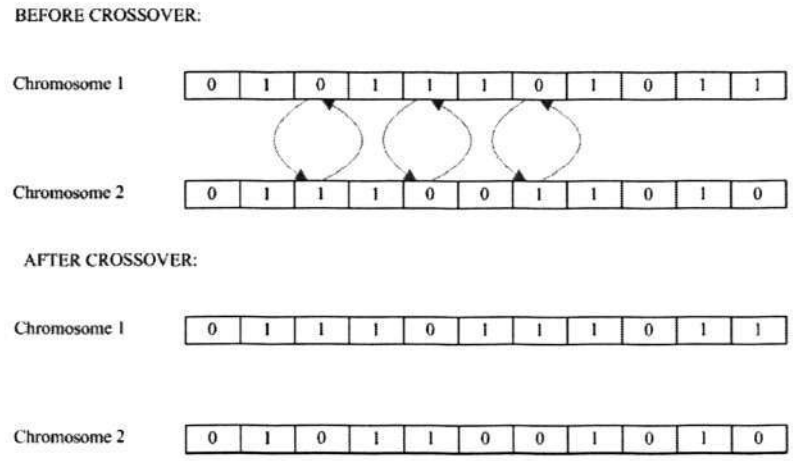


Figure 4.3: The crossover procedure.

chromosome can become different. Crossover and mutation are the two methods

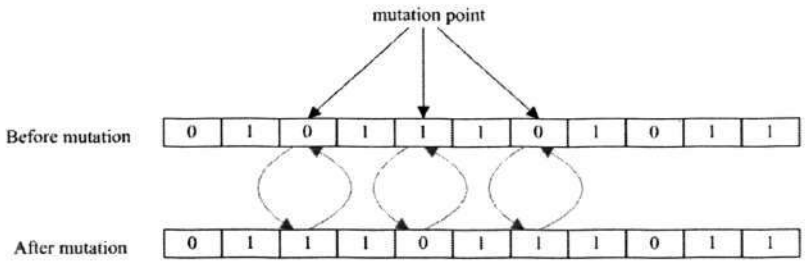


Figure 4.4: The mutation procedure.

that the GA explores new submap configurations and evolve the population toward higher fitness. Theoretically, higher mutation rate and higher crossover rate can prevent premature convergence, i.e., the situation that the population stops evolution before it reaches the desirable configuration. However, GA may take a long time to process when p_m and p_c are set too high.

Although GA can search the whole configuration space, it cannot guarantee a global optimal solution unless the problem itself is convex. In this work, GA is employed only to prevent the gradient search algorithm trapped by a local minimum.

4.3.3 Encoding submap configuration by chromosome

As explained in the previous section, genetic algorithm maintains a population of chromosomes, while each chromosome encodes a submap configuration $\mathcal{X}^i$. As explained in Section 4.2.1, each $\mathcal{X}^i$ represents a possible deployment of all the submaps as in Fig.4.2.

$$\begin{aligned}\mathcal{X}^i &= \left(\mathrm{L}^i(0), \mathrm{x}^i(1), \ldots, \mathrm{x}^i(m)\right)^T \\ &= \left(\underbrace{n_0^i, e_0^i, \tau_0^i}, \underbrace{\psi_1^i, \phi_1^i, \gamma_1^i}, \cdots, \underbrace{\psi_m^i, \phi_m^i, \gamma_m^i}\right)^T\end{aligned}$$

The most popular way of chromosome encoding is to use binary code, i.e., each item in Eq.4.33 is turn to binary form. However, it is inconvenient to encode float number, here all the items in Eq.4.33 are decoded to unsigned decimal integers. For position data as $n_0^i, e_0^i,$, the resolution will at the pixel level. For angle, the unit will be degree. Since GA's objective in this work is to provide a rough estimate for further refined searching performed by SNAKE algorithm, such resolution is already sufficient.

The roadmap image's resolution is 590 $\times$ 504 in this work. For this reason, $n_0^i, \psi_1^i, \psi_2^i, ..., \psi_m^i$ should not be bigger than 590. Since $2^9 < 590 < 2^{10}$, a 10 digits binary string should be sufficient to encode each item in $n_0^i, \psi_1^i, \psi_2^i, ..., \psi_m^i$. In the same manner, people can use a 9-digit binary code to encode $e_0^i, \phi_1^i, \phi_2^i, ..., \phi_m^i$. For the angular values, since $256 < 360 < 512$, a 9-digit binary code should be sufficient.

Therefor, to represent a submap configuration as in Fig.4.2, GA needs a binary string at the length of $9 \times (10 + 9 + 9) = 252$.

4.3.4 Appearance matching

The likelihood $f\Big(\mathrm{L}(s), \mathrm{M}_s\Big) = -\log p(I|\mathrm{L}(s), \mathrm{M}_s)$ is the probability of observing image I given that the submap s is at the pose $\mathrm{L}(s)$ with 3D structure described by M_s. Here, the submap graph can be regarded as a generative model from which the target environment's appearance in image can be inferred. To calculate this likelihood, this generative model must be projected into the image plane of the camera, or equivalently, the plane where the appearance information exists. The projection of local 3D submaps into the image plane is computationally expensive, taken the huge number of points that exist in the 3D submap model. Since this likelihood calculation is performed every time GA evaluates the fitness of a chromosome, an efficient approximation for above projection is necessary.

The proposed algorithm assumes that the appearance prior is often acquired by a roadmap, so that the captured image resembles a top-view and the submaps' local appearance will not be affected by their relative pose in the terrain plane. Therefore, these local appearance can be calculated for only once and then stored for all the following processing.

Here the camera model is denoted as $\mathcal{C}$. With this model, submap $\{\mathrm{L}(s), \mathrm{M}_s\}$ can be represented in the image domain as its projection corrupted by noise η:

$$\hat{\mathrm{I}}(s) = \mathcal{C}(\mathrm{M}_s, \mathrm{L}(s)) + \eta = \sum_{j \in \mathrm{M}_s} C(x_{s,j}, y_{s,j}, z_{s,j}) + \eta_{s,j} \tag{4.33}$$

Due to occlusions, shadowing, and those objects whose heights are below the vertical laser scanner, there may exist outliers or matched pixels in the rendered image. In the computer vision domain, modeling such outliers has long been an

outstanding problem. A few algorithms have been developed for visual surveillance research[60] [18]. However, a limitation in this work is that, there is only one image for building the reference image model. Consequently, it is impossible to build a multiple model as in [60]. In this work, a simpler uni-model representation is employed, similar to the work in [29]. The noise η is taken to be a mixture of a Gaussian and a uniform distribution:

$$p(\eta_{s,j}) = (1-\varepsilon)\mathcal{N}(0,\sigma_p) + \varepsilon/256 \tag{4.34}$$

where $0 \leq \varepsilon \leq 1$. The uniform noise is bounded over a finite interval of intensity values while $\mathcal{N}(\cdot)$ is zero-mean normal distribution whose variance may change with spatial position. In general, the variance σ_p is sufficiently small that the area of the Gaussian outlier the bounded interval may be ignored. Similarly, the whole submap configuration in the image domain can be calculated as:

$$\hat{\mathrm{I}}(\mathcal{L}) = \sum_{s\in[1,N_s]} \hat{\mathrm{I}}(s) \tag{4.35}$$

Given the generative submap graph, the likelihood of each submap s is defined independently. The algorithm samples, with replacement, $i = 1, 2, ...N_p$ pixel locations $\{u_{s,i}, v_{s,i}\}$ uniformly from the projected region of submap s, i.e., $\hat{\mathrm{I}(s)}$. The gray value differences between points on the appearance model and the corresponding submap projection are independent and are modeled as a mixture of a zero mean normal distribution and a uniform outlier distribution.

The appearance matching likelihood of submap s is then expressed as:

$$f\Big(\mathrm{L}(s), \mathrm{M}_s\Big) = \frac{\varepsilon}{256} + \frac{1-\varepsilon}{\sqrt{2\pi}\sigma} \exp\left(-\sum_{i=1}^{N_p} \frac{(\mathrm{I}(u_{s,i}, v_{s,i}) - \hat{\mathrm{I}}(u_{s,i}, v_{s,i}))^2}{2\sigma_p^2}\right) \tag{4.36}$$

4.4 SNAKE-based local energy minimization

In this work, a greedy searching strategy inspired by active contour (SNAKE) [49] is developed to refine the result of GA. Active contour is essentially a gradient greedy shape fitting algorithm, which tries to minimize an energy function derived from the image so that it takes on its smaller values at the features of interest. In the scenario of SLAM, edges are natural features to be exploited, because the edges often represent structures in the environment, e.g., roads and buildings.

The basic idea is to sequentially adjust the pose for each single submap based on the available local appearance information and mapping information, and iterate this procedure until convergence. A potential field is constructed in the domain of the input roadmap image. The internal force computed from the initial submap configuration tries to keep the submap graph in the configuration as it is constructed, i.e., impose consistences between each pair of consecutive submaps. Simultaneously, the constraint from appearance information is modeled as forces which guide submaps to a globally consistent configuration, where they match the input image best.

With respect to the LCAI in (4.20), the configuration of submaps that minimizes E^{LC} must satisfy the Euler equation:

$$\nabla E^{Ext}(\mathcal{L}) - \epsilon \times \nabla E^{M}(\mathcal{L}) = 0 \tag{4.37}$$

To find a solution for (4.37), submap configuration is made dynamic by treating $\mathcal{X}$ as a function of iteration t, as well as s. Then the configuration at iteration t is denoted as $\mathcal{X}^{(t)}$, or equivalently, $\mathcal{L}^{(t)}$, and the poses of submaps within it

are consistently written as $\mathrm{L}^{(t)}(s)$. Here time '-1' is defined as the moment right before loop-closure is detected. Correspondingly, time '0' is the moment when loop-closure is confirmed, i.e., submap N_m is replaced with submap 1. Thereafter, $t = 1, 2, 3, ...$ represents the index of iteration.

Taken the Euler function in (4.37), the submap configuration can be iteratively updated as:

$$\mathcal{L}^{(t+1)} = \mathcal{L}^{(t)} + \underbrace{\nabla E^{Ext}\big(\mathcal{L}^{(t)}\big)}_{\mathrm{F}^{Ext}} + \underbrace{\big(-\epsilon \times \nabla E^{M}(\mathcal{L}^{(t)})\big)}_{\mathrm{F}^{M}} \tag{4.38}$$

This is essentially a force balance process, in which these two forces push/attract the submap configuration $\mathcal{L}$ to the desired position.

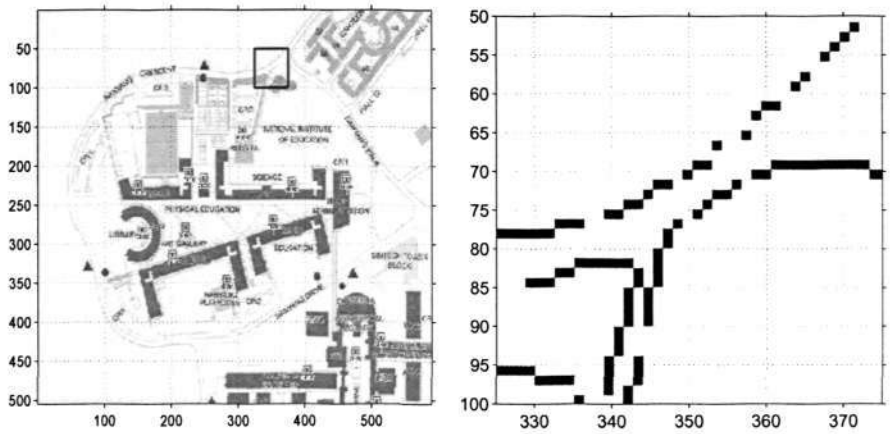
4.4.1 Constraint from appearance information

The image is denoted as $I(u, v)$ here, its edge map can be conveniently calculated by well-established techniques such as Canny edge detector [10]:

$$f^{edge}(u, v) = -|\nabla I(u, v)|^2 \tag{4.39}$$

As can be observed, $f^{edge}(u, v)$ gains high values at position close to edges (corresponding to real-world structures), while has low values in the homogenous regions. An illustrative edge map computed from the picture is depicted in Figure 4.5.

The gradient of edge map, ∇f^{edge}, has vectors pointing toward the edges. This is a very important characteristic in the context of LCAI, because it gives a clue that how the deployment of a submap should be adjusted for a better match to the appearance information. However, the 2D vectors in the gradient of edge



(a) Appearance information $I(u, v)$ (b) Close-look at $f^{edge}(u, v)$

Figure 4.5: Illustration of edge map. (a) The region to be examined is marked by a rectangle. (b) The generated edge map.

map ∇f^{edge} basically have very limited 'capture range', they gain high values only within the immediate vicinity of the edges, if the submap configuration is initially deployed outside this vicinity, the contribution of appearance information in the loop-closing optimization will be trivial.

The solution proposed by Xu [111] is to replace the vector field of edge image's gradient by a dense vector field V. For each pixel (u,v) on the image, a vector $V(u, v)$ is generated to measure, if a point is deployed at this pixel, how its position should be adjusted. A popular way to denote the vectors in V is to use their projection in the x-axis and y-axis:

$$V(u, v) = [\mathrm{v}_x(u, v), \mathrm{v}_y(u, v)] \tag{4.40}$$

So that the 'force' acting on a point can be computed by combining the force acting on it in the direction of x-axis and y-axis. An illustrative dense vector field can be found in Figure 4.6. As can be observed, the vectors (plot as arrows) provide

valuable information on how the submap's position should be adjusted.

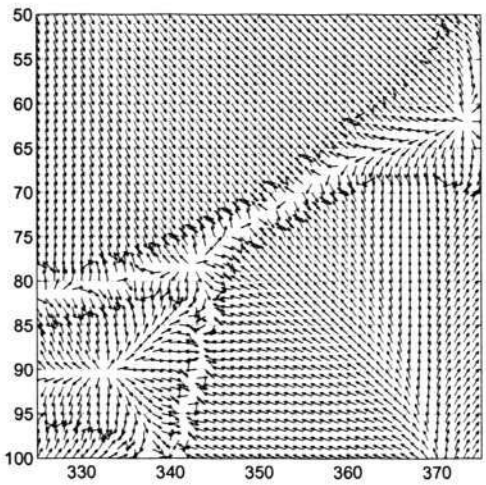


Figure 4.6: The potential field V generated in the region depicted in Figure 4.5.

This vector field is derived from the image to minimize the following target energy function:

$$\varepsilon = \int\int \mu(\mathrm{v}_x(u)^2 + \mathrm{v}_x(v)^2 + \mathrm{v}_y(u)^2 + \mathrm{v}_y(v)^2) + |\nabla f^{edge}|^2 |V - \nabla f^{edge}|^2 dxdy \quad (4.41)$$

where $\{\mathrm{v}_x(u), \mathrm{v}_x(v), \mathrm{v}_y(u), \mathrm{v}_y(v)\}$ is the partial derivatives of the vector field and μ is a constant. This variational formulation has several desirable characteristics. First, in the homogeneous regions where there is no particular information about the environment, in other words, $|\nabla f^{edge}|$ is small, the energy is dominated by the sum of squares of the partial derivatives of the vector field, which is denoted by the first item on the right side. This yields a smoother vector field compared to the one of edge gradient, so that the 'capture range' of the edges can be drastically increased. Second, when $|\nabla f^{edge}|^2$ is large, the second term dominates the integrand. By minimizing it, the V is kept nearly equal to the gradient of the edge map in the vicinity of the edges, so that the edge information is preserved. To

derive the gradient field which minimizes the energy in (4.41) is beyond the scope of this thesis, the readers are referred to [111] for detailed explanation.

Since the 3D submap description is very comprehensive, it is reasonable to assume that the map items have correspondences in the roadmap pictures. Compared with other techniques which employ global constraints, the proposed algorithm involves no manual pre-processing: no effort is required to extract any form of *a priori* global map structure, such as the 'roadmap network' employed in [110]. This job is implicitly performed by the edge detection algorithm in an efficient manner.

As formulated in (4.3), submaps in this work are implemented as point clouds containing N_s points. Given the external and internal parameters of the camera by which the image is captured, these submaps can be conveniently projected into the image domain through coordinate transformation. It is assumed that the camera has already been calibrated and its position is also available. Since the roadmaps are normally built based on a flat plane this assumption should not be over-restricted. To make the formulation concise, this camera model is wrapped by a coordinate transformation operator $\bigodot$, which projects the j^{th} 3D point in submap M_i to the image plane:

$$(u_{i,j}, v_{i,j}) = (x_{i,j}, y_{i,j}, z_{i,j}) \odot \mathrm{L}^{(t)}(s) \tag{4.42}$$

Since each submap M_s is modeled as a rigid body, when deployed within the potential field V, it moves as metal in a magnetic field. The force acting on the submap can be calculated by averaging the forces acting on all the N_s 3D points

inside it. Following the definition of the dense vector field in (4.40), this external force can be decomposed in the direction of x-axis and y-axis:

$$\mathrm{F}^{Ext}\Big(\mathrm{L}^t(s), \mathrm{M}_s\Big) = \begin{pmatrix} \frac{\sum_{j=1}^{N_s} \mathrm{v}_x(u_{i,j}, v_{i,j})}{N_s} \\ \frac{\sum_{j=1}^{N_s} \mathrm{v}_y(u_{i,j}, v_{i,j})}{N_s} \end{pmatrix} \tag{4.43}$$

4.4.2 Constraint from initial submap configuration

4.4.2.1 The constraint in a general form

To propagate the loop-closing error, each submap's pose should be adjusted accordingly. Such adjustment can be imagined as 'force' acting on the submaps, which is similar to the force calculated from appearance information.

For convenience, let us define:

$$\varphi\big(s, s-1\big) = \ominus \mathrm{L}(s-1) \oplus \mathrm{L}(s) - \mathrm{x}(s) \tag{4.44}$$

which can be linearized at the linearization point $\hat{\mathrm{L}}(a), \hat{\mathrm{L}}(b)$ using the Jacobian term:

$$\mathcal{J}_a = \frac{\partial\Big(\varphi(a,b)\Big)}{\partial \mathrm{L(a)}}\bigg|_{\hat{\mathrm{L}}(a),\hat{\mathrm{L}}(b)} = \begin{pmatrix} \cos\hat{\tau}(b) & \sin\hat{\tau}(b) & 0 \\ -\sin\hat{\tau}(b) & \cos\hat{\tau}(b) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{4.45}$$

$$\mathcal{J}_b = \frac{\partial\Big(\varphi(a,b)\Big)}{\partial \mathrm{L(b)}}\bigg|_{\hat{\mathrm{L}}(a),\hat{\mathrm{L}}(b)}$$

$$= \begin{pmatrix} -\cos\hat{\tau}(b) & -\sin\hat{\tau}(b) & -\Big(e(a)-e(b)\Big)\sin\hat{\tau}(b) + \Big(n(a)-n(b)\Big)\cos\hat{\tau}(b) \\ \sin\hat{\tau}(b) & -\cos\hat{\tau}(b) & -\Big(e(a)-e(b)\Big)\cos\hat{\tau}(b) - \Big(n(a)-n(b)\Big)\sin\hat{\tau}(b) \\ 0 & 0 & -1 \end{pmatrix} \tag{4.46}$$

Using above Jacobian terms, the transformation in (4.44) can be linearized as follows:

$$\varphi(a,b) = \mathcal{J}_a\mathrm{L}(a) + \mathcal{J}_b\mathrm{L}(b) - \Big(\underbrace{\mathcal{J}_a\hat{\mathrm{L}}(a) + \mathcal{J}_b\hat{\mathrm{L}}(b) - \psi\big(\hat{\mathrm{L}}(a), \hat{\mathrm{L}}(b)\big) + \bar{\mathrm{x}}_{N_m}}_{y}\Big) \tag{4.47}$$

According to the elaboration in [32], the energy function in (4.22) can be finally formulated as:

$$E^M(\mathcal{L}) \;=\; \mathcal{L}^T A\mathcal{L} - 2\mathcal{L}^T B + C \tag{4.48}$$

where

$$A = \sum_{\{a,b\}\subset\mathcal{X}} \begin{pmatrix} \ddots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \mathcal{J}_a^T(\mathrm{P}_b^{-1})\mathcal{J}_a & \cdots & \mathcal{J}_a^T(\mathrm{P}_b^{-1})\mathcal{J}_b & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \mathcal{J}_b^T(\mathrm{P}_b^{-1})\mathcal{J}_a & \cdots & \mathcal{J}_b^T(\mathrm{P}_b^{-1})\mathcal{J}_b & \\ \cdots & \cdots & \cdots & \cdots & \ddots \end{pmatrix}$$

$$B = \sum_{\{a,b\}\subset\mathcal{X}} \begin{pmatrix} \cdots & \mathcal{J}_a^T(\mathrm{P}_b^{-1})\mathrm{y} & \cdots & \mathcal{J}_b^T(\mathrm{P}_b^{-1})\mathrm{y} & \cdots \end{pmatrix}^T$$

where y, C are constants and each pair of $\{a, b\}$ represents a transformation between two submaps. For simplicity of notation, let us define:

$$\begin{pmatrix} \mathcal{J}_a^T(\mathrm{P}_b^{-1})\mathcal{J}_a & \mathcal{J}_a^T(\mathrm{P}_b^{-1})\mathcal{J}_b \\ \mathcal{J}_b^T(\mathrm{P}_b^{-1})\mathcal{J}_a & \mathcal{J}_b^T(\mathrm{P}_b^{-1})\mathcal{J}_b \end{pmatrix} = \begin{pmatrix} Q_{a,a}^b & Q_{a,b}^b \\ Q_{b,a}^b & Q_{b,b}^b \end{pmatrix} \tag{4.49}$$

In a similar manner,

$$\mathcal{J}_a^T(\mathrm{P}_b^{-1})\mathrm{y} = R_a^b \tag{4.50}$$

Based on the loop graph introduced in Figure 4.2, there should be:

$$B = \begin{pmatrix} R_1^2 & R_2^2 + R_2^3 & \cdots & \cdots & R_{N_m-1}^{N_m-1} + R_{N_m-1}^{N_m} & R_{N_m}^{N_m} \end{pmatrix}^T \tag{4.51}$$

Accordingly, the force representing the mapping information can be computed by (4.38):

$$-\nabla E^M(\mathcal{L}) \;=\; -\frac{\partial(\mathcal{L}^T A\mathcal{L} - 2\mathcal{L}^T B + C)}{\partial\mathcal{L}} = (B - A\mathcal{L}). \tag{4.52}$$

Please note that the iteration index t is abbreviated in this equation.

4.4.2.2 Specific form for single loop

For the mapping process Λ in this work, the submaps are constructed incrementally, and each submap is only connected with its two adjacent submaps. As a result, before loop-closure is detected, the information matrix A at time -1 is a tri-diagonal matrix. Instantaneously, a loop-closing algorithm detects that submap N_m is identical to submap 1, so that the submap configuration resembles a circularly-linked list. Given such submap configuration, the general form of $-\nabla E^M(\mathcal{L})$ in (4.52) can be further analyzed as follows.

At each iteration t, submaps' poses are adjusted sequentially, beginning from the last one. The force acted on submap $(N_m - 1)$ is :

$$\mathrm{F}^{M,t}(N_m-1) = \Big(R^{N_m-1}_{N_m-1} + R^{N_m}_{N_m-1} - \big(Q^{N_m-1}_{N_m-1,N_m-1} + Q^{N_m}_{N_m-1,N_m-1}\big)\mathrm{L}^t(N_m-1) - Q^{N_m-1}_{N_m-1,N_m-2}\mathrm{L}^t(N_m-2) - \underbrace{Q^{N_m}_{N_m-1,1}\mathrm{L}^t(1)}_{large}\Big) \tag{4.53}$$

As explained previously, the item marked in the right side of above equation accounts for the major part of loop-closing error, and the energy that error introduces. This item 'drives' submap $N_m - 1$ toward submap 1, in effort to 'close-the-loop'. Accordingly, all of the following submaps are shifted to propagate back this loop-closing error:

$$\mathrm{F}^{M,t}(s) = \Big(R^s_s + R^{s+1}_s - Q^s_{s,s-1}\mathrm{L}^t(s-1) - \big(Q^s_{s,s} + Q^{s+1}_{s,s}\big)\mathrm{L}^t(s) - Q^{s+1}_{s,s+1}\mathrm{L}^t(s+1)\Big)$$

, until the first submap:

$$\mathrm{F}^{M,t}(1) = \Big(R^2_1 - Q^2_{1,1}\mathrm{L}^t(1) - Q^2_{1,2}\mathrm{L}^t(2) - Q^{N_m}_{1,N_m-1}\mathrm{L}^t(N_m-1)\Big)$$

The pose of submap M_s is updated according to its two adjacent submaps $s-1$ and $s+1$. Such an updating scheme essentially embodies the concept of relaxation, which iteratively employs local, greedy and easy-to-implement processing to obtain an optimal solution. Every adjustment reduces the energy contained in (4.20), as such iteration proceeds, the minimum of (4.22) is expected to be achieved.

It can be observed that, there is no item corresponding to x(1) in above formulations. In other words, the energy in (4.22) will not be affected by how the submaps are deployed in a global coordinate, because this energy only represents the inner connections between different submaps. Consequently, the configuration which minimizes (4.22) can only guarantee the consistency among submaps, rather than the consistency between this configuration and the real-world.

Appearance information is introduced as an additional constraint to compensate the insufficiency of loop-closing with mapping information alone. Therefore, minimizing the energy in $E^{Ext}(\mathcal{L})$ essentially establishes a correspondence between the submap configuration and the world coordinate system.

4.5 Experimental results

4.5.1 Platform and environment

Several experiments were conducted to examine the performance of the proposed loop-closing algorithm. The testing platform is a 4×4 vehicle with a range scanners mounted on the top, as in Figure 4.7. This vehicle is also equipped with a Inertial Science DMARS-I IMU to estimate its motion. A high accuracy GPS is used to provide ground truth for reference. The experiments were carried out on a campus.

(a) The vehicle (b) Top laser

Figure 4.7: The testing platform with laser scanner mounted on the top.

Several photos of the testing environment are available in Figure 4.8.

(a) (b) (c)

Figure 4.8: Three photos taken at different places in the vehicle's trajectory.

4.5.2 Constructing 3D submaps $\mathcal{M}$

The total length of the vehicle's trajectory is 1,890 meters. During the experiment, 6,237 frames of 2D range scans were collected. Using the submap segmentation technique discussed in the previous chapter, totally 32 submaps have been constructed.

The submap's origin is set to be the vehicle's pose at the moment when this

submap is constructed. All the 3D points within the submap are then stored relative to its local coordinate system. Since the objective of this work is to achieve global consistency, the local map building is naively performed by directly projecting the range scans from the vertical laser scanner to the vehicle's poses read from IMU. As observed in the experiment, since the submap's length is constrained, i.e, generally less than 30 meters, the accumulated error of IMU readings within each submap is moderate. However, incorporating a local SLAM scheme with EKF or particle filter will surely benefit the submap building process and achieve higher accuracy within the local submaps, as in [40] [112][68].

Using above approach, a 3D submap M_s can be imaged as a pile of 2D laser slices horizontally deployed. By accumulating such 'slices' of range data, a submap can be conveniently represented by a point cloud. Two submaps are depicted in Figure 4.9 and Figure 4.10.

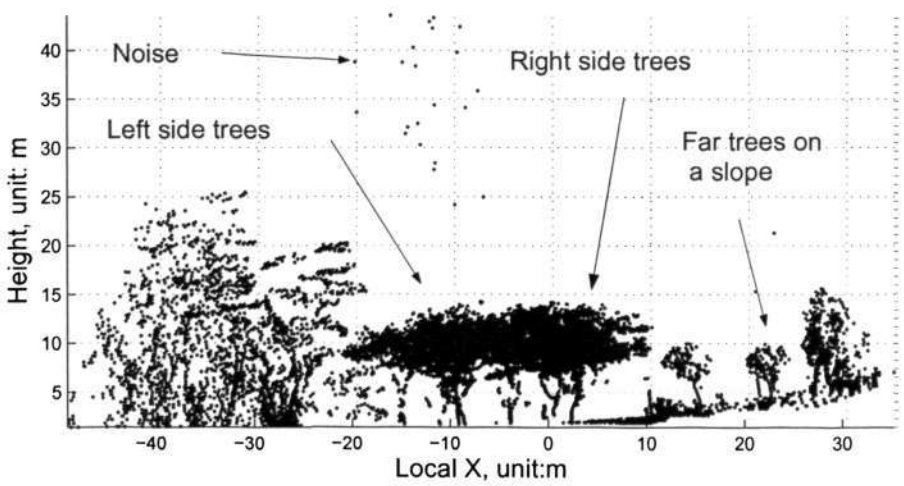


Figure 4.9: The front view of the 3D submap 1.

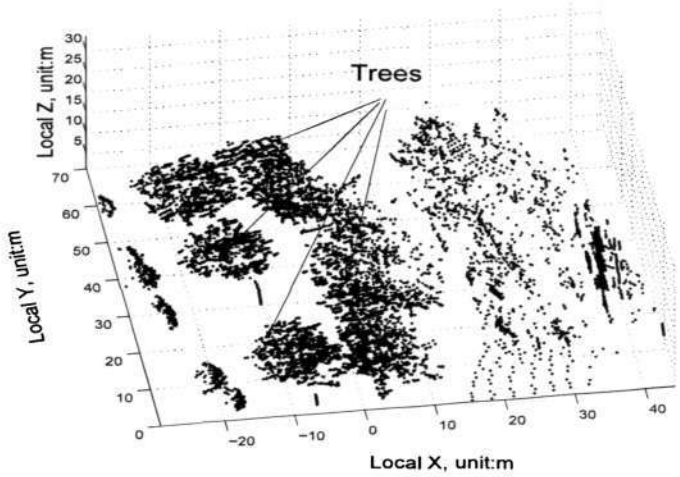


Figure 4.10: Submap 2 from 45° view. The road can be identified by the trees beside it.

4.5.3 The vehicle motion model $\mathcal{P}$

The parameter $\mathcal{P}$ models the uncertainty of the estimated transformations between submaps. If the deformable submap graph is imaged as a sequence of blocks connected by springs, the value of $\mathcal{P}$ determines the toughness of the springs, or equivalently, how much the two submaps can deform from their initial configuration. Although various approaches are available to compute the $\mathcal{P}$, as in [90] and [64], in this work a synthesis $\mathcal{P}$ is employed to examine the proposed algorithm.

In the experiments, the transformation uncertainties are set to be proportional to the aptitude of vehicle's transformation with respect to the previous submap. Let $\bar{\mathrm{x}}_s = \begin{pmatrix} \Delta\bar{x}_s & \Delta\bar{y}_s & \Delta\bar{\gamma}_s \end{pmatrix}$ denotes the transformation between submap $s-1$ to s, the corresponding synthesis P_s is calculated as:

$$\mathrm{P}_s = \begin{pmatrix} \varsigma_x \Delta\bar{x}_s & 0 & 0 \\ 0 & \varsigma_y \Delta\bar{y}_s & 0 \\ 0 & 0 & \varsigma_\gamma \Delta\bar{\gamma}_s \end{pmatrix}$$

where $\{\varsigma_x, \varsigma_y, \varsigma_z\}$ are constants controlling the amount of error. Since the vehicle is

moving on the road, ς_x and ς_y are set to be small to represent the small translation error. Meanwhile, the ς_γ is set to be comparatively large to model the error during the turning. The initial submap configuration constructed by the above uncertainty model is depicted in Figure 4.11. It can be observed that this synthetic uncertainty

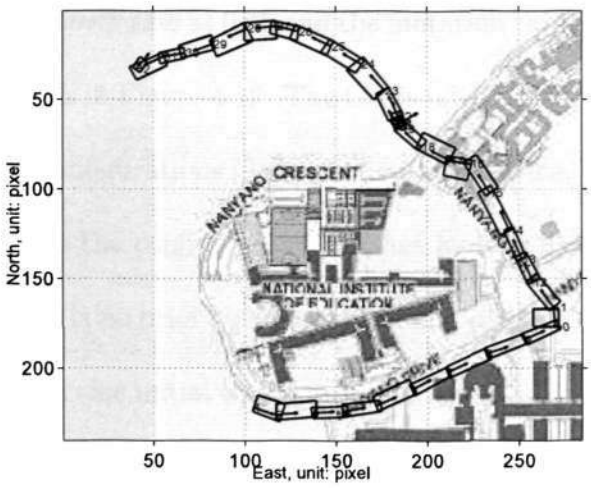


Figure 4.11: The initial submap configuration built by a synthesis uncertainty model $\mathcal{P}$.

model is consistent with the general characteristic of SLAM's error: the longer the vehicle travels, the more error will be accumulated.

4.5.4 Initialization using genetic algorithm

In this work, the objective of genetic algorithm is essentially to obtain a satisfactory initial configuration for the following gradient-based submap deformation. Therefore, it is unnecessary to employ too large a population or too many generations, which can consume a lot of computational resources.

In this work, the size of population is fixed at 150 throughout the evolution, that is, at each generation, there will be 150 chromosomes compete against each other

for survival. To examine the characteristics of the GA, the number of generations is fixed at 350. However, it is also possible to terminate the evolution if a convergence is detected, in which the mean fitness of the population stay unchanged (or with minor changes) for a certain number of generations.

By setting the crossover rate at 0.65 and the mutation rate at 0.025, the submap configuration evolves as in Figure 4.12. The GA is initialized by randomly generating possible submap configurations in the configuration space. The first figure, i.e., the iteration 1, depicts the configuration that has highest fitness when the GA is initialized. Since there is no prior on the position and heading of the configuration, it can be observed that this initial configuration is significantly wrong. Thereafter, the population starts to evolves. Several generations are selected to observe the evolution of the population. For each of them, the configuration with maximum fitness is depicted. These figures demonstrate that, by selecting chromosomes with high fitness and removing those poor chromosomes, the GA can gradually find a configuration that close to the correct one. Such improvement in fitness can also be observed from history of mean fitness during the evolution of population, as depicted in Figure 4.13.

Basically, as a global search strategy, the genetic algorithm is not designed for realtime applications. It generally consumes considerable computational resources and is slow to converge. Qualitatively speaking, its complexity scales linearly with the size of population, the number of generations, and the processing of each gene.

$$\mathsf{Ops}_{GA} \propto [\#\mathsf{population}] \times [\#\mathsf{generation}] \times [\mathsf{size(submap)}] \tag{4.54}$$

The processing time of a chromosome depends on the number of points sampled from it. The more points are sampled, the more expressive the submap is; meanwhile, the more computation is required to process a submap.

4.5.5 Local search using only appearance information

Submap 15 is selected to examine how a submap 'moves' in the potential field generated from pictorial information. Figure 4.14 indicates the pose of submap No. 15 in the global coordinate. Correspondingly, a close-up look at the GVF field in that region is depicted in Figure 4.15.

The local environment at submap 15 is a T-shape crossroad. It can be noticed that, without any knowledge on the vehicle's trajectory, the field equally pushes submap 15 in the two indicated directions F1 and F2. This leads to a wrong convergence as showed in Figure 4.16. The result reveals the limitation of using appearance information alone for loop-closing.

The problem of submap 15 actually exists in all the other submaps when there is no information from the mapping result. Figure 4.17 shows the result of loop-closing with only appearance information. It can be observed that, although all of the submaps coincide with the vehicle's true trajectory, there exist serious conflictions between each pair of adjacent submaps. This is because that the transformations between two nearby submaps estimated from robot mapping have not been modeled in the optimization process.

4.5.6 Local search using only mapping information

If only the mapping information Φ is available, there will be a generic loop-closing problem, which can be formulated as:

$$p(\mathcal{X}|\bar{\mathcal{X}}, \mathcal{M}, \mathcal{P}) \tag{4.55}$$

Without constraint from appearance information, it is very difficult for the SLAM to converge by visiting the environment for only once, as showed in Figure 4.18. The result in this figure is a contrast to the one with only appearance information available (see Figure 4.17): each two consecutive submaps are connected in a smooth manner, and very little confliction can be observed, but the whole submap structure is erroneous. Shortly speaking, the map is only consistent at the local level, but incorrect at the global level. This result proves that the guidance from global information is definitely necessary if the map needs to converge within just one loop.

4.5.7 Local search using both appearance and mapping information

The final mapping result of the proposed algorithm is showed in Figure 4.21. The parameter setting for this result is $\epsilon = 0.05$. Accuracy of the algorithm is measured by the root-mean-square (RMS) error between submap configuration and the ground truth state $\mathcal{X}^g = \left(\mathrm{L}_1^g, \mathrm{L}_2^g, \cdots, \mathrm{L}_{N_m}^g\right)$. The ground truth was obtained by the onboard GPS/INS system, which is depicted in Figure 4.19. Therefore, the

RMS error is defined as:

$$e = \sqrt{\frac{1}{N_m - 1} \sum_{s=1}^{N_m - 1} \left(\mathrm{L}(s) - \mathrm{L}_s^g \right)^2} \tag{4.56}$$

The submap graph takes 6 iterations to converge to an optimal configuration, which is showed in Figure 4.21. During these 6 iterations, the submap graph moves gradually to the correct configuration, which can be observed through the reduction of RMS error in Figure 4.20. By comparing Figure 4.21 with the ground truth acquired from GPS data, it can be observed that the proposed algorithm can achieve both local and global consistences by visiting the environment for only once. After rendering the points cloud $\mathcal{M}$ to *the maximum a posteriori* submap configuration using VRML, the final 3D mapping result for the target environment is depicted in Figure 4.22. It can be observed that the environment can be accurately mapped by one robot visiting the environment for only once.

The vehicle platform in this work is a 4×4 pickup with a minimum speed, while Howard [40] used a small robot which could travel as slow as necessary, so that samples (laser reflections) in [40] are much denser than ours in Figure 4.22. However, the presented mapping result can nevertheless provide thorough description for the environment, Figure 4.23 is a close look at the 3D map, the structure across the road can be easily identified to be a bridge. This bridge is also depicted in the center of Figure 4.8(b).

To demonstrate the superiority of the proposed algorithm, the accuracies of different loop-closing techniques are depicted in Figure 4.24. For each submap, the Euclidean distance between the LCAI estimate and GPS ground truth is com-

pared: the proposed technique achieves much higher accuracy than the other two methods. It can be further noticed that, for generic loop-closing with only mapping information (marked with squares), the error grows almost monotonously from the beginning to the middle of the submap configuration. Such error is caused by accumulation of local filtering errors: the longer the robot travels, the bigger the error will be. Loop-closing tries to re-distribute this error to obtain a more consistent map, which leads to the drastic decrease of error from the middle to the end of submap configuration.

4.5.8 The parametric ϵ

The setting of ϵ plays an important role in the LCAI, it controls how much the constraint from appearance information can deform the initial submap configuration. Mathematically, it relates the squared pictorial gradient in (4.21) to the non-dimensional χ^2 in (4.22). However, deriving ϵ analytically is a challenging task, because the quality of the input image (with respect to matching with real world structure) is difficult to measure. Based on the available data, in this thesis the author tries to qualitatively analyze the characteristics of the ϵ.

During the experiment, it is observed that bigger value should be given to ϵ when the image's quality is poor, e.g., blurry or noisy. The original image, and a synthetic blurry image are depicted in Figure 4.25. As can be observed, the blurry image provides much less information about the environment than the original image. Consequently, if the same ϵ is employed in this case, we actually 'over-weight' the contribution of appearance information in the LCAI. Consequently,

the final mapping result is distorted by the error coming with the blurry image, as in Figure 4.27.

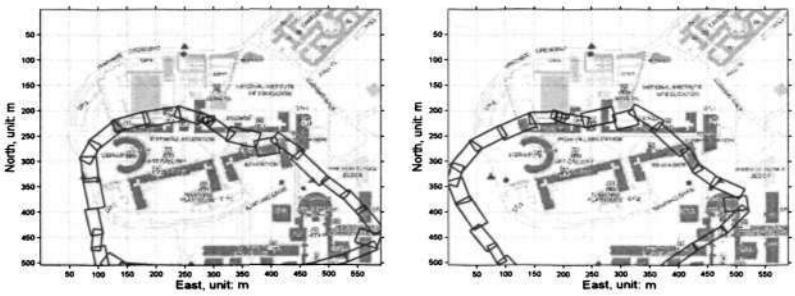
A bigger ϵ puts more weight on the loop-closing using only mapping information, which is equivalent to release the constraint from appearance information. This can to some extent alleviate the distortion existing in Figure 4.26. The mapping result using bigger ϵ is depicted in Figure 4.27. Being blurry, the input image now is less informative than the original one, consequently, LCAI basically cannot achieve the performance as using non-blurred image: there are still minor distortions existing in the final map, as in the regions C and D indicated by the arrows.

The setting of ϵ is also variant to the scale of input image. In the experiments, only two scales of the input image are available, one is at the scale of 1 pixel = 1.189 meter, as used in Figure 4.21. The other one is at a higher scale: 1 pixel = 3.303 meter. These two input images are depicted in Figure 4.28.

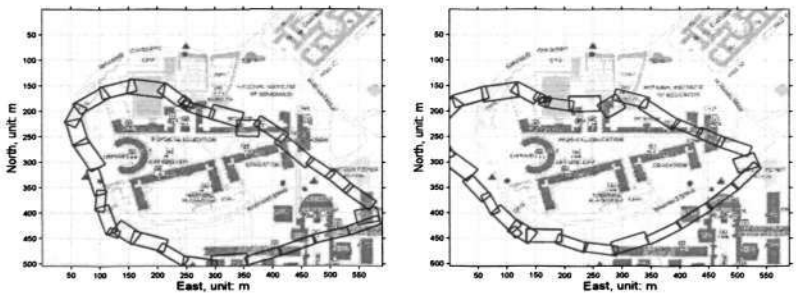
When image has a larger scale, i.e., larger meter per pixel value, the constraint calculated from mapping information is reduced when projected into the image domain, while the external force from appearance information will be about the same due to the characteristic of the dense vector field. Therefore, the increase of scale essentially amplifies the constraint from appearance information. If the same $\epsilon = 0.05$ is employed as the one used in the small scale image, the resulting map could be seriously distorted. For example, in the regions marked as A and B in Figure 4.29(a), the constraint from mapping information has been released in scaling, so it cannot compensate the force from appearance information which

strongly attracts submaps to wrong places. It can thus be observed from Figure 4.29, by setting ϵ to larger value, the distortion can be reduced.

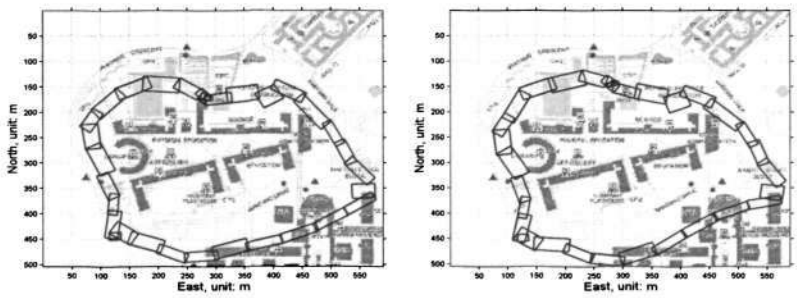
For large-scale input image, a larger value of ϵ should be employed. The distortion existing in Figure 4.29(a) is reduced when ϵ is set to be 0.09, the improvement can be found in Figure 4.29 (b). Please note that there are still errors existing in the result using large ϵ, as indicated by C. This can be attributed to the less detailed information provided by the large-scale image, which has a lower resolution than the small scale one. The accuracy of the mapping results using different setting of ϵ is plotted in Figure 4.30. By comparing it with Figure 4.24, it can be observed that, although the accuracy is improved by increasing the ϵ, the overall performance is not as good as the one using original image. This can also be explained by the lose of information when the image is zoomed out.



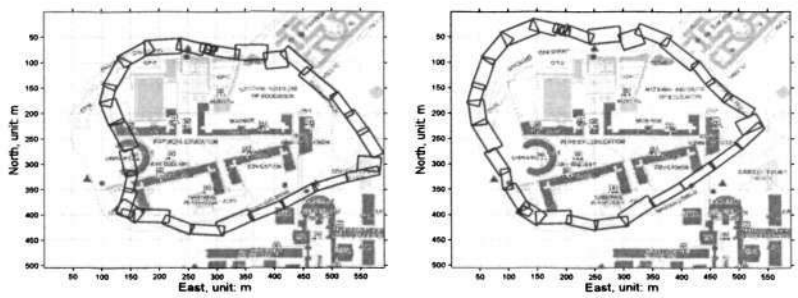
(a) 1st generation (b) 35th generation



(c) 67th generation (d) 147th generation



(e) 262nd generation (f) 294th generation



(g) 343th generation (h) 350th generation

Figure 4.12: The evolution of the submap configuration in the genetic algorithm.

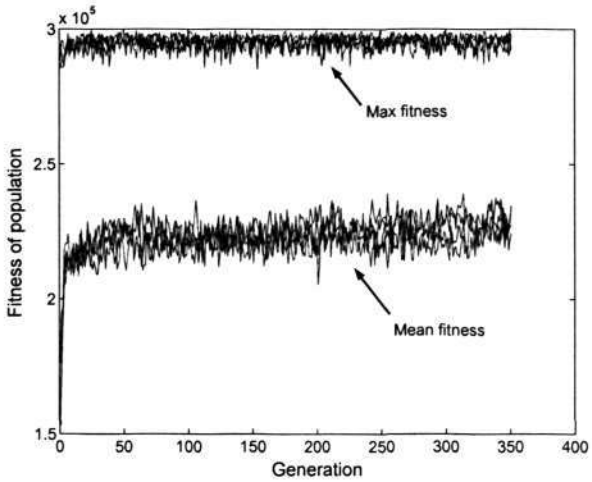


Figure 4.13: The evolution of mean fitness in LCAI based on genetic algorithm. The corresponding parameter setting is $p_c = 0.65$; $p_m = 0.025$; ϵ=0.05

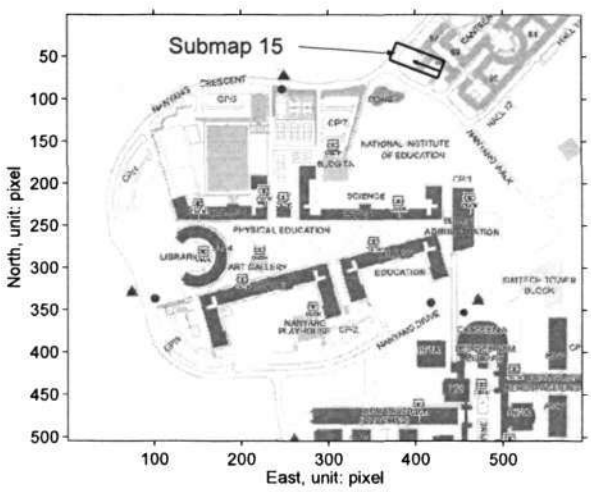
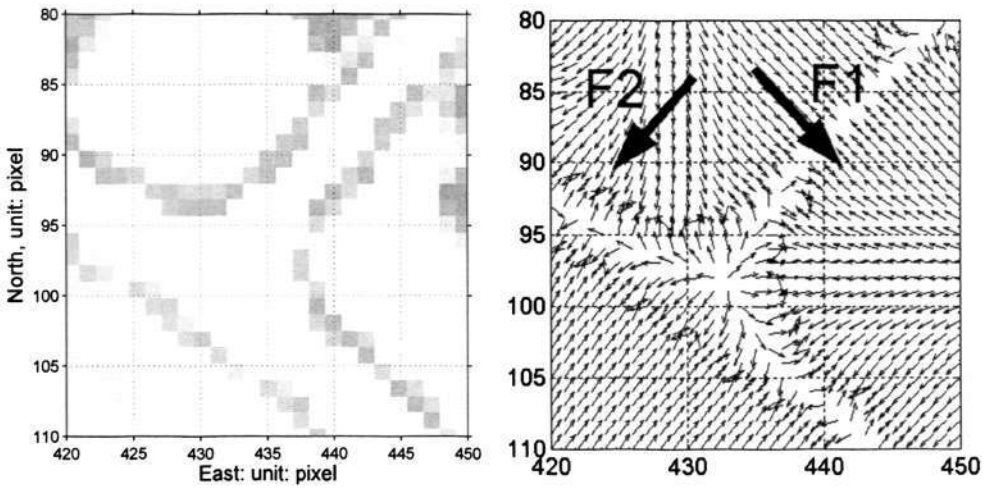


Figure 4.14: Submap 15 in the global coordinate. A close-up look at this region can be found in Figure 4.15.



(a) Zoom-in look at the photo.

(b) Potential filed

Figure 4.15: (a) Close-look at the roadmap image. (b) The generated GVF field at the region of submap 15, within the region above the T-shape road, forces are mostly in two directions.

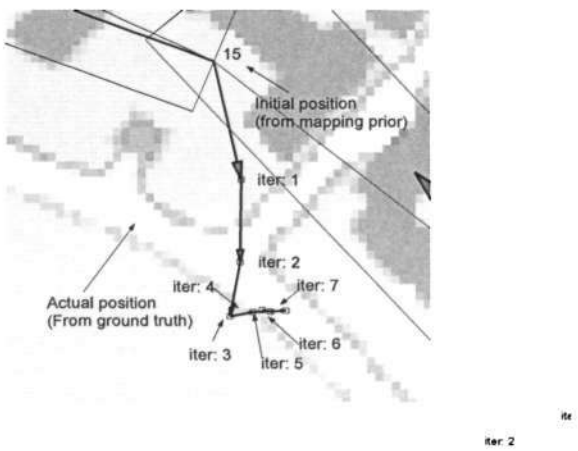


Figure 4.16: Submap 15's shift in the potential field without SLAM information.

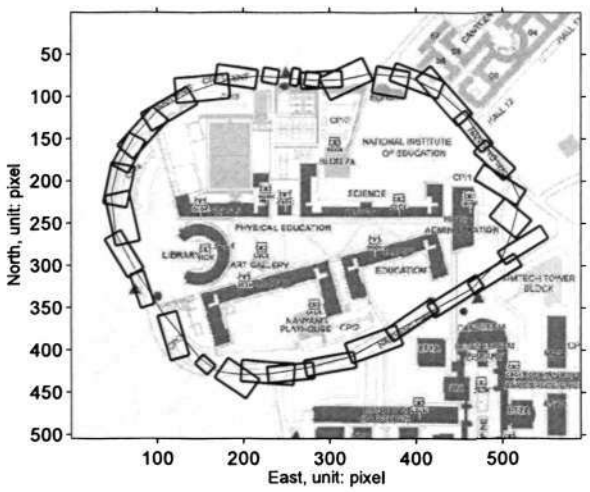


Figure 4.17: The final mapping result using only appearance information.

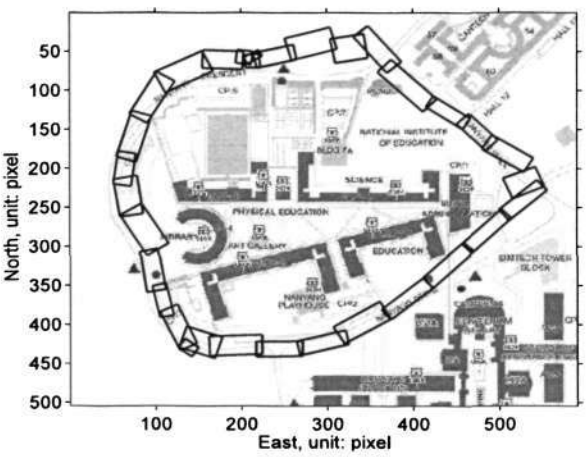


Figure 4.18: The loop-closing result using only mapping information.

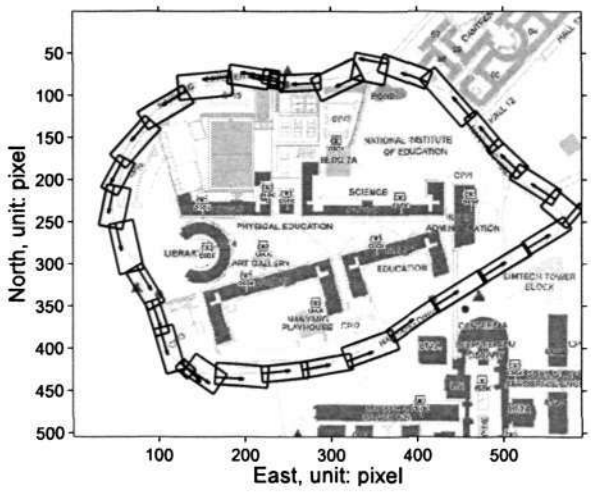


Figure 4.19: The map ground truth calculated from GPS data.

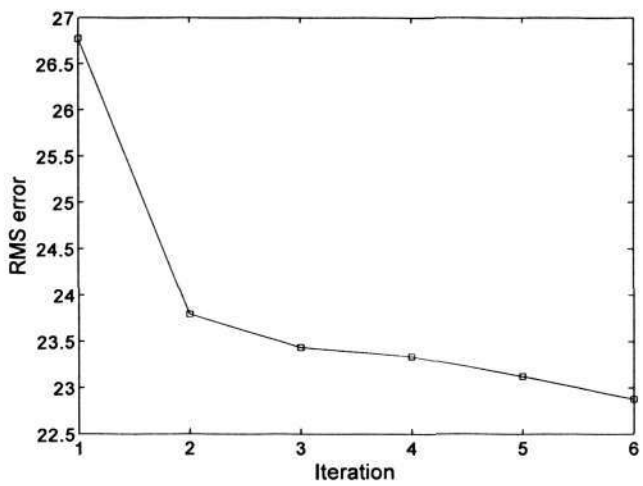


Figure 4.20: The submap graph takes 6 iterations to converge.

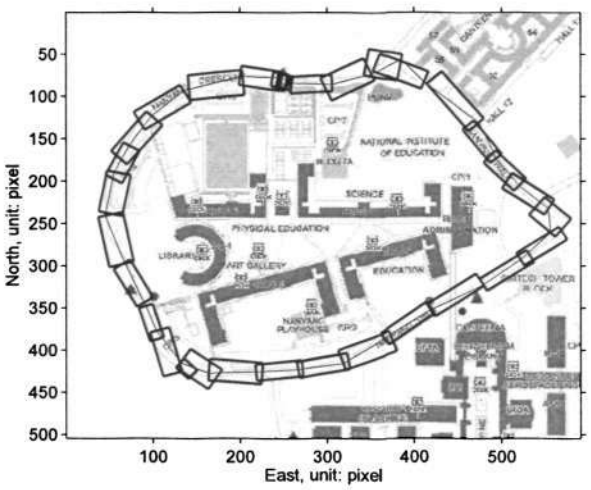
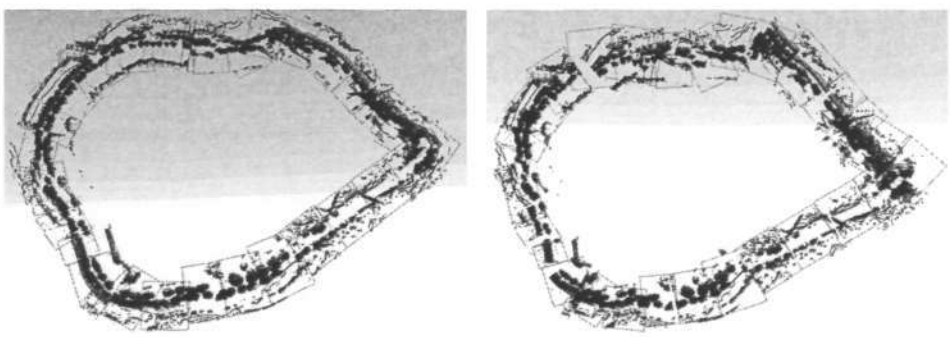


Figure 4.21: The mapping result using the proposed algorithm. The loop has been accurately closed by visiting the environment for only once.



(a) Ground truth (b) Result of LCAI

Figure 4.22: The 3D environmental map rendered by VRML, using the ground truth from GPS/INS and the result of LCAI.

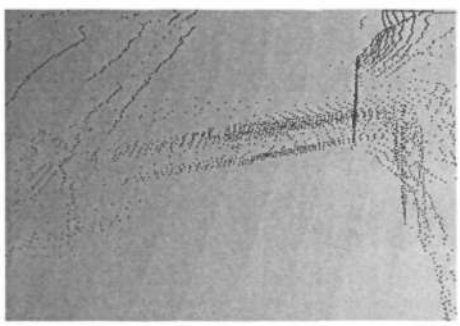


Figure 4.23: Close-look at the VRML mapping result: a bridge can easily be identified. The bridge is also observed in Figure 4.8(b).

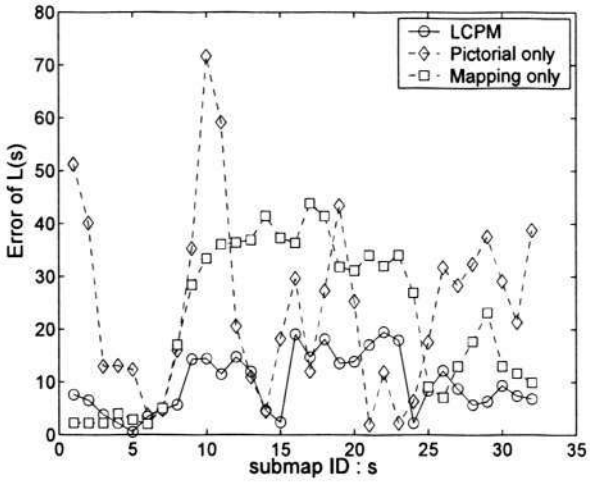
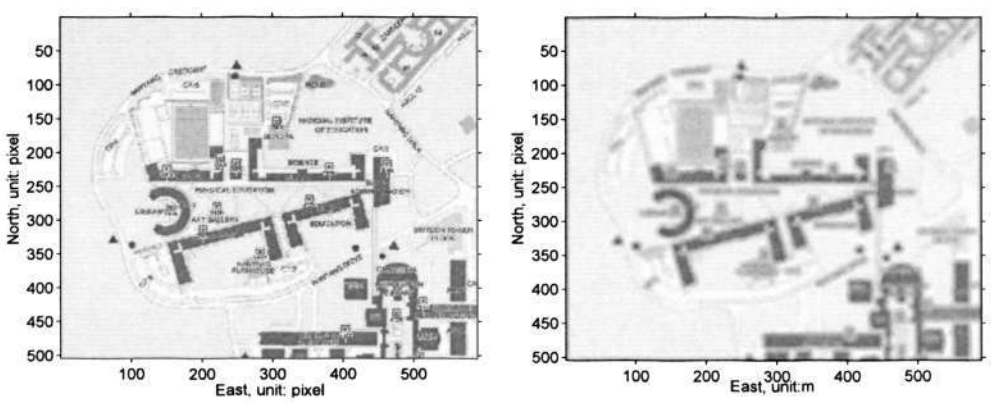


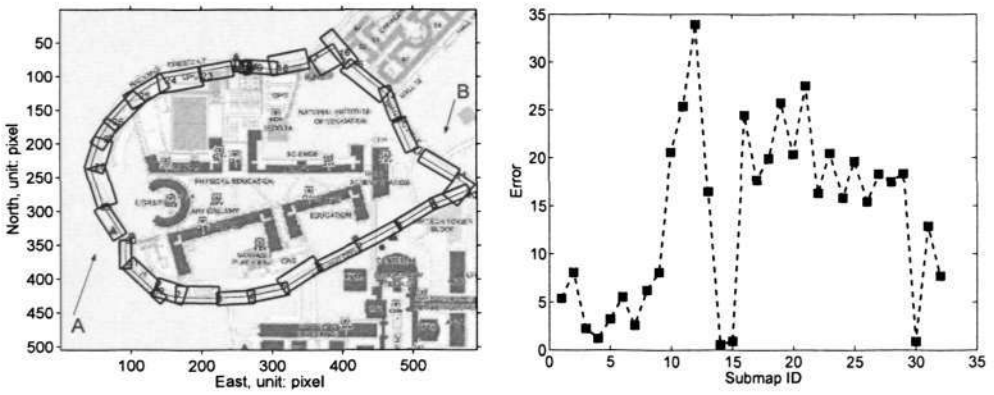
Figure 4.24: Accuracy comparison between LCAI, mapping information only (generic SLAM), and appearance information only.



(a) Original image

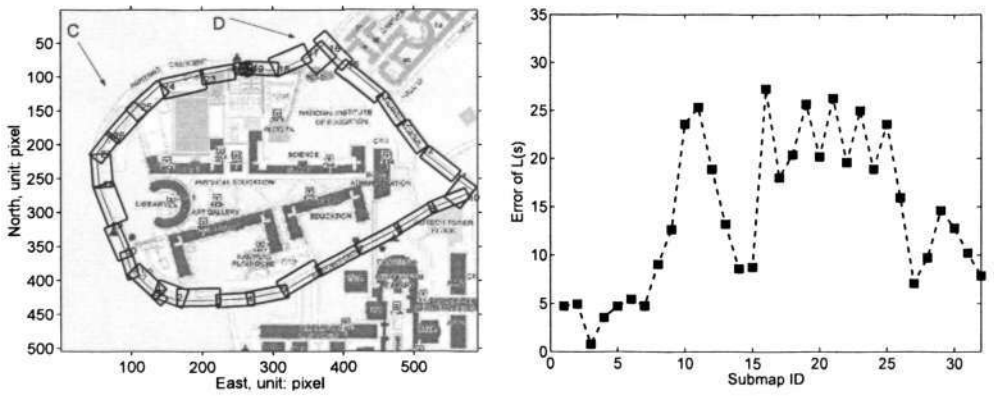
(b) Blurred image

Figure 4.25: The original image and the blurred image. They are used to analyze the characteristic of ϵ.



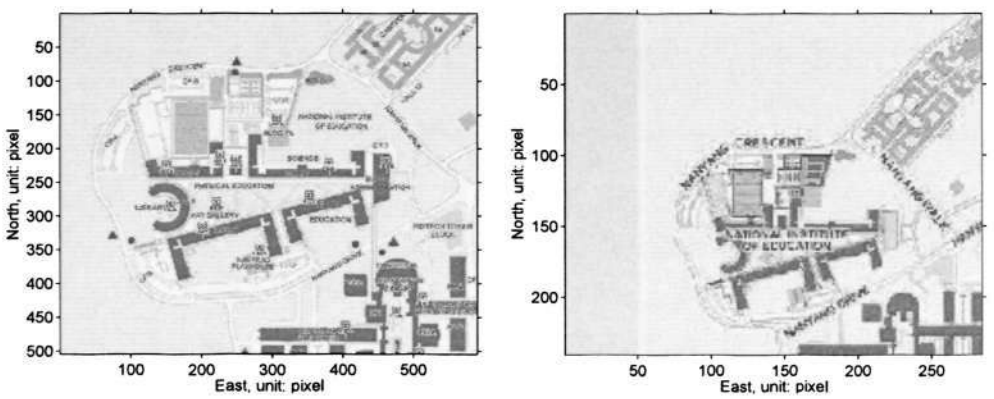
(a) Mapping result (b) Error analysis

Figure 4.26: The LCAI result using blurry image, when $\epsilon = 0.05$. Significant distortion can be observed in the indicated regions as A and B.



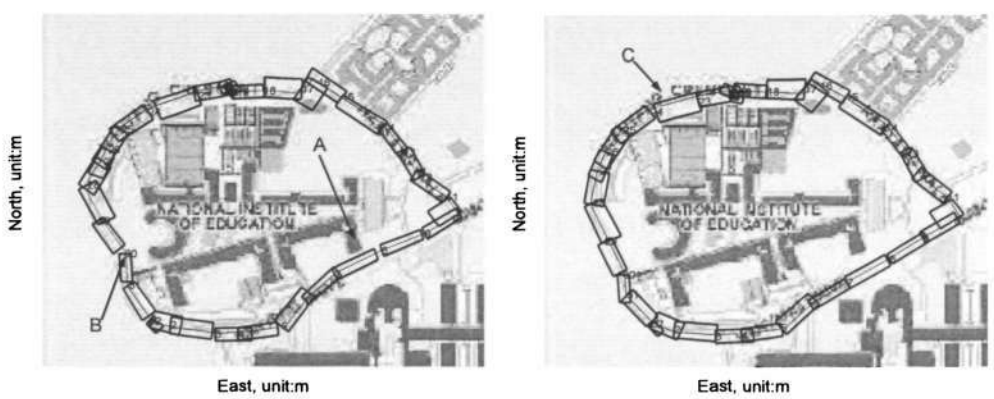
(a) Mapping result (b) Error analysis

Figure 4.27: LCAI using blurred input image when $\epsilon = 0.09$. By increasing the ϵ, the distortion can be reduced. However, some minor distortions can still be observed.



(a) 1 pixel = 1.189 meter (b) 1 pixel = 3.303 meter

Figure 4.28: The images at two different scales. They are used to analyze the characteristic of ϵ.



(a) Mapping result at $\epsilon = 0.05$ (b) Mapping result at $\epsilon = 0.09$

Figure 4.29: Comparison of mapping result using large-scale image, at different settings of ϵ.

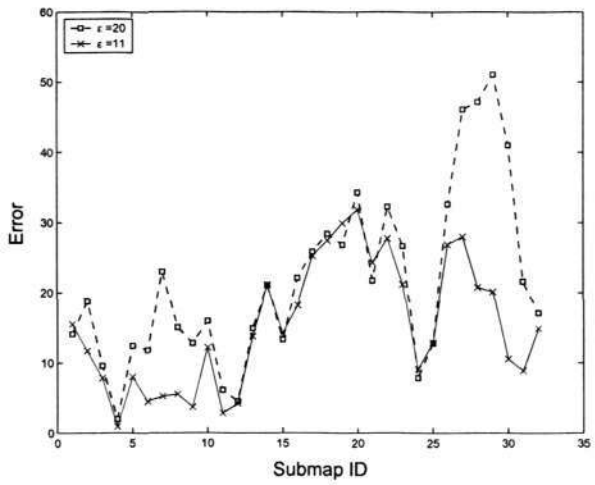


Figure 4.30: Accuracy comparison between different ϵ values for zoom-out image. Setting ϵ to a larger value can to some extent reduce the error when the input image is at a larger scale.

4.5.9 Algorithmic complexity of gradient search

Using absolute pose representation, the generic SLAM algorithms generally scale as $\mathcal{O}(n^3)$, where n in the total number of map items. In this work, the relative pose representation introduces highly sparse matrices, which can be exploited in the optimization by means of well-established sparse methods. Therefore, the presented algorithm can close the loop in $\mathcal{O}(N_m^1)$ computation time, where N_m is the size of submap transformation $\mathcal{X}$.

However, the appearance information causes additional computational burdens in the off-line and online processing:

$$\mathsf{Ops}_{SNAKE} = \mathsf{Ops}_{offline} + \mathsf{Ops}_{online} \tag{4.57}$$

The off-line processing's objective is to construct the potential field from input

image, whose operation is linear:

$$\mathsf{Ops}_{offline} \propto [\mathsf{image\ size}] \times [\#\mathsf{iteration}] \tag{4.58}$$

The more iterations it loops, the further the field can affect. How many iterations should be used may vary with the applications. This pre-processing can be conducted independently, which means that the field can be constructed even before mapping begins. Furthermore, for a given environment, once the GVF field V is built, no more adjustments are necessary. Therefore, one single pictorial model can support several SLAM algorithms running in a parallel mode, which is a desirable characteristic in multi-robot mapping tasks. The algorithm is implemented using Matlab version 6.5, on a laptop with a Pentium M 1.5G Hz CPU. For a raw image at the size of 504 × 509 pixels, it took 96.1256 seconds to loop for 160 iterations and construct the GVF.

The online operation of LCAI scales linearly with two terms:

$$\mathsf{Ops}_{online} \propto [\mathsf{submap\ number}] \times [\mathsf{submap\ size}] \tag{4.59}$$

The number of submaps is determined by the submap segmentation scheme.

The submap's size depends critically on submap representation. In this work, an unstructured representation is employed. A submap's size is equivalent to the number of 3D points that it contains. To improve the efficiency, the raw range data are pre-processed to remove those points too close to each other. The algorithm then takes around 2 seconds to optimize the map with 6 iterations. According to the experiments, the time consumed by each single iteration is about the same, which is between 0.29 and 0.3 second.

4.6 Conclusion

In this chapter, a appearance information based technique is presented to conduct large scale mapping by only one single robot going through the environment for only once. The map is represented as a sequence of submaps deployed as a deformable configuration in a global coordinate system. The submap graph is estimated by *maximum a posteriori*, based on both mapping information and appearance information. The MAP estimation is finally formulated as an energy minimization problem, the genetic algorithm and active contour algorithm are applied to optimize the submap configuration efficiently.

A major limitation of LCAI in its current form is the setting of parameter ϵ. It plays an important role in the iterative energy minimization process, however, to derive the ϵ is a challenging task as it is difficult to obtain a quantification of the error in the image. On the other hand, the aerial photos or satellite images required by LCAI are often sensitive and not available to the public, this also limits further experiments to analyze the characteristics of ϵ.

Due to the limitations on experimental facilities, in this work, no satellite or aerial images can be utilized for fusion. The author advocates here that, the joint research between image processing and SLAM will become a promising direction. As demonstrated in [82], today's remote sensing and active contour technology has become advanced enough to provide global constraints for a SLAM algorithm. This topic deserves further research endeavors.

Similar to genetic algorithm, simulated annealing (SA) [50] is another popular

optimization technique. Rather than an analogy to the competition in nature, SA solves the optimization problem by approximating thermodynamics. Energy in thermodynamics is the cost function in SA, the ground state, change of state, and temperature in thermodynamics translate to the optimal solution, the neighboring and the control parameter in simulated annealing. Therefore, the abstract system can be described as if it were a thermal physical system for which the aim is to locate the ground state as the temperature is diminished. Both these two techniques are powerful in terms of their searching capabilities. However, a major advantage of GA against SA is the fact that GA can be implemented in a parallel computing system. The chromosomes can processed independently, even by different computers, while simulated annealing must be computed sequentially. With its discrete nature, the genetic algorithm can be possibly implemented on a discrete system composed of several computing units. Each computing unit may process a certain portion of the whole population, so the computational time would be significantly reduced.

However, the author also agree that, considerable progresses have been made in the simulated annealing domain. A lot of innovative derivatives with attractive characteristics have been proposed. Using simulated annealing in SLAM is still an open issue that deserves further research.

Chapter 5

Conclusion and Recommendations

5.1 Conclusion

This thesis presents a probabilistic framework for conducting mobile robot localization and mapping using appearance information. Being completely independent of metric level feature information, the proposed algorithm outperforms the conventional SLAM algorithms with respect to its capability of handling highly unstructured cross-country environments. Based on a hierarchical map representation, this framework can achieve much higher robustness and efficiency in large-scale localization and mapping tasks. Meanwhile, the proposed framework makes trivial assumptions about the characteristics of the sensor input, and therefore can be applied to various mapping scenarios and sensor configurations. The major innovations in this thesis are itemized below.

- **Hierarchical map representation**

  In this thesis, the map representation is organized as a three-layer hierarchy. The top layer is the submap layer: successive sensor measurements are grouped into a sequence of clusters that distinctively describe the appearance

information of the local environments. The second layer is the topological node layer which further segments the submaps according to a fixed resolution. This layer makes it convenient to model the motion of the robot at the topological level. The third, also the lowest layer, stores the metric information for the raw data. It provides the most comprehensive description of the environment.

- **Appearance based inference**

  By exploiting the environment's appearance information, rather than metric geometric information, the proposed SLAM framework in this thesis is completely independent of features or landmarks, which enables it to be applied to highly unstructured cross-country environments. By employing a linear dimensionality reduction technique, a low dimensional manifold of the raw measurement space is constructed. After the high dimensional raw sensor measurements are projected onto this manifold, they are expected to form compact clusters that can be approximated by a series of Gaussian distributions. These Gaussian distributions naturally become the 'glue' between the map topology and Bayesian inference. As such probabilistic analysis at the metric level can be conveniently applied to the topological level. In this work, probabilistic analysis is applied to the loop-closure detection problem in a large cross-country environment and satisfactory results are achieved.

- **Appearance-based loop-closing optimization**

  In this paper, an appearance-based technique is proposed to conduct loop-

closing with one single re-visiting. The map is represented as a sequence of submaps deployed in a deformable configuration. The posterior PDF of the map configuration is calculated by MAP, using both mapping information and appearance information. The MAP estimation is finally formulated as energy minimization problem and solved in a coarse-to-fine manner. A powerful but expensive global searching approach, i.e., a genetic algorithm, is first employed to search the submap configuration space. After convergence (which is not necessarily accurate), a local gradient search algorithm inspired by the active contour method is applied to further tune the configuration, achieving higher accuracy in a much more efficient manner.

By integrating the above innovations into a general mobile robot localization and mapping framework, the algorithms presented in this thesis can outperform conventional metric SLAM in following aspects:

- **Coarse-to-fine analysis**

  By employing a hierarchical representation, both the mapping and localization process can be performed in a coarse-to-fine manner. The time-consuming analysis can be first applied to the higher levels. Only when a convergence is achieved need it be applied to lower level representations, which significantly reduces the searching space.

- **Landmark-independent**

  It is known that the map topology is inherently symbolic and difficult to process in a numeric manner. A major contribution of this thesis is to use Gaus-

sian models to calculate the observational probabilities of local regions, and therefore bridge the gap between map topology representations and Bayesian probabilistic inference. The experimental results demonstrate that the proposed Eigenmap technique can robustly detect loop-closure in cross a country environment where no geometrical features are available.

- **Out of central SLAM loop**

  Different from the conventional loop-closure detection techniques which rely on feature tracking, in this inference framework, no vehicle pose prediction is necessary. In other words, it is no longer required that the vehicle's global pose or the features' global position are accurately estimated. So even if the vehicle's localization error is huge, the detection algorithm can nevertheless work properly. This is a desirable property for a tracked vehicle moving in a cross-country environment, because in this case, the localization error could be quite big after a short distance. With large localization error, it is impossible to use any 'gating' mechanism to determine whether the vehicle has re-visited a certain place.

- **Handling large environment**

  By exploiting the global appearance as an additional source of information, the presented algorithm can drastically reduce the time necessary for map convergence. This is an extremely desirable characteristic in large scale SLAM applications, where the robot's trajectory could be as long as a few kilometers. Meanwhile, the presented loop-closing algorithm makes almost

no assumption about the input measurements, so it can be applied to various SLAM problems without any major modifications as long as measurements can be represented as a point cloud.

- **Efficiency and robustness**

  The two-stage loop-closing optimization approach presented in this thesis makes a reasonable compromise between the algorithm's efficiency and robustness. Robustness is assured by the genetic algorithm based global searching strategy, while the local gradient optimization reduces the total computational time and makes the algorithm efficient. Additionally, the presented two stage process is highly scalable. By tuning a few parameters of the GA and the active submap deformation, the computational time can be adjusted according to the usability of the input information.

5.2 Recommendations

This thesis has presented a comprehensive, self-contained framework for mobile robot localization and mapping in large unstructured outdoor environments. The author also wants to propose some potential yet promising directions for future research.

- **Nonlinear dimensionality reduction**

  This thesis introduced an innovative 'feature extraction' approach which uses the linear dimensionality reduction technique to extract the useful information from raw sensor data. Although powerful, such a strategy nevertheless

has its own limitation: the manifold where the data actually exist may not be linear. Consequently, PCA and its derivatives may not fully handle the distribution of the data. A possible remedy is to employ non-linear dimensionality reduction techniques such as ISOMAP[91].

- **Viewpoint invariance**

  Currently, a limitation of the proposed appearance model is that it cannot handle viewpoint invariance. If the vehicle revisits a place coming from a different direction, it is impossible for the appearance-based technique to detect the loop-closure. In the pattern recognition community, one way to solve this problem is to build a more comprehensive training pool which can teach the algorithm the scenes of the same place from different perspectives, such as trace transform [76].

- **UGV-UAV collaborative mapping**

  The proposed loop-closing optimization with appearance information can be conveniently extended to the collaborative mapping between unmanned ground vehicle (UGV) and an unmanned aerial vehicles (UAV). Currently, the performance of LCAI is still limited by the availability of comprehensive appearance information. In terms of global appearance information, the best scenario of conducting LCAI is the collaboration between UGV and UAV. The UAV can provide real-time color images which exactly represent the environment, and the ground vehicle may then enrich the image with local 3D models. On the other hand, the UGV can also use the rich pictorial

information collected by the UAV to ensure the map's consistency.

Author's Publications

Journal Paper

1. C. Chen and H. Wang, "Appearance-based topological Bayesian inference for loop closing detection in cross-country environment," *International Journal of Robotics Research* (**IJRR**), 25(10):953–983, 2006.

2. C. Chen and H.Wang, "Active submap configuration: Large-scale mapping using appearance prior," *International Journal of Robotics and Automation*, 22(2), 2007.

Conference Paper

1. Cheng Chen, Han Wang, "Large scale loop-closing with pictorial matching," in *International Conference on Robotics and Automation* (**ICRA**), pp. 1194–1199, May 2006.

2. Cheng Chen, Han Wang, "Loop-closing detection: an appearance-based topological Bayesian inference framework," in *International Conference on Intelligent Robots and Systems* (**IROS**), pp. 322–327, Aug 2005.

3. Ng Teck Chew and Shen Jian and Gong Ziming and Guzman Javier Ibanez and Chen Cheng, "Vehicle following with obstacle avoidance capabilities in

natural environments," in *International Conference on Robotics and Automation* (**ICRA**), pp. 4283–4288, Aug 2004.

4. C. Chen and H. Wang, "Adaptive pose and location estimation for indoor robot," in *International Conference on Intelligent Transportation System* (**ITSC**), pp. 834–837, Aug 2003.

Bibliography

[1] Kai Oliver Arras, Jos Castellanos, Martin Schilt, and Roland Siegwart. Feature-based multi-hypothesis localization and tracking using geometric constraints. *Robotics and Autonomous Systems*, 44:41–53, 2003.

[2] Matej Artac, Matjaz Jogan, and Ales Leonardis. Mobile robot localization using an incremental eigenspace model. In *Proc. Int'l. Conf. Robotics and Automation*, pages 1025–1030, 2002.

[3] Yaakov. Bar-shalom and Thomas. E. Fortmann. *Tracking and data asscia-tion.* Academic Press, London, 1988.

[4] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

[5] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 48:259–302, 1986.

[6] Paul J. Besl and Neil D. McKay. A method for registration of 3d shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2:239–256, 1992.

[7] Michael Carsten Bosse. *Atlas, a framework for large scale automated mapping and localization.* PhD thesis, Massachusetts Institute of Technology, Febrary 2004.

[8] Michael Carsten Bosse, Paul M. Newman, John Leonard, M. Soika, W. Feiten, and S. Teller. An Atlas framework for scalable mapping. In *Proc. Int'l. Conf. Robotics and Automation* , pages 1899–1906, 2003.

[9] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[10] John Canny. A computational approach to edge detection. *IEEE Trans. Pattern analysis and machine intelligence*, 8(6):679–698, 1986.

[11] Raja Chatila and Jean Paul Laumond. Position referencing and consistent world modelling for mobile robots. In *Proc. Int'l. Conf. Robotics and Automation*, 1985.

[12] Ng Teck Chew, Shen Jian, Gong Ziming, Javier Ibañez-Guzmán, and Chen Cheng. Vehicle following with obstacle avoidance capabilities in natural environments. In *Proc. Int'l. Conf. Robotics and Automation*, volume 5, pages 4283–4288, 2004.

[13] Haili Chui and Anand Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image understanding*, 89:114–141, 2003.

[14] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. Pattern analysis and machine intelligence*, 23(6):681–685, 2001.

[15] Ingemar J. Cox and Sunita L. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. Pattern analysis and machine intelligence*, 18(2):138–150, Feburary 1996.

[16] Andrew D. J. Cross and Edwin R. Hancock. Graph matching with a dual-step em algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(11):1236–1253, 1998.

[17] James L. Crowley, Frank Wallner, and Berent Schiele. Position estimation using principal components of range data. In *Proc. Int'l. Conf. Robotics and Automation* , pages 3121–3128, 1998.

[18] Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati. Detecting moving objets, ghosts, and shadows in video streams. *IEEE Trans. Pattern analysis and machine intelligence*, 25(10):1337–1342, 2003.

[19] Frank Dellaert, Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Using the CONDENSATION algorithm for robust, vision-based mobile robot localization. In *Proc. IEEE Conf. Computer Vision and pattern Recognition*, volume 2, pages 588–594, 1999.

[20] M. W. M. Gamini Dissanayake, Paul M. Newman, Steven Clark, Hugh F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building problem. *IEEE Trans. Robotics and Automation*, 17(3):229–241, 2001.

[21] Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequencial Monte Carlo Methods in Practice.* Springer-Verlag, New York, 2001.

[22] Tom Duckett. A genetic algorithm for simultaneous localization and mapping. In *Proc. Int'l. Conf. Robotics and Automation*, pages 434–439, May 2003.

[23] Tom Duckett, S. Marsland, and J. Shapiro. Learning globally consistent maps by relaxation. In *Proc. Int'l. Conf. Robotics and Automation*, pages 3841–3846, 2000.

[24] Tom Duckett, Stephen Marsland, and Jonathan Shapiro. Fast, on-line learning of globally consistent maps. *Autonomous Robots*, 12:287–300, 2002.

[25] Tom Duckett and Ulrich Nehmzow. Mobile robot self-localization using occupancy histograms and a mixture of Gaussian location hypotheses. *Robotics and Autonomous Systems*, 34:117–129, 2001.

[26] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* John Wiley & Sons, New York, 2000.

[27] A. Elfes. Occupancy grids: a probabilistic framework for robot perception and navigation. *Journal of Robotics and Automation*, 3:249–265, 1987.

[28] Carlos Estrada, José Neira, and Juan D. Tardós. Hierarchical SLAM: Real-time accurate mapping of large environments. *IEEE Trans. Robotics*, 21(4):588–596, August 2005.

[29] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[30] Dieter Fox. Adapting the sample size in particle filters through KLD-sampling. *International Journal of Robotics Research*, 22(12):985–1003, 2003.

[31] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Active Markov localization for mobile robots. *Robotics and Autonomous Systems*, 25:195–207, 1998.

[32] Udo Frese, Per Larsson, and Tom Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Trans. Robotics*, 21(2):196–207, April 2005.

[33] Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 4:377–388, 1996.

[34] Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, and Eric Mjolsness. New algorithms for 2d and 3d point matching pose estimation and correspondence. *Pattern Recognition*, 31:1019–1031, 1998.

[35] Venu Govindu and Chandra Shekhar. Alignment using distributions of local geometric properties. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21:1031–1043, 1999.

[36] José E. Guivant and Eduardo Mario Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Trans. Robotics and Automation*, 17(3):242–257, 2001.

[37] Jens-Steffen Gutmann and Kurt Konolige. Incremental mapping of large cyclic environments. In *Proc. Conf. Intelligent Robots and Applications (CIRA)*, 1999.

[38] Dirk Hähnel, Wolfram Burgard, Dieter Fox, and Sebastian Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Proc. Int'l. Conf. Intelligent Robots and Systems* , pages 206–211, 2003.

[39] Dirk Hahnel, D. Schulz, and Wolfram Burgard. Mapping with mobile robots in populated environments. In *Proc. Int'l. Conf. Intelligent Robots and Systems* , 2002.

[40] Andrew Howard, Denis F. Wolf, and Gaurav S. Sukhatme. Towards 3d mapping in large urban environments. In *Proc. Int'l. Conf. Intelligent Robots and Systems* , pages 419–424, 2004.

[41] Carine Hue, Jean-Pierre Le Cadre, and PatrickPerez. Sequential Monte Carlo methods for multiple target tracking and data association. *IEEE Trans. Signal Processing*, 50:309–325, 2002.

[42] Luca Iocchia and Daniele Nardi. Hough localization for mobile robots in polygonal environments. *Robotics and Autonomous Systems*, 40:43–58, 2002.

[43] H. Ishiguro and S. Tsuji. Image-based memory of environment. In *Proc. Int'l. Conf. Intelligent Robots and Systems*, pages 634–639, 1996.

[44] Patric Jensfelt and Henrik Christensen. Pose tracking using laser scanning and minimalistic environmental models. *IEEE Trans. Robotics and Automation*, 17(2):138–147, April 2001.

[45] Patric Jensfelt and Steen Kristensen. Active global localisation for a mobile robot using multiple hypothesis tracking. *IEEE Trans. Robotics and Automation*, 17(5):748–760, oct 2001.

[46] Patric Jensfelt, Olle Wijk, David J. Austin, and Magnus Andersson. Experiments on augmenting condensation for mobile robot localization. In *Proc. Int'l. Conf. Robotics and Automation*, pages 2518–2524, 2000.

[47] R. Jonker and A. Volgenant Amsterdarn. A shortest augmening path algorithm for dense and spare linear assignment problems. *Computing*, 38:325–340, 1987.

[48] S. J. Julier, and J. K. Uhlmann. The Scaled Unscented Transformation. In *Proc. the IEEE American Control Conference*, pages 4555–4559, 8–10 May 2002.

[49] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[50] S. Kirkpatrick, C. D. Jr. Gerlatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[51] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts. *IEEE Trans. Pattern analysis and machine intelligence*, 26(2):147–159, 2004.

[52] Kurt Konolige and Ken Chou. Markov Localization using correlation. In *Proc. the Int'l. Conf. Artificial Intelligence (IJCAI)*, pages 1154–1159, 1999.

[53] Kurt Konolige, Dieter Fox, Benson Limketkai, Jonathan Ko, and Benjamin Stewart. Map merging for distributed robot navigation. In *Proc. Int'l. Conf. Intelligent Robots and Systems*, 2003.

[54] B. J. A. Krose, N. Vlassis, R. Bunschoten, and Y. Motomura. A probabilistic model for appearance based robot localization. *Image and vision computing*, 19:381–391, 2001.

[55] Benjamin Kuipers and Patrick Beeson. Bootstrap learning for place recognition. In *Proc. National Conf. Artificial Intelligence(AAAI)*, pages 2512–2517, 2002.

[56] Suresh Kumar, Jose Guivant, and Hugh Durrant-Whyte. Informative representations of unstructured environment. In *Proc. Int'l. Conf. Robotics and Automation*, pages 212–217, 2004.

[57] Suresh Kumar, Fabio Ramos, Ben Upcroft, and Hugh Durrant Whyte. A statistical framework for natural feature representation. In *Proc. Int'l. Conf. Intelligent Robots and Systems*, pages 1–6, Aug 2005.

[58] Cody Kwok, Dieter Fox, and Marina Meilă. Real-time particle filters. *Proceedings of IEEE*, 92(3):469–484, 2004.

[59] Pierre Lamon, Illah Nourbakhsh, Bjorn Jensen, and Roland Siegwart. Deriving and matching image fingerprint sequences for mobile robot localization. In *Proc. Int'l. Conf. Robotics and Automation*, pages 1609–1614, May 2001.

[60] Dar Shyang Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Trans. Pattern analysis and machine intelligence*, 27(5):827–832, 2005.

[61] John J. Leonard and Hugh Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. Robotics and Automation*, 7(3):376–382, June 1991.

[62] John. J. Leonard, Hugh F. Durrant-Whyte, and Ingemar .J.Cox. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, 11(4):286–298, 1992.

[63] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

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

[65] Raj Madhavan and Hugh Durrant -Whyte. Natural landmark-based autonomous vehicle navigation. *Robotics and Autonomous Systems*, 46:79–95, 2004.

[66] Aleix M. Martinez and Avinash C. Kak. PCA versus LDA. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[67] Joseph Modayil, Patrick Beeson, and Benjamin Kuipers. Using the topological skeleton for scalable global metrical map-building. In *Proc. Int'l. Conf. Intelligent Robots and Systems* , pages 1530–1536, 2004.

[68] M. Montemerlo and S. Thrun. Large-scale robotics 3-d mapping of urban structures. In *Proc. Int'l. Symp. on Experimental Robotics (ISER)*, pages 702–718, 2004.

[69] Michael Montemerlo and Sebastian Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. In *Proc. Int'l. Conf. Robotics and Automation* , 2003.

[70] H. Moravec. Towards a general theory of topological maps. *Artificial Intelligence*, 152:47–104, 2004.

[71] Philippe Moutarlier and Raja Chatila. Stochastic multisensory data fusion for mobile robot location and environment modelling. In *Proc. Int'l. Symp. on Robotics Research*, 1989.

[72] José Neira and Juan D. Tardós. Data association in stochastic mapping using the joint compatibility test. *International Journal of Robotics Research*, 17:890–897, 6 2001.

[73] Juan Nieto, Tim Bailey, and Eduardo Nebot. Scan-SLAM: Combining EKF-SLAM and scan correlation. In *Proc. Int'l Conf. Field and Service Robotics (FSR 05)*, 2005.

[74] Sang Min Oh, Sarah Tariq, Bruce N. Walker, and Frank Dellaert. Map based priors for localization. In *Proc. Int'l. Conf. Intelligent Robots and Systems*, pages 2179–2184, 2004.

[75] Mourad Oussalah. Suboptimal multiple model filter for mobile robot localization. *International Journal of Robotics Research*, 20(12):977–989, 2001.

[76] Maria Petrou and Alexander Kadyrov. Affine invariant features from the trace transform. *IEEE Trans. Pattern analysis and machine intelligence*, 26(1):30–44, January 2004.

[77] Josep M. Porta and Ben J. A. Krose. Appearance-based concurrent map building and localization using a multi-hypotheses tracker. In *Proc. Int'l. Conf. Intelligent Robots and Systems*, pages 3424–3429, 2004.

[78] Josep M. Porta, Bas Terwijn, and Ben J. A. Krose. Efficient entropy-based action selection for appearance-based robot localization. In *Proc. Int'l. Conf. Robotics and Automation*, pages 2842–2847, September 2003.

[79] Josep M Porta, Jakob Verbeek, and Ben Krose. Active appearance-based robot localization using stereo vision. *Autonomous Robots*, 18:59–80, 2005.

[80] Ananth Ranganathan and Frank Dellaert. Inference in the space of topological maps: an MCMC-based approach. In *Proc. Int'l. Conf. Intelligent Robots and Systems* , pages 1518–1523, 2004.

[81] Donald Reid. An algorithm for tracking multiple targets. *IEEE Trans. Automatic Control*, 24:843–854, 1979.

[82] Marie Rochery, Ian H. Jermyn, and Josiane Zerubia. Higher order active contours. *International Journal of Computer Vision*, 69(1):27–42, 2006.

[83] Thomas Roefer. Using histogram correlation to create consistent laser scan maps. In *Proc. Int'l. Conf. Intelligent Robots and Systems* , pages 625–630, 2002.

[84] S. Sukkarieh, E. M. Nebot, and H. F. Durrant-Whyte. A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation*, 15:572–578, June 1999.

[85] Bernhard Sch olkopf and Alexander J. Smola. *Learning with kernels.* The MIT Press, Cambridge, Massachusetts, 2002.

[86] Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B. Cremers. People tracking with a mobile robot using sample-based Joint Probabilistic Data Association Filters. *International Journal of Robotics Research*, 22(2):99–116, 2003.

[87] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. J. Cox and G. T. Wilfong, editors, *Autonomous robot vehicles*, pages 167–193. Springer Verlag, New York, 1988.

[88] Benjamin Stewart, Jonathan Ko, Dieter Fox, and Kurt Konolige. The revisiting problem in mobile robot map building: A hierarchical Bayesian approach. In *Proc. the Conf. Uncertainty in AI (UAI)*, 2003.

[89] Omar Tahri and Francois Chaumette. Complex objects pose estimation based on image moment invariants. In *Proc. Int'l. Conf. Robotics and Automation*, 2005.

[90] Juan D. Tardós, José Neira, Paul M. Newman, and John J. Leonard. Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotics Research*, 21(4):311–330, 2002.

[91] Joshua. B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[92] Sebastian Thrun. A probabilistic on-line mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20:335–363, 5 2001.

[93] Sebastian Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann, 2002.

[94] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics.* MIT Press, 2005.

[95] Sebastian Thrun, Dieter Fox, and Wolfram Burgard. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128:99–141, 2000.

[96] Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8):693–716, 2004.

[97] Kinh Tieu and Paul Viola. Boosting image retrieval. *International Journal of Computer Vision*, 56(1-2):17–36, 2004.

[98] Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, 61(3):611–622, Series B.

[99] Nicola Tomatis, Illah Nourbakhsh, and Roland Siegwart. Hybrid simultaneous localization and map building: closing the loop with multi-hypotheses tracking. In *Proc. Int'l. Conf. Robotics and Automation*, pages 2749–2754, May 2002.

[100] Nicola Tomatis, Illah Nourbakhsh, and Roland Siegwart. Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems*, 44:3–14, 2003.

[101] W. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, pages 167–193. Springer Verlag, LNCS, 2000.

[102] Matthew Turk and Alex Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conf. Computer Vision and pattern Recognition*, pages 586–591, 1991.

[103] Iwan Ulrich and Illah Nourbakhsh. Appearance-based place recognition for topological localization. In *Proc. Int'l. Conf. Robotics and Automation*, pages 1023–1029, 2000.

[104] Michael Veeck and Wolfram Burgard. Learning polyline maps from range scan data acquired with mobile robots. In *Proc. Int'l. Conf. Intelligent Robots and Systems* , pages 1065–1070, 2004.

[105] C. J. Veenman, M. J. T. Reinders, and E. Backer. Establishing motion correspondence using extended temporal scope. *Artificial Intelligence*, 145:227–243, 2003.

[106] Cor J. Veenman, Marcel J. T. Reinders, and Eric Backer. Resolving motion correspondence for densely moving points. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(2):54–72, 2001.

[107] Nikos Vlassis and Ben Krose. Robot environment modeling via principal component regression. In *Proc. Int'l. Conf. Intelligent Robots and Systems* , pages 677–672, 1999.

[108] Chieh Chih Wang. *Simultaneous localization, mapping and moving object tracking.* Ph.d thesis, Carnegie Mellon University, April 2004.

[109] Chieh-Chih Wang, Charles Thorpe, and Sebastian Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas. In *Proc. Int'l. Conf. Robotics and Automation*, pages 842–849, September 2003.

[110] W. S. Wijesoma, K. W. Lee, and Javier Ibanez Guzmán. Motion constrained simultaneous localization and mapping in neighborhood environments. In *Proc. Int'l. Conf. Robotics and Automation*, pages 344–349, May 2005.

[111] Chenyang Xu and Jerry L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Processing*, 7(3):359–369, March 1998.

[112] H. Zho and R. Shibasaki. Reconstructing urban 3D model using vehicle-borne laser range scanners. In *Proc. 3rd Int'l. Conf. on 3-D Digital Imaging and Modeling*, pages 349–356, 2001.

[113] Chao Zhou, Yucheng Wei, and Tieniu Tan. Mobile robot self-localization based on global visual appearance features. In *Proc. Int'l. Conf. Robotics and Automation*, pages 1271–1276, 2003.