

SHAPE REGISTRATION USING OPTIMIZATION FOR MOBILE ROBOT NAVIGATION

by

Feng Lu



A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

© Copyright by Feng Lu 1995

Shape Registration Using Optimization For Mobile Robot Navigation

Doctor of Philosophy, 1995

Feng Lu

Department of Computer Science

Univeristy of Toronto

Abstract

The theme of this thesis is shape registration (also called shape matching or shape alignment) using optimization-based algorithms. We primarily address this problem in the context of solving the mobile robot self-localization problem in unknown environments. Here the task is matching 2D laser range scans of the environment to derive the relative position and heading of the robot. The difficulties in this problem are that the scans are noisy, discontinuous, not necessarily linear, and two scans taken at different positions may not completely overlap because of occlusion. We propose two iterative scan matching algorithms which do not require feature extraction or segmentation. Experiments demonstrate that the algorithms are effective in solving the scan matching problem.

Based on the result of aligning pairwise scans, we then study the optimal registration and integration of multiple range scans for mapping an unknown environment. Here the issue of maintaining consistency in the integrated model is specifically raised. We address this issue by maintaining individual local frames of data and a network of uncertain spatial relations among data frames. We then formulate an optimal procedure to combine all available spatial relations to resolve possible map inconsistency. Two types of sensor data, odometry and range measurements, are used jointly to form uncertain spatial relations.

Besides the applications for mobile robots, we also study the shape registration problem in other domains. Particularly, we apply extensions of our methods for registration of 3D surfaces described by range images, and 2D shapes from intensity images.

Acknowledgements

First of all, I would like to thank my supervisor, Evangelos Milios, for his continued inspiration, guidance, encouragement, and support throughout my studies. He has been conscientious in his attention to every aspect of my professional development. I owe him enormous thanks.

I thank the other members of my committee, Michael Jenkin, Allan Jepson, John Tsotsos, and my external examiner, Ingemar Cox, for the time and thought they have put into hearing my ideas and reading my thesis. Their constructive criticism has greatly improved the thesis.

I am grateful to Piotr Jasiobedzki for his advice and help on my research work and also for bringing me into the retinal image analysis project.

I benefited from the discussions with the fellow graduate students in the Vision/AI group and I am also grateful for their helps and friendship. In particular, I would like to thank Xuan Ju, Victor Lee, Yiming Ye, Richard Mann, Sean Culhane, Zhengyan Wang, Niels Lobo, David Wilkes, Chris Williams, and Evan Steeg.

I thank Songnian Zhou for his generous help and encouragement in my early years at the University of Toronto.

Finally, I thank my wife for her love and sacrifices and my parents for their moral support.

Contents

1	Introduction	1
1.1	Shape Registration	1
1.2	Applications of Shape Registration	2
1.3	Mobile Robot Navigation	3
1.3.1	Robot Pose Estimation	4
1.3.2	Building a World Model from Sensor Data	5
1.4	Thesis Outline	6
I	Review of Related Work	8
2	Review of Related Work	10
2.1	Mobile Robot Navigation	10
2.1.1	Issues in Robot Navigation	10
2.1.2	Pose Estimation	13
2.2	Matching Sensor Data for Pose Estimation	16
2.2.1	Correlation Methods	17
2.2.2	Combinatorial Search	17
2.2.3	Statistical Data Association and Estimation	18
2.2.4	Iterative Optimization Methods	20
2.2.5	Our Work	21
2.3	World Models for Robot Navigation	22
2.3.1	Types of World Model	22
2.3.2	Dynamic World Modeling	23
2.3.3	Consistency in Dynamic World Models	26

2.3.4	Our Work	29
2.4	Shape Registration	29
2.4.1	Model-based Recognition	30
2.4.2	Matching Planar Shapes	31
2.4.3	Registration of 3D Shapes	32
2.4.4	Our Work	34
II	Matching Range Scans for Robot Pose Estimation	35
3	A Search/Least-Squares Algorithm for Matching Scans	37
3.1	Problem Definition	37
3.1.1	Pose Estimation by Aligning Scans	37
3.1.2	Criterion for Matching Scans	38
3.1.3	Iterative Scan Matching Algorithms	38
3.2	Search/Least-Squares Matching Algorithm	39
3.3	Projection of Reference Scan	40
3.4	Fitting Tangent Lines	41
3.5	Correspondence Definition	43
3.6	Correspondence Search	46
3.7	Optimization	46
3.8	Discussion	48
4	Point Correspondence Based Matching Algorithm	50
4.1	General Approach	50
4.2	Rules for Correspondence	51
4.2.1	Closest-Point Rule	51
4.2.2	Matching-Range-Point Rule	52
4.2.3	Combining the Two Rules	54
4.3	Matching Scans	55
4.4	Detecting Outliers	57
4.5	Convergence of Iterative Algorithm	57
4.6	Discussion	58

5	Scan Matching Experiments	59
5.1	Combining The Two Algorithms	59
5.2	Sensing Strategy	60
5.3	Experiments with Simulated Data	60
5.4	Experiments with Real Data	68
5.4.1	Experiments Using The ARK Robot	68
5.4.2	Experiments at FAW, Germany	69
5.5	Comparison with Other Methods	76
III	Consistent Registration of Multiple Range Scans for Robot Mapping	81
6	Consistent Registration Procedure	83
6.1	Problem Description and Approach	83
6.1.1	Pairwise Scan Matching	84
6.1.2	Constructing a Network of Pose Constraints	85
6.1.3	Combining Pose Constraints in a Network	85
6.2	Optimal Estimation from a Network of Constraints	86
6.2.1	Problem Definition	86
6.2.2	Solution of Optimal Linear Estimation	87
6.2.3	Special Networks	89
6.2.4	Analogy to Electrical Circuits	90
6.3	Formulation of Constraints for Robot Pose Estimation	92
6.3.1	Pose Compounding Operation	92
6.3.2	Pose Constraint from Matched Scans	94
6.3.3	Pose Constraints from Odometry	96
6.4	Optimal Pose Estimation	98
6.5	Algorithm Implementation	98
6.5.1	Estimation Procedure	98
6.5.2	Iterative Process	99
6.6	Discussion	99

7	Experiments and Alternative Approaches	101
7.1	Introduction	101
7.2	Experiments with Simulated Measurements	101
7.3	Experiments with Real Data	106
7.4	An Alternative Solution without Linear Approximation	108
7.5	Sequential Estimation	109
7.6	Discussion	111
IV	Other Shape Registration Applications	112
8	Registration of Range Images	114
8.1	Introduction	114
8.2	Point Correspondence Based Methods	114
8.2.1	Least-Squares Solution	115
8.2.2	Iterative Algorithms	116
8.3	Methods Using Tangent Information	119
8.3.1	Algorithm Description	119
8.3.2	Solution from Minimizing Distance Function	120
8.3.3	Experiments	123
8.3.4	Rotation Search Method	124
9	Registration of Planar Image Shapes	128
9.1	Introduction	128
9.2	Rotation Search/Least-Squares Method	129
9.2.1	Problem Formulation	129
9.2.2	Correspondence Definition and Least-Squares Matching Distance	130
9.2.3	Curve Model and Correspondence	132
9.2.4	Search Algorithm	133
9.2.5	Experimental Results	134
9.3	Iterative Point Correspondence Based Algorithm	136
9.3.1	Iterative Least-Squares Procedure	136
9.3.2	Correspondence Rules and Algorithms	141

9.3.3 Experiments of Planar Shapes Registration	142
9.4 Discussion	144
10 Conclusion	148
10.1 Summary of Major Results	148
10.2 Directions for Future Research	149
A Error from Approximate Point Correspondence	151
B Derivation of Identity	153

Chapter 1

Introduction

1.1 Shape Registration

To put it simply, shape registration is to align two similar shapes, typically a data (object) shape and a model shape, by moving one around with respect to the other. We consider that two shapes are in alignment if they spatially conform with each other to achieve a minimum distance according to some distance criterion. For example, Fig. 1.1(a) shows two similar shapes misaligned with each other. A possible registration result is given in Fig. 1.1(b) where we rotated and translated one shape to make it aligned with the other shape. The type of shapes we consider here are contour curves (in 2D) and surfaces (in 3D). In this thesis, we are more concerned about the geometric alignment of shapes rather than the determination of whether the given shapes are similar, although we may obtain a measure of similarity as a by-product. Under this context, we use the terms shape registration, shape alignment, and shape matching almost interchangeably.

A typical approach to shape registration is to use distinctive features. For example, in aligning the triangles in Fig. 1.1, we may locate the corner points or the linear segments in the shapes and use them as features. By making correspondences between pairs of features on the two shapes, a registration can be determined.

Feature-based methods may be inappropriate for matching smooth free-form shapes where it is difficult to detect distinctive features reliably. Consider the example in Fig. 1.2 and imagine how a human would align these shapes. One may draw the two shapes on separate transparencies and visually align them. After overlapping the two transparencies containing the shapes, one can try to make a series of small adjustments to the relative position and orientation to improve the alignment, and finally derive an optimal registration. This intuitive process can be achieved computationally

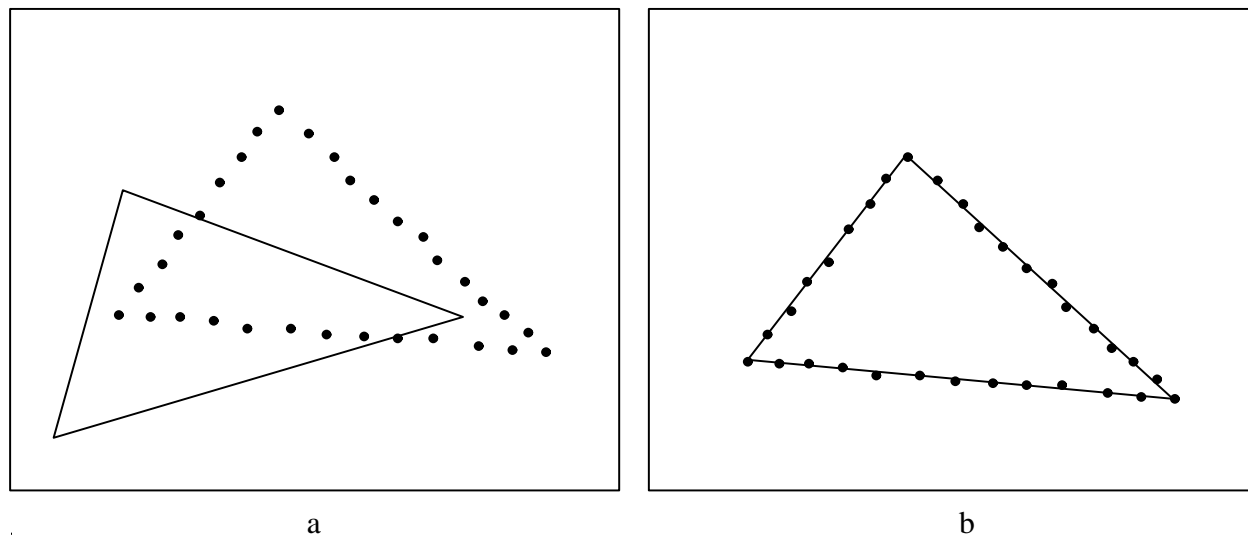


Figure 1.1: Shape registration. Corners or linear segments can be used as features to align these shapes.

using iterative optimization techniques. In this thesis, we will pay close attention to this type of optimization-based matching methods.

1.2 Applications of Shape Registration

Shape registration is an important problem in computer vision, especially in object recognition, motion analysis, and robot navigation.

There are many applications for shape registration. For instance, in the medical and surgical fields, various imaging sensors can provide specific information for a patient (e.g. Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and 3D Ultrasound images). There is a real need to register all of these 3D images in the same reference system, and to then link these images with the operating instruments such as guiding systems or robots [76]. To achieve this goal, one possibility is to use some anatomical surfaces as references in all these images. In some cases, it becomes necessary to use a laser range finder to acquire the skin surface of a patient, and then to register this reference surface with the skin surface segmented on another imaging device.

In the industrial world, it is often required to inspect 3D parts to determine if they agree with the geometric design specifications. This can be achieved by digitizing the 3D object using a high-accuracy laser range finder over a shallow depth of field and matching the data with an

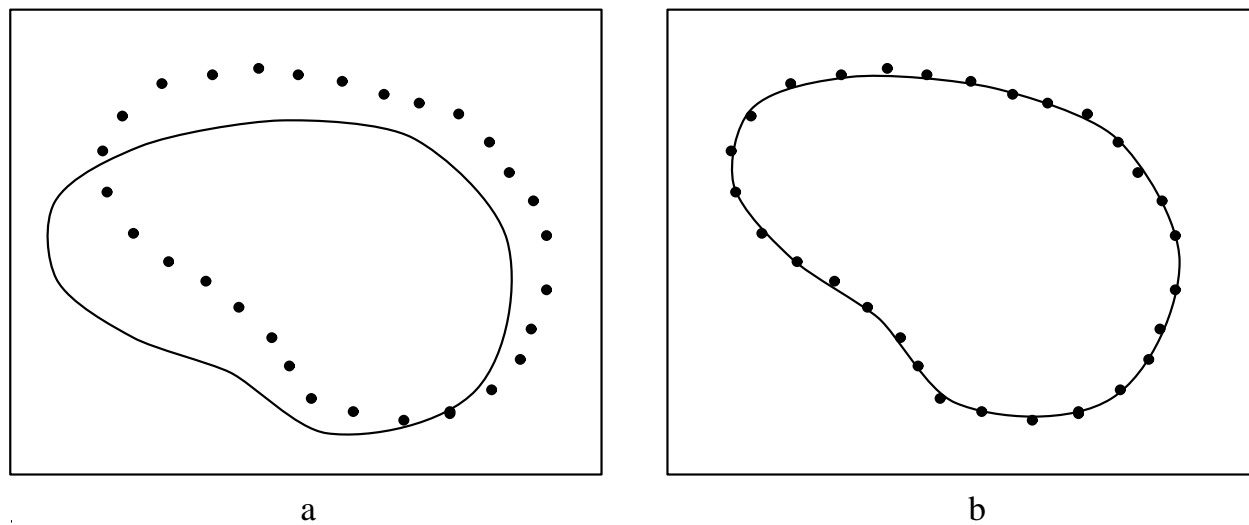


Figure 1.2: The smooth free-form shapes in (a) can be registered by making a series of small adjustments to the relative position and orientation. Part (b) gives a possible registration result.

ideal CAD model. Furthermore, it is also important to build integrated models from existing 3D objects. An automatic model acquisition system could significantly improve the speed and flexibility compared to conventional interactive techniques such as computer-aided geometric design (CAGD) and coordinate-measurement machines (CMM). Since a range map only samples the surface of the object which is visible from a given viewpoint, the acquisition of several range views is mandatory in order to scan the entire object. Therefore, to obtain an integrated model, it is necessary to transform and register all the partial views into a common reference frame.

1.3 Mobile Robot Navigation

The task of mobile robot navigation is to guide the vehicle through its world based on sensory information. The questions that a robot often faces are:

- Where am I?
- Where are other places relative to me?
- How do I get to other places from here?

We will address the first question, which concerns the robot pose estimation problem, by using laser range sensing on the robot and then matching the range measurements. We will also study

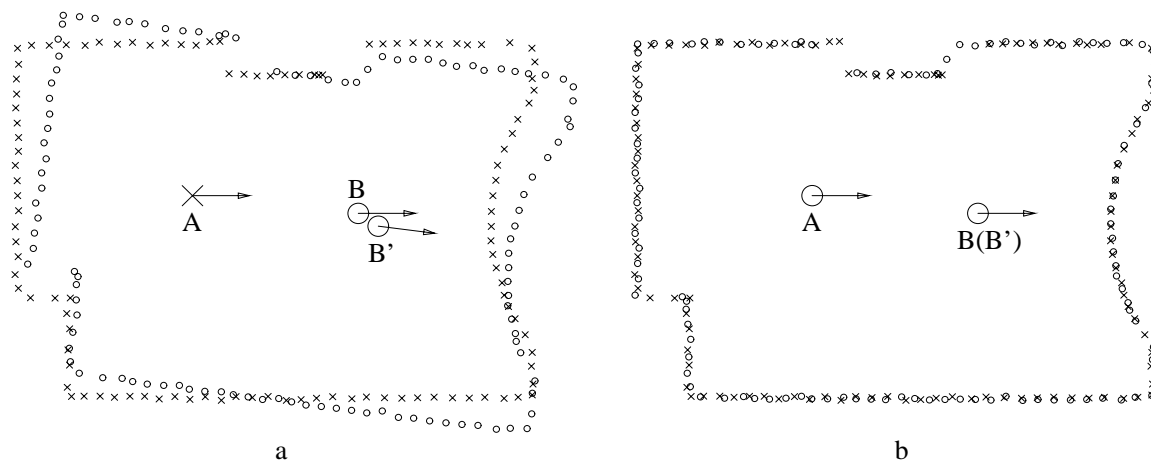


Figure 1.3: Example of robot pose estimation from matching range scans. A is a reference pose at which a reference scan (labeled as x's) is taken. The robot is now at pose B in reality, but thinks it is at pose B' due to error. Part (a) shows that the scan at B (labeled with small circles) does not align with the reference scan because of the pose error. Part (b) shows the result of aligning the two scans. The pose B' is corrected to the true pose B at the same time.

the issue of building world models which is relevant to many aspects of navigation.

1.3.1 Robot Pose Estimation

We represent the contour of the world in a horizontal plane using a 2D curve model. Suppose that the robot is equipped with a laser range finder which rotates in the same horizontal plane and takes range measurements at many directions. If these measurements are sufficiently dense, the points corresponding to the intersection points of the laser beams and the world contour essentially form the same shape as the world model. Then by aligning the shape described by the data points with the model shape, we can determine the pose of the robot in the model coordinate system.

A world model may not always be available *a priori* to the navigation task. For example, a robot exploring an unknown environment does not have such a model to reference its position. One solution is to build a world model from previously collected sensor data and then use this model to reference the robot's new position. A simple case is to just use one previous range scan as a reference. By registering the current range scan with the reference scan, we can determine the relative pose. We will study this problem in great detail in the thesis. Here we give an example to illustrate the problem (Fig. 1.3).

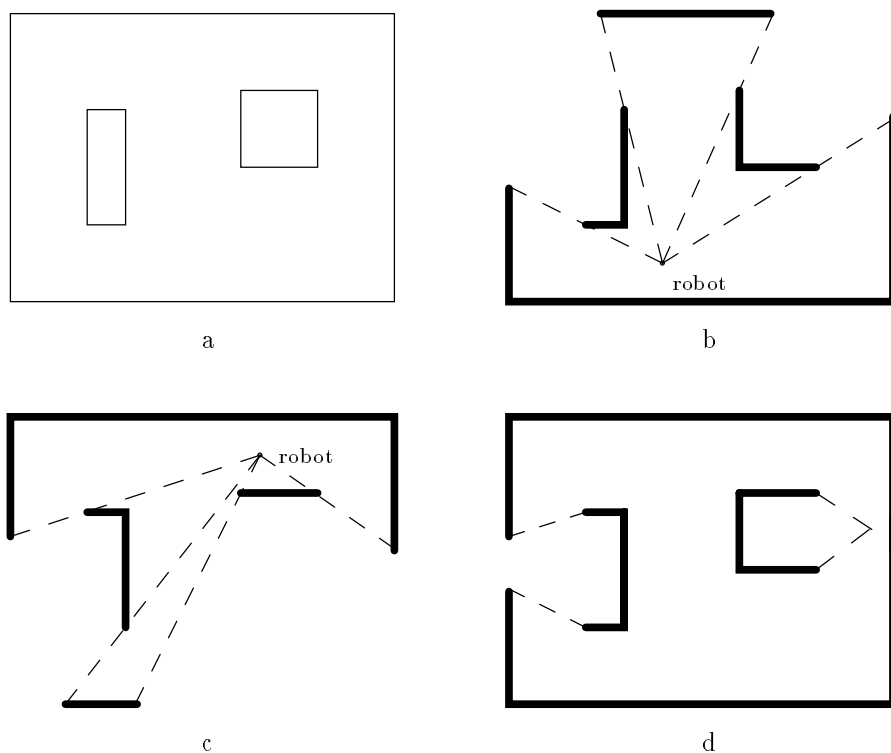


Figure 1.4: Building a world model from scans of sensor data: (a) the world being explored, (b) accumulated model so far, (c) a new scan of range data, (d) updated model.

1.3.2 Building a World Model from Sensor Data

A world model can be useful in solving problems such as path planning and robot self-localization. Acquiring a world model of an unknown environment could also be the goal of a robot's mission. A robot may build a world model by integrating successive frames of sensor data. Figure 1.4 shows an example of dynamically building a world contour map from range data.

In order to integrate multiple frames of sensor data, it is essential to first register them in a common coordinate system. A possible approach is to align each new frame of data to a previous frame or a cumulative model. But a potential problem with this approach is that the resulting model may become inconsistent as local registrations are derived independently from different parts of the model and there are errors in the registrations. Especially, if a long chain of registrations are compounded, there can be a significant error in the integrated model. Figure 1.5 illustrates this problem. Although there is only a small registration error at every step of adding new data, error

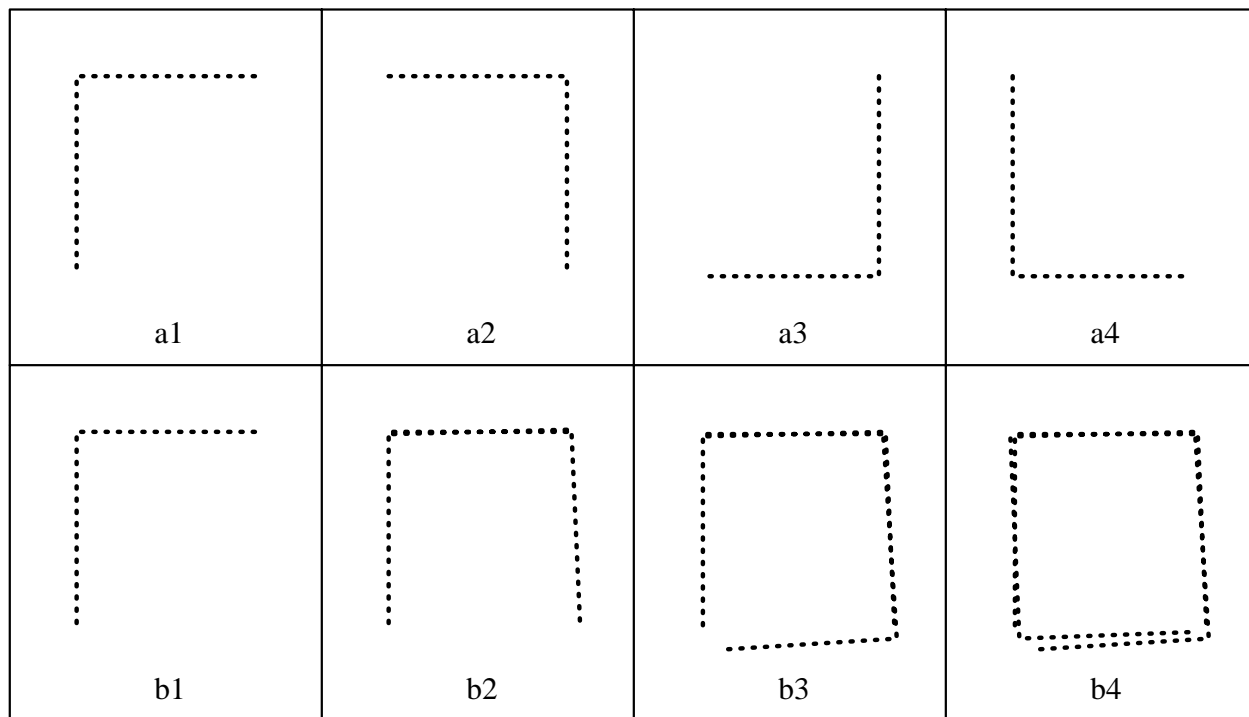


Figure 1.5: Inconsistent model due to accumulation of registration errors. (a1–a4) show local frames of sensor data from a rectangular environment. Each frame only overlaps with two adjacent frames. (b1–b4) are the cumulative models where only slight registration error is present at each step. The final model in (b4) appears to be inconsistent in the bottom part.

accumulation leads to a significant inconsistency in the final model.

In this thesis, we will study the problem of integrating multiple range scans and we will specifically raise the issue of maintaining model consistency.

1.4 Thesis Outline

The theme of the thesis is shape registration or shape matching. We first address this problem in the context of mobile robot applications. In particular, we propose algorithms for aligning 2D shapes of the environment which the robot observes in the form of range scans. Relative poses of the robot can be derived through the alignment of scans. Moreover, by aligning multiple scans, we can coherently integrate the range measurements for mapping the environment.

Besides the applications for robot navigation, we also study the shape registration problem in other domains. Particularly, we examine our methods as well as other techniques for registration

of 3D surfaces described in range images and registration of 2D shapes from intensity images.

Throughout the thesis, we focus on matching based on iterative optimization, as opposed to feature correspondence.

The rest of the thesis contains four parts. We give a brief outline below.

First of all, we present a literature review in Part I (Chapter 2). Areas related to our work, including robot navigation, pose estimation, robot map building, and shape registration techniques, will be surveyed. The significance of our own work in the context of the literature will also be pointed out.

In Part II of the thesis (Chapter 3 to Chapter 5), we study the problem of robot pose estimation in unknown environments by matching range scans. We propose two iterative algorithms which can effectively align a range scan against another scan so as to derive their relative pose. Experiments using both simulated data and real data will be presented.

In Part III (Chapters 6 and 7), we study the consistent registration and integration of multiple range scans for mapping the environment. First, pose constraints are derived from matching pairwise scans as well as from odometry sensing, which form a network of pose relations. Then we formulate an optimal estimation procedure based on the maximum likelihood criterion to combine all the pose relations and derive the scan poses. Experiments using simulated and real data will be shown.

Part IV (Chapters 8 and 9) investigates the shape registration problem in domains other than mobile robotics, namely the registration of 3D range surfaces and 2D planar image shapes. We will examine the extensions of our algorithms (from Part II) as well as other techniques.

Finally, Chapter 10 concludes the thesis with a summary and a discussion of possible future research directions.

Part I

Review of Related Work

We present a literature review in the areas related to our research work. The organization of the review and its relationship with our own work are the following.

First of all, in Section 2.1, we review the topic of mobile robot navigation and particularly, robot pose estimation. This discussion gives a big picture of the area and lays a background for the thesis since our work in Part II and Part III are related to mobile robot navigation.

In Section 2.2, we specifically examine existing techniques of matching sensor data for robot pose estimation. A substantial part of our own work (presented in Part II of the thesis) belongs in this area.

In Section 2.3, we discuss the types of world models used for robot pose estimation and the techniques for dynamically building models from sensor data. One issue in model building that we are particularly interested in is the global consistency in integrating sensor data. Our research work regarding this issue is presented in Part III of the thesis.

Finally, in Section 2.4, we review the general shape matching techniques in a broader domain. Shape matching is the theme throughout the thesis and it is the central issue in the problem of robot pose estimation using range data. Besides, we also study shape matching techniques for other applications such as 3D range surface registration and planar image shape registration (presented in Part IV of the thesis).

Chapter 2

Review of Related Work

2.1 Mobile Robot Navigation

Navigation is a fundamental requirement of autonomous mobile robots. It is defined as “the science of getting ships, aircrafts, or spacecrafts from place to place; esp: the method of determining position, course, and distance traveled” [91]. Navigation and position estimation has been extensively studied for ships, missiles, aircrafts, and spacecrafts. However, navigation for mobile robots remains a difficult problem, given the available knowledge and experience in the land, marine and aerospace communities. The reason for this is clear: it is not navigation itself that is the problem, rather it is the reliable acquisition or extraction of sensor information, and the automatic correlation or correspondence of these measurements with a navigation map that makes the autonomous navigation problem so difficult.

2.1.1 Issues in Robot Navigation

The goal of sensor-based autonomous robot navigation is to build a system which dynamically guides and controls a mobile robot from its start position to a predefined end position, while avoiding known or unknown obstacles. The robot should efficiently interpret data from its on-line sensors to determine its relationship to the world and the current state of the task, and then should plan and execute an appropriate course of action to accomplish the task.

Previous research in mobile robotics has typically addressed the following types of problems.

Path planning

First, path planning in a known environment is studied. The optimality criterion in this class of problems is the minimization of the cost for traversal between a start and an end position, while

avoiding obstacles [75]. A common approach is to represent the world as a graph and perform an A^* search or to apply some form of Dijkstra's algorithm to determine the shortest path. The visibility graph algorithm [81, 97] is an example in this class. Voronoi diagrams have been used to plan a path that stays away from obstacles as far as possible [21, 113]. Explicit representation of free space using overlapping cones called "freeways" is also proposed for path planning [18]. Another approach is to recursively subdivide free space into smaller cells until a collision-free path can be found through a connected set of free cells [20]. A multiresolution quadtree structure has been suggested [62] for the efficient representation of free space. The theoretical issue of path planning complexity with complete information (the "Piano Mover's Model") is studied in [104].

If environmental sensing is imperfect, uncertainty should be considered in path planning. A method using Sensory Uncertainty Field for planning a robot path that minimizes uncertainty at navigation time is proposed in [114] with preliminary results. A method which integrates dynamic path planning with self-localization and landmark extraction is discussed in [10].

Trajectory planning

Then, robot trajectory or motion planning in the presence of moving obstacles is studied. The goal is to find an optimal robot trajectory (consisting of both a path and the motion along the path) which avoids collision with moving obstacles. Some theoretical results about the complexity of trajectory planning can be found in [100, 22]. Some heuristic approaches for planning a collision-free path in the presence of moving obstacles are presented in [46, 47, 66].

The concept of configuration space has been widely used for solving the path planning problem. Here the original problem of planning the motion of an object through a space of obstacles is transformed into an equivalent, but simpler, problem of planning the motion of a point through a space of enlarged configuration space obstacles [81, 80]. Configuration space provides an effective framework for investigating motion planning problem for robot arm manipulators [88].

Algorithmic exploration

Another type of problem is exploration of an unknown world with perfect range sensing and odometry information [87]. Here the issues are primarily the covering of the entire environment and the complexity of the algorithm as a function of the complexity of the environment (number of vertices and edges of objects). Some research work on this class of problem can be found in [97, 88, 37, 120].

Path execution and self-localization

A practical problem which has attracted much research attention is the problem of path execution within a known or mostly known real environment. The focus here is typically on the sensing required to accurately execute a preplanned path, and the key problem is robot self-localization (the “where am I” problem). One primary issue in solving this problem is how to match sensed data (vision, sonar, laser, infrared etc.) against map information. A comprehensive survey of the literature in this area can be found in [30, 59]. We will discuss the robot self-localization problem in more detail in the following subsection.

Local obstacle avoidance

The issue of local obstacle avoidance arises when the world is not perfectly known. During the execution of a path, if an unexpected obstacle is encountered, the path needs to be modified or local strategies should be employed to avoid collision with the obstacle. A common approach is based on the artificial potential field [3, 67, 17, 2]. A heuristic approach for path replanning is used in [16].

Exploration with imperfect sensing

The problem of exploration of an unknown world with imperfect range sensing and odometry information is addressed [71]. The general approach is to gather sensing information and accumulate local measurements into a global model. Here self-localization of the robot is still an important issue. We will review some of the techniques in building world models in a later section. We will discuss a specific issue in mapping an unknown environment, that is to maintain the consistency among the sensing data as they are integrated into a cumulative global model.

System architecture

Finally, the issue of system architecture is studied for coordinating different actions of a robot system, such as path following, obstacle avoidance, exploration, etc.. Brooks proposed a subsumption architecture which is much different from the conventional centralized robot control [19]. Connell built a functioning mobile robot, Herbert, based on this subsumption architecture [27]. The ARK robot combined a classical occupancy grid based global path planner and a low-level subsumption based architecture to accomplish both path generation and path execution [101]. Another autonomous mobile robot system, AuRA, employed a motor schema-based control system which is implemented using artificial potential fields [3, 4].

2.1.2 Pose Estimation

Robot pose estimation, or self-localization, is a key issue in robot navigation. The goal of pose estimation is to keep track of the robot's position and heading direction with respect to a global reference frame. The reference frame is either defined by models of external landmarks or by the robot's initial pose. In most robot applications, only the 3D robot pose consisting of a 2D position (x, y) and a heading direction θ is considered. For more complicated tasks, the estimation of a full 6D pose may be required.

Essentially, pose estimation is done in two ways: dead reckoning and external referencing. In dead reckoning, the current robot pose (as relative to a previous pose) is measured using an internal sensor (an odometry). In external referencing, robot pose is determined with respect to external landmarks.

Pose Estimation by Dead Reckoning

Dead reckoning is done by integrating an internal sensor (e.g. odometry) over time. Odometry is typically implemented by shaft encoders on wheels to record the distance traveled or the change in heading angle. Dead reckoning is convenient and inexpensive, and it is present in most mobile robot systems. In certain man-made clean environments, odometry usually gives accurate measurements about the travel distance and turning angle over a short period. However, a serious problem with dead reckoning is that small measurement errors (due to surface roughness, wheel slippage etc.) may build up, leading to unbounded pose errors. Thus odometry alone is not sufficient for pose estimation for a navigation task. Nevertheless, odometry provides good initial pose estimates which may greatly help landmark-based pose estimation procedures.

Pose estimation using odometry is a straightforward open-looped process. Some research work on odometry configurations and their error models can be found in [118, 110]. A method for correcting systematic odometry errors is discussed in [15].

Given a previous pose and the odometry estimate of a relative pose, the current pose is computed by a compounding operation. Smith and Cheeseman [107] formulated the change of uncertainties through compounding. As a part of a Kalman filter model, a time state transition equation for the change in robot pose and its uncertainty is formulated in [41, 78].

Pose Estimation by Landmark Recognition

Because of the accumulation of pose error in odometry measurements, it is necessary to check with external references from time to time. A typical approach is to locate known landmarks in the

environment using external sensors (video camera, sonar, laser rangefinder etc.). Then compare the sensed landmarks against their representations in a metric model so as to derive a correction to the pose error.

Depending on a given task, it is also possible to navigate a robot without a metric model, i.e. to use qualitative information only. Kuipers and Levitt described a hierarchy of such approaches, namely sensorimotor, topological, and procedural approaches [72]. Here sensorimotor means elementary sensing and motor actions; topological means a network of places and routes without metric information; and procedural means local control strategies such as obstacle avoidance, route following and landmark tracking. A qualitative solution by tracking which side of landmark-defined lines the robot is on is proposed in [79]. A navigation strategy by means of path remembering is used by the Herbert robot [27]. A neural network based method is proposed in [38] for estimating robot pose from images without using explicit object models.

Navigation methods which do not use a metric model have to rely on the unambiguous recognition of landmarks which is a very difficult problem in general.

Combining dead reckoning and external sensing

An effective approach to solving the navigation problem is to combine dead reckoning with external sensing that uses a metric model. By maintaining a pose estimate, the robot can use odometry to obtain an approximate current pose. This pose estimate provides expectations and constraints for searching and data association and it greatly simplifies landmark recognition. On the other hand, the recognized landmarks provide correction to the odometry pose errors.

A typical pose estimation procedure is shown in Figure 2.1. In this procedure, first a pose estimate is obtained from odometry. The world model, together with the pose estimate, provides prediction about the current sensor observation. After taking the sensor measurements, the correspondence between the sensor data and the predicted part of the model is determined, and any discrepancy between each corresponding data-model pair is used to correct the estimate of the robot pose. The major issues of this procedure are discussed below.

Sensing and feature extraction

The commonly used sensing methods are vision and range sensing. Vision is the most flexible sensing technique. Although it only provides 2D images, various information such as depth, shape, motion can be extracted indirectly. The computer vision community has extensively studied these problems [43, 56, 64]. To interpret the image data for pose estimation, usually features are first

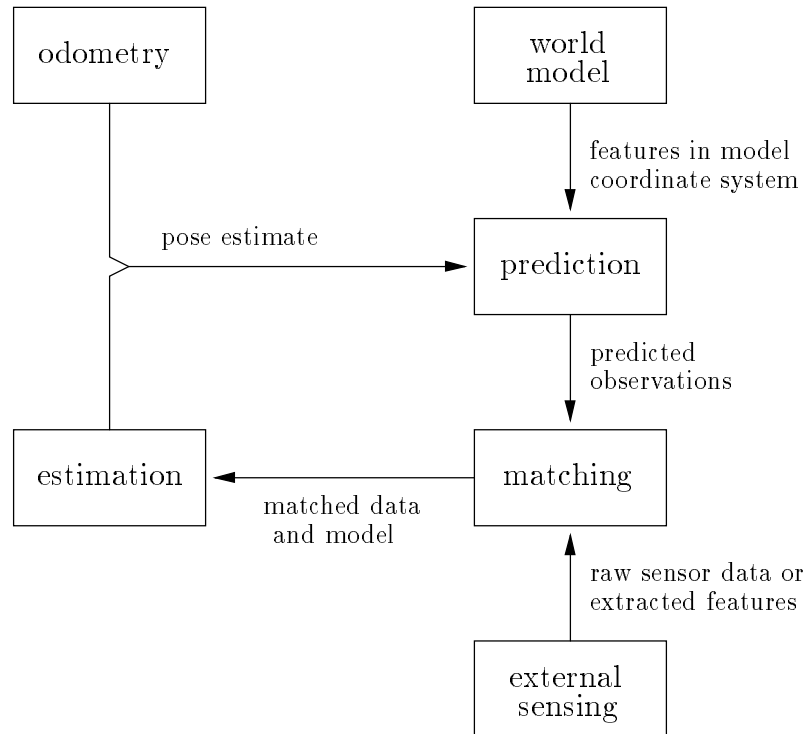


Figure 2.1: A typical procedure of pose estimation using both odometry and external sensing. First a pose estimate is obtained from odometry. The world model, together with the pose estimate, provides prediction about the current sensor observation. After taking the sensor measurements, the correspondence between the sensor data and the predicted part of the model is determined, and any discrepancy between each corresponding data-model pair is used to correct the estimate of the robot pose.

extracted. A widely used example drawn from image processing is the so-called “edge”, which is some curve in an image in which a rapid intensity change occurs. Such a feature is hypothesized to correspond to some three-dimensional entity, either a change in reflectance or a change in the surface structure. A potential problem is that feature detectors may be unreliable, and erroneous features lead to errors in pose estimation.

For robot pose estimation, range information is particularly useful and convenient to process. Commonly used range sensors include sonar and laser range-finder. Sonar is inexpensive, but its measurements are very noisy due to its wide beam width and specular reflection. Laser range-finder gives more accurate readings and its range of measurements is also larger. Typically, a single shot of a range sensor gives one depth measurement. The sensor is thus often rotated in order to take a scan or an image of measurements which gives more context of the shape of the current environment. From a scan of range data, geometric features can be extracted for matching and recognition. The commonly used features are line segments, corners, depth continuities, curvature extrema, etc.. Geometric features are relatively more stable than image features. But due to occlusion, sensor noise, and pose uncertainty, these features are still difficult to interpret.

World model for pose estimation

A metric *a priori* world model provides the reference for pose estimation. The model keeps the geometric or other properties of features for recognition and the global coordinates of the features. If an estimate of the current pose is available, predictions about the current visible features can be made (i.e. to select the relevant features from the model). We will discuss in more detail about world models in a later section.

Matching and estimation

Matching is a procedure of associating features from sensing data with the model. With the feature-model correspondence, the current robot pose can be estimated. We discuss the matching techniques in the following section.

2.2 Matching Sensor Data for Pose Estimation

The central issue in robot pose estimation is to match sensor data to a world model so as to infer the robot pose. Suppose that there is an error in the robot pose in the sense that the robot mistakenly thinks it is at one place while it is actually somewhere else. This pose error may lead to a discrepancy between the location of a data feature in a sensor coordinate system and the location

of the corresponding model of that feature (assuming a known model). By comparing the sensor data with the model, the robot can solve or correct its pose based on this discrepancy.

There are two issues involved in the matching techniques: finding the correspondence between the data and the model, and solving for the robot pose using the correspondence. The two issues are closely related to each other as one can usually be easily solved assuming the other. We discuss some matching techniques for pose estimation as the following.

2.2.1 Correlation Methods

Images (of either intensity or range) can be matched to another image by a correlation-based approach. Here one image (or a small region in the image) is placed in all possible relative positions and orientations with respect to the other image. Then one particular transformation between the two images which minimizes some sort of distance measure is selected as the matching solution. In robot navigation applications, correlation method is often used for matching grid-type world representations. For example, Elfes used a heuristically improved correlation method to match sonar maps which are represented in occupancy grids [42]. Correlating two grids usually takes extensive computation, especially when the grids are at different orientation.

Weiss *et al* used a correlation method to match range scans for keeping track robot position and orientation. They first correlate the “angle histograms” of two scans to recover the relative orientation. Then the X and Y histograms are used to compute translation.

Straightforward correlation-based matching methods are generally unable to handle outliers. For example, if some areas are visible in one scan but not in the other because of occlusion, the correlation of the two scan may produce arbitrarily bad estimation. Robust techniques which limit the influence of outliers have also been studied [13].

2.2.2 Combinatorial Search

Many matching techniques are feature-based. The typical process is to first identify a set of features in the sensor data that potentially corresponds to entities in the world, and then match the features to a world model through some form of combinatorial search. In order to limit the exponential explosion in search complexity, various constraints and heuristics are applied.

In a vision-based robot navigation system, Fennema *et al* used a heuristic search technique coupled with an objective function to match image features to a known model [44]. The features used in their approach are the 2D line segments extracted from intensity images. The objective

function is defined on each possible state of correspondence between the lines in data and lines in the model, by taking into account both the fitting error for the matched data-model pairs and the penalty for the unmatched model parts. The search strategy is to add or delete k lines from the data-model correspondence at every step, and try to search for a state with the minimum matching error. Once the correspondence is determined, the 3D camera pose is solved by a least-squares spatial fitting procedure.

The above approach is very computationally expensive. An example in the paper shows that it can take hours to perform the model matching if the search is not focussed. Many issues still need to be addressed. Such as how to determine a good starting correspondence; how to select k lines to add/delete; etc.. The value of k also plays an important role in the method. Small k (such as $k = 1$) may leads to local minima in the evaluation function. For a larger value of k , the search space increases by $O(N^k)$ where N is the number of possible pairings of model and data lines.

In addition to the high computational complexity in feature-based matching technique, another major problem is that the quality of matching depends on the reliability of features.

2.2.3 Statistical Data Association and Estimation

Probabilistic models have been used to represent measurements with uncertainty. Typically, a measurement is modeled as a random variable with assumed Gaussian distribution. Uncertainty is then represented by the variance of the random variable.

Given a pose estimate, data-model correspondence can be determined by associating each feature with the closest model. Besides, a threshold on the confidence of matching can be set based on the pose uncertainty. If a feature is too far away from the closest model (exceeding the statistical threshold), it can be rejected as an outlier. An extensive study about statistical data association can be found in [9].

A difficult decision in data association is that, if more than one feature is within the acceptance zone of a model, which one(s) should be selected? Possible strategies include selecting the closest one, selecting the average of all close ones, or not to select any feature at all in case of ambiguity. Each strategy has its own drawbacks.

In the work by Leonard and Durrant-Whyte [78], geometric features called “regions of constant depth” are extracted from a scan of sonar range data. On the other hand, predictions about possible feature occurrences are generated based on a model and an estimate of the sensor pose. Sensing errors and positional errors are modeled as additive Gaussian random noises with known

covariances. A statistical test that measures the difference between observation and prediction is applied to every pair of feature and model. Pairs satisfying the test are accepted as good matches. Other features that do not correspond to any model are discarded as outliers. A potential problem with this approach is that it relies on the reliable extraction of robust features from sonar data. But sonar data are usually very noisy and it might be difficult to obtain enough reliable features.

To register line segments extracted from images to model, Ayache and Faugeras also used probabilistic predictions of feature locations and their uncertainties [7]. To further improve the reliability of feature association, they employed a strategy which starts with registering features which have a small error covariance, or features which can be matched unambiguously. Uncertainty in the positions of the remaining features may be reduced when the robot's pose is updated using the good matching pairs. Then these remaining features can be registered more reliably.

Kosaka and Kak applied a more sophisticated procedure to probabilistically associate features to models based on a maximum likelihood criterion for the mapping function [69]. For each model element, the candidates of corresponding feature are the ones within the uncertainty region. The procedure sequentially selects trial model-feature correspondences and updates the robot pose using a Kalman filter. The optimal mapping is considered as a path in the correspondence search space which contains the minimum number of missing features and a maximum value for the objective function. Here the objective function is defined from the distances in the selected correspondence pairs, taking into account the sequential updates to the pose from previously selected correspondences along the current path. Backtracking is needed to select the optimal mapping function.

To avoid making irreversible decisions about data association at too early a stage, Cox and Leonard used a multiple hypothesis approach [32] which maintains more than one interpretation of sensor measurements at one time and carries these hypotheses over time. Thus ambiguity in data association in one step can be possibly resolved in a later step when more evidence is gathered.

Once the correspondence of features are determined, the next issue is to solve for or update the pose estimate using the correspondences. The Kalman filter and the extended Kalman filter are the most commonly used estimation algorithms. The linear Kalman filter gives an optimal estimate in the sense of least-squares or minimum variance. The extended Kalman filter linearizes a non-linear problem and then applies the linear Kalman filter [49, 90]. Many mobile robot navigation systems apply the (extended) Kalman filter for robot pose estimation and model construction (e.g. [7, 70, 34, 78]).

To apply the Kalman filter, random variables are modeled with mean values and covariance

matrices. The sensor observation is modeled as

$$z = Hx + w$$

where x is the parameter to be estimated, usually the robot's pose; z is the sensor observation reading; w is the random sensor noise modeled as white Gaussian; H is a linear transformation. The recursive Kalman filter gives the new estimate of x as:

$$\begin{aligned} x &= x_0 + K(z - Hx_0) \\ K &= S_0 H^T (W + H S_0 H^T)^{-1} \\ S &= (I - KH)S_0 \\ W &= E[ww^T] \end{aligned}$$

where x and S are the mean value and covariance matrix of the new estimate of x ; x_0 and S_0 are the previous mean value and covariance matrix of x .

Straightforward correction to robot pose (non-statistical) has also been employed in navigation systems. For example, in [33], the average angle between data and model line segments are used to fix the robot's orientation, and the average displacement between data and model are used to correct the robot position. Spatial fitting methods based on least-squares fitting error are also used to solve for robot pose. These methods are often applied iteratively. We will study them in the next subsection.

2.2.4 Iterative Optimization Methods

The process of assigning correspondence and solving pose estimates can be carried out iteratively. In each cycle, a previous pose estimate is used to associate correspondences of data features to the model (assigning features to the closest model or using a statistical test). The correspondence pairs can be used to derive a potentially more accurate pose estimate (typically using a least-squares spatial fitting method). The process repeats until the pose solution converges.

Cox *et al* proposed an iterative matching algorithm which uses range measurements directly as features (thus avoiding feature extraction) [31, 28, 29]. In their approach, the model consists of line segments of a world contour, while the data are the points on the world boundary received from a laser range-finder. The system maintains an estimate of the robot's pose, so the data points can be placed near the linear world model. Matching is accomplished by associating each point with its closest line segment in the model. A point may be regarded as outlier if it is too far away from

the model. Once the correspondences are determined, a least-squares fitting is constructed as the following: Each data point is transformed with an undetermined congruence consisting of a rotation angle θ and translation vector t . For each data point v_i and the corresponding line model m_i , a fitting error as a function of θ and t is defined by summing up the distances from the transformed point $R(\theta) * v_i + t$ to the line m_i . Then the congruence is solved by minimizing the fitting error. The entire process of associating correspondence and fitting is iterated until convergence. Finally, the derived congruence is applied to correct the robot pose.

Cox's method is efficient and effective for indoor linear environments (i.e. whose contours consist of mostly long line segments). The algorithm has the nice property that it does not require feature extraction, while it can also handle outliers to some extent. A limitation of the method is that it requires an analytical world model. Thus the method can not be directly applied for navigation in unknown environments. We are able to extend this algorithm, however, for matching a set of data points against another set of data by first fitting line segments to one set of data (see Chapter 5).

In the CMU Navlab project, Hebert *et al* described a gradient descent optimization algorithm for matching terrain elevation maps [53]. The purpose of matching is to align the local maps so as to integrate them into a global map. The displacement (and correspondence) between two maps is determined by an unknown transformation v . A distance function E is defined as the sum of the squared distances between the corresponding points in the two maps. The function is then minimized with respect to v using an iterative gradient descent of the form: $v^{i+1} = v^i + k \frac{\partial E}{\partial v}(v^i)$. The derivatives needed in the algorithm involve the gradient of the terrain surfaces which are computed from the maps. The algorithm requires an initial estimate of v to start the iteration.

2.2.5 Our Work

We take an approach to the robot self-localization problem that aligns laser range scans, similar to the approach of Cox *et al*. However, we consider more general cases as (1) an arbitrary two-dimensional world, not necessarily polygonal; and (2) an unknown environment. We intend to address the localization problem for both path execution and exploration of unknown world. Since an analytical model of the environment is not available to us, we have to match a range scan to another reference scan in order to derive relative robot pose. This is much harder than matching data points to a linear world model.

In Part II of the thesis, we present two algorithms for aligning a pair of range scans. These algorithms use iterative optimization techniques and they do not require explicit feature interpretation.

The first algorithm is based on a combined search/least-squares procedure to minimize a distance function and solve for the relative transformation between the scan poses, where the distance function is defined from the approximated tangent lines at pairs of corresponding points on the two scans. The second algorithm is based on iteratively associating point-to-point correspondences and solving incremental updates to the transformation.

2.3 World Models for Robot Navigation

We first discuss several types of *a priori* world models which are commonly used in navigation systems. Then we discuss the dynamic construction of models from sensing data. Finally, the issue of consistent data registration in dynamic model building is studied.

2.3.1 Types of World Model

A world model, also called a map, is an internal representation of the geometric, topological, or physical properties of the world. Major uses of a model include path planning and pose estimation. We are primarily interested in world models for pose estimation.

For the purpose of pose estimation, the world model should keep two types of information: the metric locations of the world entities and the descriptions of their features. The locations, usually represented by coordinates in a global reference frame, serve as the reference for computing vehicle pose. The descriptions are about the geometric or other types of distinct features that can be detected by sensors in order to recognize the entities. For primitive features, such as lines and points in images, the descriptions are often implicitly embedded in the recognition algorithm.

If there are fixed, easily detectable beacons in the environment, then the world representation for robot self-localization may be simply a map recording the locations of these beacon. Many robot systems for industrial environments, such as AGV's (automated guided vehicles), use this approach. However, installing the beacons means altering the environment, which is only practical in very restricted domains. For more flexible autonomous robot applications, natural features should be used.

A 2D contour model represents the boundaries of the objects (or free space) in the world as projected to the floor plane. The contour curve can be represented as a set of line segments (such as in [29, 33, 34]), by a parametric curve (e.g. using the curvature primal sketch [45]), or by features of the contour such as corners or occlusion boundaries [77, 78]. We also treat a set of points on the contour as a model for matching with another set of points (see Chapter 3). Typically, the

contour model is used for matching with range data or with features that have been extracted from range data. It appears that the contour model is adequate for navigation in a structured environment (e.g. a simple indoor environment) and the construction of such a model is relatively straightforward. However, since the model only represents geometric information and only in 2D space, it may be unable to model more complicated environments.

Occupancy grid is a simple representation scheme suitable for modeling an environment from range measurements [102]. This type of model represents a tessellation of the environment in 2D or 3D grids, while each elemental cell in the grid identifies the state of a specific position as being “free space” or “obstacle”. Sometime, the confidence about whether a cell is occupied is also maintained. Because of its simplicity, a grid model is chosen in some navigation systems (e.g. [2, 94]).

Another type of model working with image data is the 3D CAD model [44]. It represents the 3D line segments in the scene which can be detected in intensity images. Line segments in the image are extracted as features to be matched with the model. The model is quite flexible in modeling various kinds of environments, and the represented 3D structures are more informative for navigation than the contour model. However, if the world contains many curves, it would be inconvenient to model the world with line segments. Building such a detailed 3D model is usually a tedious task, in many situations even unrealistic. If the CAD model is used, the related feature matching problem can be very computationally expensive.

For robot exploration and mapping in large-scale spatial environments, qualitative and topological models have also been suggested [71, 79, 37].

2.3.2 Dynamic World Modeling

The goal of dynamic world modeling is to acquire and represent geometric information about the environment automatically by the robot to facilitate navigation and other tasks. Here the word “dynamic” emphasizes that the world representation (model) is built or updated dynamically with successive readings of sensor data as the robot explores the world.

Dynamic world modeling is involved in mobile robot navigation in an unknown (or partially unknown) environment. The robot cumulatively builds a world model from the sensor data as it moves around. The robot maintains the following information at each cycle of updating the model: (1) a current global world model representing the areas discovered so far; (2) some knowledge about its own pose; and (3) a robot-centered geometric model of the local environment perceived at the current pose. The dynamic modeling process is to match the local model with the current global

model and update the global model.

In an unknown environment, the robot has no absolute reference. The same sensor data must be used for both correcting the robot's pose and updating the world model. This has been remarked as "pulling yourself up by your bootstraps" [34]. However, it is only possible to do this provided there is a substantial overlap between the observed data and the current model. This overlap provides the basis for pose correction, while the new information is added to the model.

Ayache and Faugeras proposed ideas of representing and maintaining spatial relationships of geometric primitives which are recognized by the robot [7]. In their approach, successive frames of local descriptions are fused to form a more global, coherent, and accurate representation. First, local descriptions of the environment (points, lines, etc.) are extracted from images. A set of geometric relations between the primitives of two visual maps are defined, such as "identical", "contained", "parallel", and "orthogonal". The geometric relations are expressed as the constraints on the parameters of the primitives. These relations are searched for among all the primitives in two maps. Each pair of primitives is registered by computing a probabilistic distance measure and verifying an acceptance test. After registration, two primitives are combined to produce a better estimate of their parameters and reduce the uncertainty. The tool used for matching and updating is the extended Kalman filter. The geometric relations provide nice constraints for interpreting image features. However, it appears that accidental alignments may lead to many false alarms if some weak relationships (such as parallel) are searched for among all pairs of primitives. There is another problem unsolved in this approach: When the relations among the primitives are detected, how can these relations be maintained systematically and consistently as more and more image data are acquired?

Kriegman *et al* [70] discuss the problem of exploring the free space for a mobile robot operating in an unknown indoor environment using stereo vision. A model of world contour is constructed and maintained by connecting all the observed feature points on the contour without violating the visibility constraints. At each step of sensing, feature points are extracted from stereo images and the points are matched to the existing ones in the model. Those points with correspondence in the model can help reducing the uncertainty in the model. Otherwise new points are inserted in the model to augment the discovered free space. An $O(n^3)$ algorithm is used to update the model while abiding the visibility constraints. A more efficient algorithm ($O(n \log n)$ time) for connecting the contour points without violating the visibility constraints is available [1]. The algorithm requires that all the points belong to a single simple object.

In [74], the problem of matching a local model with a global model and constructing a new global model is studied using simulated data. From each scan of range data, line segments are extracted and classified according to the status of how it is being occluded (for example, a line segment labeled as “left partial wall” means that the right end of the line is an occlusion boundary). The lines are matched to a global model with the constraints according to the line types and their poses. The matching process starts with a initial matching of complete walls and then propagates to the entire model. After a successful matching, several rules are applied to update the global model. The constructed model is globally consistent, which is desirable for global navigation. However, the weakness of this work is that sensing and line extraction are assumed to be perfect, which is quite impractical in real applications. Moreover, the approach did not make use of the robot’s pose in registering local models onto the global model. Instead, a (constrained) global search is used to match models elements, which is quite inefficient.

Crowley discussed the approach of building and maintaining a composite local model [33, 34] which is an intermediate level of representation of the environment in between a global model and a sensor model. The composite local model reflects the immediate state of the local environment and it is used for local navigation. The model is built up by integrating recent information from sensors, taken from different poses, and also information from a pre-learned global model. The primitives of the composite local model are the line segments extracted from ultrasonic range data, representing the horizontal boundary of the world free space. Confidence measures or probabilistic uncertainties of model primitives are maintained. Matching a pair of line segments is accomplished by a sequence of tests on the difference in orientations, co-linearity, and overlap, with respect to the standard deviations of the uncertainties. With a successful matching, model parameters, as well as the estimated pose of the robot, are updated using the Kalman filter. In this system, the line segments in the model are maintained and updated independently from each other. The model can be inconsistent after several steps of updating. The approach must assume a linear world and require segmenting the range data into lines. Some similar approaches to constructing a world model consisting of line segments using sonar or laser sensing are reported in [38, 50].

The occupancy grid model is also used for representing an environment and integrating sensor data [42, 94, 2, 35]. A grid model is useful when a geometric interpretation of data is difficult due to large amounts of sensory noise, such as in the case of sonar.

In order to model a 3D outdoor scene, elevation maps are typically used. Asada discussed a mapping system which extracts the heights of obstacles in the scene from range images and

correlates local maps to update a global map [5].

A method of continuously updating a map using range measurements by a neural network is discussed in [95]. Pose error in different sensor locations is not considered in this work. A study on models for cognitive maps using neural networks with potential application for mobile robots is discussed in [99].

Dynamic construction of a topological model requires precise identification of landmarks or markers [37]. So far we have only seen theoretical work in this direction.

2.3.3 Consistency in Dynamic World Models

The general approach to dynamic world modeling is based on aligning successive frames of sensor data or aligning each local frame with a cumulative global model, and then integrating the new frame of data into the global model by averaging the data or using a Kalman filter. A major problem with this approach is that the resulting world model eventually becomes inconsistent as different parts of the model are updated independently. Moreover, even if some inconsistency in the model is detected in a later stage, it may not be resolved because the data in the model may have been permanently integrated.

Clearly, the problem is that new frames of data are irreversibly integrated into the global model at too early a stage. In order to avoid this problem, the local coordinate systems of the data frames must be maintained and there must be some mechanism to resolve the inconsistency among the data frames once such inconsistency is recognized.

Consider the following scenario. The model contains two frames of data which are associated with poses (or local coordinate systems) P_1 and P_2 , respectively. Suppose that a new frame of sensor data is collected at pose P_3 and geometric relationships of both P_3P_1 and P_3P_2 are derived from matching the frames of data (the relationships are illustrated in Fig. 2.2). Now if we register frame P_3 to either frame P_1 or P_2 , it may be inconsistent with the other frame because the three relationships, P_1P_2 , P_3P_1 , and P_3P_2 , could be inconsistent as they are derived from different sources. Therefore, in order to register the new frame to the model, we should first resolve the inconsistency among the three poses. Based on the three relationships and their uncertainties, we can update P_1 , P_2 , and P_3 to make them consistent (i.e. to ensure that the composition of P_1P_2 , P_2P_3 and P_3P_1 yields an identity transformation). Generally, if there are other frames of data in the model which are related to P_1 and P_2 , they should also be updated at the same time to ensure consistency within the model.

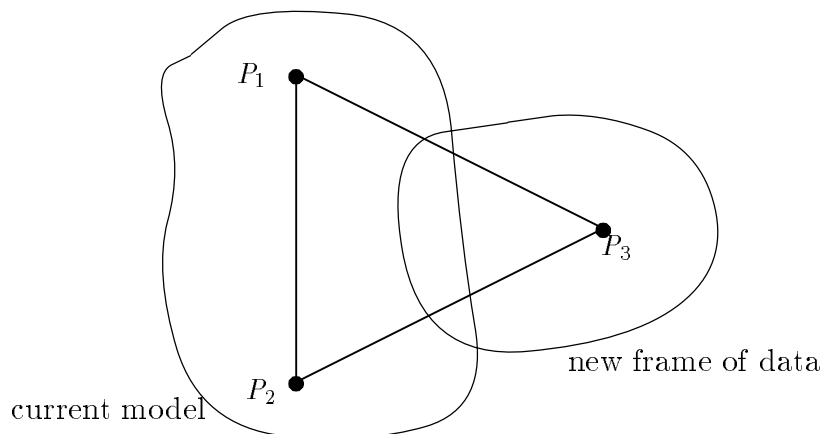


Figure 2.2: Illustration of relationships between model and new data.

Chatila and Laumond studied the consistency issue in dynamic world modeling in their HILARE robot project [25]. In their system, range signals are segmented into objects and each discovered object is associated with a local object frame. The local frame itself is referenced in an absolute global frame along with the uncertainty on the robot’s pose at which the object frame is constructed. New sensor data are matched to the current model of individual object frames. If some object which has been discovered earlier is seen again, its object frame pose is updated (by averaging). In circumstances that the uncertainty of some object frame in the model is less than the uncertainty of the current robot pose (note: this happens when the object frame is created earlier, and later the robot sees the object again), the robot’s pose may be corrected with respect to that object frame. After correcting the current robot pose, the correction is propagated backwards with a “fading” effect to correct the previous poses. The object frames created at these poses can also be updated.

Although the HILARE system considered the issue of resolving model inconsistency, the solution does not appear to be satisfactory in the following aspects. First of all, the system associates local frames with “objects”. But if the results of segmenting sensor data or matching the data with model are imperfect, the “objects” and therefore the local frames may not be defined or maintained consistently. When a previously recorded object is detected again, the system only attempts to update the poses (and the associated frames) along the path between the two instances of detecting this object, while the global consistency among all frames in the model may not be maintained. HILARE only uses a scalar random variable to represent the uncertainty of a three-degree-of-freedom pose, which seems inadequate. Using a heuristic “fading” function to determine the updates to the poses also appears to be inadequate. A statistical filter may be applied here.

Smith and Cheeseman [107] formulated a representation of spatial relationship and uncertainty using mean vectors and covariance matrices, and provided compounding and merging operations to estimate the combined spatial relationship and uncertainty from a network of uncertain relations. By taking all the available spatial relations into account at the same time, consistency among the estimates are ensured. However, their method of combining spatial relations has a limitation that it can only deal with some networks which are reducible to a combination of serial and parallel connections, but a more complicated network of relations can not be combined by their method.

Durrant-Whyte [40] discussed the representation of spatial uncertainty and the geometric and topological constraints among uncertain measurements. In another related work [39], they studied consistent integration and propagation of sensor information. Here the consistency among sensor data is enforced by explicitly applying constraints on the loops of the network to update the coordinate transformations along the network edges. The updating procedure is formulated as a constrained optimization problem. In another approach to a similar problem, Tang and Lee [115] formulated a geometric feature relation graph for consistent sensor data fusion. They proposed a two-step procedure for resolving the inconsistency in a network of measurements of relations. In the first step, the compromise between conflicting measurements of relations is achieved by the fusion of these measurements. Then in the second step, measurements of relations in the network are modified to be consistent with this compromise optimally based on the assumed normal distributions of the uncertain measurements of relations. The updating procedure is based on an optimization criterion subject to the consistency constraints on the loops in the network. It appears that the separated fusing and updating steps may yield a suboptimal overall solution. Also the constrained optimization formulation leads to a large and complicated non-linear system which is computationally expensive to solve.

Kamgar-Parsi *et al* also addressed the issue of global consistency in registering multiple overlapping bathymetric range images [63]. They first use contours of constant range as primitives to locally match pairs of intersecting images. Then the global registration among all images is accomplished by solving an optimization problem based on all the local constraints, as well as energies for penalizing image stretch or bent. The optimization problem is not formulated in the context of statistical estimation. The configuration of image intersections considered here is in the form of regular lattices made up of only horizontal and vertical image strips.

2.3.4 Our Work

We also study the consistent sensor data integration problem (in Part III of the thesis). Similar to some of the previous approaches, we formulate a network of measurements of relations to derive consistent estimates of poses and relations. A highlight of our approach is that we formulate a maximum likelihood criterion for the optimal estimation from all the relations in the network. Thus a global optimal solution can be obtained while the whole network is ensured to be consistent. We treat poses (which are the nodes of the network), rather than the relations, as free variables. Thus we can obtain a concise closed-form solution of the optimal estimation without doing constrained optimization. We particularly focus on consistently combining laser range scans and odometry measurements for robot pose estimation and map building. We present a uniform modeling of the uncertain relations from these two types of measurements. The problem we address is similar to the one studied by Smith and Cheeseman [107]. But our approach has the advantage that an arbitrary network of measurements of relations can be consistently combined, while their method can only handle a certain type of simple network.

2.4 Shape Registration

In the previous section, we discussed sensor data matching techniques specifically for the purpose of robot pose estimation and map building. Now we want to discuss general matching techniques for a broader domain of applications. Besides robot navigation, other applications which involve shape matching include inspection or modeling of a 3D object using range images; localization or recognition of objects from 2D intensity images etc..

We regard that sensor data or features in sensor data describe the shape of an object or environment being sensed. For example, a 2D range scan made by a robot describes the contour shape of the local environment; a range image represents a 3D surface; and edges in an image may depict the shape of an object. We thus consider the matching problem as one of registering the data shape with a model shape.

The problem of shape registration (or shape matching) can be defined as the following. Given sensor data (range measurements or features in an intensity image) which describe a shape, and given a model of the same shape, determine a transformation (rotation, translation, and possibly scaling) from the sensor coordinate system to the model coordinate system such that the transformed data shape is aligned with the model shape. The alignment criteria is usually expressed in

the sense of a least-squares metric whose residual may also give a measure of the equivalence (or similarity) of the two shapes.

2.4.1 Model-based Recognition

Shape registration is closely related to model-based shape recognition. Essentially, shape recognition methods can be viewed as techniques for searching for a match between the shape of objects in the image and the shape of the model, although the recognition problem also involves other difficult issues such as the identification of an object from a possible large set of candidates. A substantial collection of work on model-based shape recognition is reviewed by Grimson [51]. The matching problem consists of two components. One is to establish the correct correspondence between the features in the scene and that in the model. The other is to estimate the parameters of the relative transformation between the two shapes based on a known correspondence. These two components are tightly coupled and each one can be solved easily assuming the other.

The search for a match can be performed either in correspondence space or in pose space. The correspondence space search examines possible pairings of features in the image and the model. Then it evaluates the degree of matching for each examined correspondence in order to find an optimal match. The interpretation tree method [51] is a typical example in this category. The disadvantage of this type of approach is that it requires exploring a potentially exponential subset of all possible correspondences.

The search for a match can also be performed in the pose space where many different transformations between the model coordinate system and the image coordinate system are examined in order to find an optimal transformation which leads to the best correspondence. Examples of techniques searching the pose space are the generalized Hough transform methods [8, 111] and transformation space sampling [23], in which all possible pairings of one model feature and one scene feature vote for the best candidate among a set of discretized transformations. The Hausdorff distance transformation method [57] is another example of searching the pose space. The disadvantage with this class of methods is that a large space of discretized transformations may have to be maintained and examined.

Some methods use a combination of correspondence space search and pose space search. Examples include the alignment method [58] which first determines the transformation based on a small set of features and then verifies it using rest of the data, and the geometric hashing method [73] which chooses transformation invariant feature representations to vote for both the object and its

pose.

The shape recognition methods we discuss here are generally aimed at finding 3D objects from 2D images. The problem of planar shape registration, which we will discuss next, can be considered as a special case which avoids the issue of 3D to 2D projection.

2.4.2 Matching Planar Shapes

We consider a simpler problem of matching 2D planar shapes in which the transformation only consists of rotation, translation, and uniform scaling. The planar shape representation and matching problem has been studied in the past. An overview of planar shape analysis techniques until 1980 is given in [98]. Various shape representation and matching methods are discussed in [117].

Besides the general shape recognition methods, other techniques have been studied for matching only planar shapes. One approach is to use rotation and translation invariant shape descriptions. For example, the multiscale method uses scale space images to match contour curves [93]. The extended Gaussian image based method is capable of matching shapes with slight deformation [92]. These transformation invariant shape descriptions often require the second derivatives of the shape and therefore are less stable in the presence of noise.

The footprint method uses signatures which are obtained from an arclength-turning angle graph to match two boundary curves under rigid motion, where a segment of image curve is matched to the model by correlating the points along the curve [61, 105]. Because arc-length sampling of curves is used to establish correspondences, this method fails if the model and data curves are incomplete (e.g. due to occlusion).

Some methods are capable of matching 2D shapes with partial occlusions. The HYPER system uses the long edges in the polygonal approximation of the shape to generate hypotheses about the pose of the shape, and then verify the transformation with other model-data correspondence pairs [6]. A template matching algorithm for recognizing partially occluded parts, which uses boundary saliency measures as weights to perform a weighted Hough transform, is described in [116].

Iterative optimization methods can be used for matching shapes. For recognizing hand-written digits, Hinton *et al* minimizes an energy function which includes both the deformation energy of a spline model and the distance between the model and the data curve [55].

2.4.3 Registration of 3D Shapes

In this subsection, we discuss techniques for the registration of 3D shapes. Commonly, 3D shapes are described by range images. Purposes of 3D shape registration include matching a range image of an object to a CAD surface model for inspection, matching partial views of an object for creating a surface model etc.. There are wide industrial and medical applications for these tasks.

A typical approach from object recognition techniques is to identify and associate features. From range images, the commonly extracted features are curvature extrema. Feature-based methods have the potential of finding an arbitrary registration without *a priori* knowledge. They also tolerate a certain amount of occlusions in the data. But the effectiveness of these methods depends on the reliability of the feature's extraction and association procedures.

Descriptions based on global properties of the shape, such as moments, may be used for registration. This type of approach is likely to fail if the shape to be registered is incomplete or it contains outliers. The registration method based on Spherical Attribute Image [54] can be regarded as a global method. But a heuristically improved procedure may allow matching partial views to some extent.

There is another type of approach to aligning shapes without explicitly interpreting features. Typically, these methods start with an initial estimate of the geometric transformation between the two shapes and they iteratively improve the estimate of the transformation. At each iteration, low level primitives (such as raw data points or oriented point) are associated to targets on the model. Then a least-squares solution of the transformation is derived by minimizing a distance measure between the two shapes. A better estimate of the transformation leads to a better correspondence between data and model, which in turn helps to improve the solution of the transformation. Upon convergence of the iterative process, the overall transformation which registers the data shape to the model is obtained. While the iterative methods avoid the difficult feature interpretation step, they have a disadvantage that an initial guess of the registration is required and only small errors can be corrected during the iterative process. In some applications, an initial guess of transformation is available from knowledge of camera locations (such as from odometry). Iterative methods are especially suitable for aligning free-form shapes described by range measurements because raw measurements can be directly used for matching and the feature extraction step is not needed.

Besl and McKay described a general-purpose algorithm for shape matching called the iterative closest point (ICP) algorithm [12]. The algorithm is iterative, and each iteration contains the establishment of point correspondences and the estimation of a rigid motion. First, each data point

in the data shape is related to the closest model point in the model shape. A rigid motion is estimated using these correspondences through a least-squares criterion. Then the data shape is transformed according to the computed motion parameters. Finally, the correspondence is updated between the transformed data and the model shape. Modified versions of the ICP algorithm have been proposed for registration of range images in the presence of outliers [89, 68]. One disadvantage of the ICP algorithm is that it converges slowly. It also appears to be ineffective in solving the rotation component of the transformation when the shape is curved.

If the shape is smooth and data are dense, it will be advantageous to use the tangent information of the shape in the matching process. The tangent information may improve the reliability of correspondence and the convergence speed. Chen and Medioni [26] described such an iterative algorithm to register range images of partial views of a 3D object. In each iteration of the algorithm, a distance function is defined as the sum of the squared distances from the control points on one shape to the tangent planes defined at the corresponding points on the other shape. The function is minimized to derive an update to the transformation variables. Both the 3D rotation and translation are treated as variables in the distance function and solved using least-squares. With dense and smooth range images where tangent planes can be robustly defined, this algorithm is very effective and efficient in registering the two shapes. The algorithm requires a good initial guess of the registration. Otherwise the iterative method may converge to a local (non-global) minimum or fail to converge at all. Methods similar to this one have also been reported (such as [36, 48]).

To ensure successful convergence to global minimum in registration of range images, Blais and Levine used a stochastic optimization method, simulated annealing, to minimize a cost function [14]. One disadvantage of stochastic optimization methods is the high computational cost.

A model-based method for 3D localization using range image is presented in [24]. In this approach, a model surface, which itself is constructed from a range image, is represented using octree-spline. Then another range image is registered to the model by an iterative optimization procedure which minimizes a least-squares distance from the image points to the model surface.

Once range images representing different views of a surface are registered, the next issue is merging multiple range images and creating a model of the surface. Some work on this issue are reported in [109, 103].

2.4.4 Our Work

Shape registration is the theme of this thesis. We study primarily the registration of 2D range scans which represent the contour shape of the environment, for the applications of robot pose estimation and map building. Our two algorithms (presented in Part II) employ iterative optimization techniques. In Part IV of the thesis, we examine our methods again for the registration of 3D range surfaces and planar image contour curves.

Our first algorithm uses tangent lines or tangent planes defined at corresponding points on two shapes to form a distance function and then minimizes the distance in order to align the two shapes. It differs from typical iterative least-squares methods (such as the one used by Chen and Medioni [26]) in that we do not include the rotation variable in the least-squares solution. Instead, we search the distance function to find the optimal rotation while solving translation (and possibly scaling) in an embedded least-squares procedure. Thus our algorithm is more robust in global convergence while it also maintains the efficiency of iterative least-squares methods. Our algorithm also includes other enhancement over Chen and Medioni's method in that we define a distance function which takes into account a cost for outliers and we also check for consistency in normal directions in correspondence association.

Our second algorithm is based on iteratively associating point to point correspondences on the two shapes and solving the transformation by a least-squares procedure, similar to the ICP algorithm by Besl and McKay [12]. However, in addition to choosing the closest point, we utilize a second correspondence rule called the matching-range point rule to determine another correspondence for a data point. By using two sets of correspondence points, our IDC (iterative dual correspondence) algorithm can effectively solve for both rotation and translation. It also converges significantly faster than the ICP algorithm.

Part II

Matching Range Scans for Robot Pose Estimation

For a mobile robot exploring an unknown environment, the only absolute reference is the one defined by its initial position. When the robot moves to a new place, it can estimate its pose either by using odometry or by comparing external measurements it obtained from its current pose against those obtained from the initial pose. Since odometry estimate may lead to unbounded pose error, it is important to compare sensor data for robot localization.

In this part of the thesis, we study the problem of matching 2D range scans for robot pose estimation. Here a range scan is a set of points as detected by a range sensor which describes the contour shape of the environment. We present two algorithms for aligning a scan with another scan in order to derive their relative pose. This set of work has been previously reported in [84, 85].

In Chapter 3, we introduce a matching algorithm which uses a combined strategy of solving rotation with a linear search and solving translation with a least-squares method. We call it the rotation search/least-squares algorithm. It uses computed tangent directions in the process of aligning the two scans.

In Chapter 4, we present another algorithm to solve the same scan matching problem. This algorithm is based on iteratively associating correspondences between the two scans and solving the transformation from the correspondence pairs. We call this the point correspondence based algorithm.

In Chapter 5, we present scan matching experiments using these two algorithms. Both simulated and real examples are given.

Chapter 3

A Search/Least-Squares Algorithm for Matching Scans

3.1 Problem Definition

In this chapter, we present an algorithm for matching range scans. The problem of robot self-localization using range measurements has been studied in the past, as we reviewed in Chapter 2. A notable example is the iterative algorithm proposed by Cox [29] which matches range measurements to line segments in the world model (see Section 2.2.4 for a discussion of their method). However, our problem is different from theirs in that we do not have an *a priori* analytical world model. Rather than matching data to a model, we can only match one set of data to another. Moreover, we consider a general case that the world contour may contain curves instead of being polygonal.

3.1.1 Pose Estimation by Aligning Scans

A range scan is a list of points corresponding to the intersection points of a laser beam with objects in the robot's environment. The laser beam rotates in a horizontal plane and emanates from a sensor mounted on the robot. Thus a range scan describes a 2D slice of the environment. Points of the range scan can be specified in a polar coordinate system whose origin is the location of the sensor, and the reference axis for the laser beam direction is the home orientation of the rotating range sensor. Each scan point is represented by the laser beam direction, and the range measurement along that direction. We refer to $(O(x, y), \theta_0)$ as the scan pose, where O is the position of the sensor in a global coordinate system and θ_0 is the sensor home orientation.

Suppose that the robot starts at pose P_{ref} (which is our reference pose) and takes a scan (call it the reference scan S_{ref}). The robot then moves through a static environment to a new pose P_{new}

and takes another scan (call it the new scan S_{new}). The approximate difference of pose P_{new} from pose P_{ref} (i.e. the relative translation and rotation) is usually known from odometry information. However, this information is often imperfect due to wheel slippage. Our task is to determine the exact difference of pose P_{new} with respect to pose P_{ref} , by aligning the two scans.

The matching problem is formulated as the following: Initially assuming that the pose of S_{new} is P'_{new} , find a rotation ω and a translation T for S_{new} such that, after the transformation, S_{new} is aligned with S_{ref} .

3.1.2 Criterion for Matching Scans

A scan is a sequence of points which represent the contour curve of the local environment. To align two scans, we implicitly align the two contour curves. However, due to the existence of random sensing noise and self-occlusion, we may not be able to perfectly align the two scans even if we know the exact relative pose between them. There exist two types of discrepancies between the scans.

In the first type, there are small deviations of scan points from the true contour due to random sensing noise. These deviations can be characterized by an error distribution function which depends on the sensor. Another type of discrepancy is the gross difference between the scans caused by occlusion (i.e. some area is only visible in one scan but not in the other). We regard this kind of discrepancy as an outlier which cannot be described by the sensing error distribution.

Naturally, one way of aligning two scans is to find a relative transformation which gives the best alignment of the overlapping part in the sense of minimum least-squares error, while ignoring the outlier parts. We can formulate this matching criterion as the minimization of a distance measure between the two scans, as a function of the rotation and translation. To exclude outliers in defining the distance measure, we can apply the concept of robust statistics [52, 13].

3.1.3 Iterative Scan Matching Algorithms

We will study scan matching algorithms which use iterative optimization. This type of algorithm is typically based on defining some distance function between the two scans and iteratively minimizing the distance function.

Two types of constraints are usually used in defining the distance function. One is based on the distances of point correspondence pairs. The other is based on the normal distances from data points to lines or tangent lines on the model. If the model is smooth and continuous, the point to

tangent line constraint should lead to better convergence than the point to point constraint. But if the model consists of noisy points, a fitting step is required to derive lines.

Once the distance function is defined, we need to determine a set of transformation variables which minimizes the distance function. Because the distance function is generally not a smooth one, gradient based search methods are not applicable. One possible way of searching for a minimum is to sample the search space. Stochastic methods (such as simulated annealing) may also be applied to minimize the non-smooth distance function. These methods are usually computationally expensive. Another commonly used method is fixed-point iterations. It does not explicitly minimize a distance function. Instead, it solves for a least-squares update and applies it to reduce pose error, at each step of the iteration. This technique is very efficient and also effective. But it requires a good initial guess to ensure convergence.

In our search/least-squares method presented below, we heuristically combine search and least-squares updates. The idea is to compromise the global convergence of a search algorithm and the efficiency of an iterative least-squares algorithm. The distance function is based on point to tangent line distances. In next chapter, we present another algorithm which is based on minimizing point to point constraints. A fixed-point iteration method will be used. Later in the experiments presented in Chapter 5, we also compare our algorithms with another method which is based on point to line constraints.

3.2 Search/Least-Squares Matching Algorithm

Our approach to the scan matching problem is to define a distance measure between the two scans and search for an appropriate rigid transformation which minimizes the distance. Although the search space is essentially a three-dimensional one (rotation and two-dimensional translation), we try to reduce the search problem to an efficient one-dimensional search plus an embedded least-squares solution, by carefully formulating the distance measure.

The idea of the matching method is briefly described below. First, we compute the tangent directions on the scans by fitting lines to sets of consecutive scan points. Then we associate approximate correspondences of scan points, assuming a known rotation angle ω (but not the translation T). From each pair of corresponding points, we formulate a linear equation about the unknown translation T . Then, by using all correspondence pairs, we define a least-squares model for T which also represents a matching distance as a function of ω . Outliers can be detected using gates and they contribute a constant cost to the distance measure. The final step is to search for a

rotation ω which minimizes this distance function. The translation T is solved through the least-squares method. We refer to this method as rotation search/least-squares method to distinguish it from pure least-squares based fixed-point iteration type of methods.

The steps of the algorithm are summarized as the following:

1. Project the reference scan S_{ref} to the pose P'_{new} so that the two scans are represented in the same coordinate system. Discard those points on S_{ref} which are likely not to be visible from the new pose.
2. Compute the tangent directions on each scan by fitting lines to a neighborhood of sample points. Discard unreliable tangent lines (such as at corners or depth discontinuities).
3. Decide on a trial value of the rotation ω from a global search procedure.
4. For each point on S_{new} , use the rotation angle ω to define an approximate corresponding point on S_{ref} and compute the point through interpolation. Check the correspondence with some thresholds in order to reject outliers.
5. Use all the correspondence pairs to construct a least-squares model for T and find the least-squares solution. Define a matching distance as a function of ω from the least-squares residual and the outlier penalties.
6. Update the rigid transformation by the least-squares solution of T .
7. Repeat steps 3–6 and find the rotation ω which minimizes the matching distance function. Also obtain the overall translation by integrating the individual updates.

The algorithm will be explained in detail in the following sections.

3.3 Projection of Reference Scan

The reference scan S_{ref} is originally defined at pose P_{ref} . We would like to project it to the (approximate) new pose P'_{new} to emulate the world as viewed from P'_{new} . This is easily done by a change of coordinate systems for the points in S_{ref} .

After projection, we determine whether each point in S_{ref} is visible from P'_{new} based on the bounded obstacle assumption and opaque assumption. (1) Assume that initially the points in S_{ref} are ordered by their polar angles (say counterclockwise). After projecting the points to the new

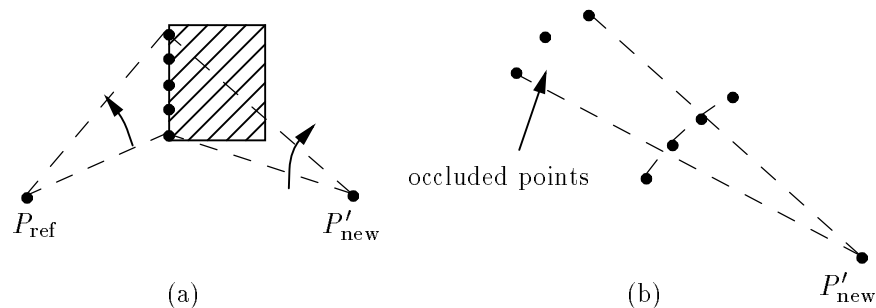


Figure 3.1: Scan points are considered invisible after projection if (a) their angles are in the wrong order or (b) there are other points blocking the rays.

pose, if the new polar angles of some points are in the wrong order (i.e. become clockwise), then the surface containing these points is facing away from the sensor and thus the points are not visible. (2) Along the rays from the new origin P'_{new} to the points, if there are other points (either from S_{ref} itself or from the new scan S_{new}) close enough to the rays, then the points further from the origin are considered hidden (see Fig. 3.1 for illustration). The points determined as nonvisible are discarded.

Finally, we obtain a set of points in S_{ref} which are ordered by angles. The projected scan can be readily used as a reference for registering S_{new} .

3.4 Fitting Tangent Lines

At each sample point on a scan, we compute an approximated tangent line by fitting to a neighborhood of sample points centered at that point.

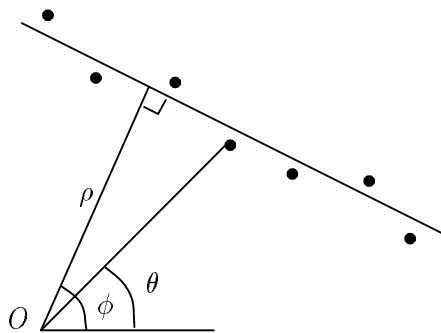


Figure 3.2: Parameters of a line fit to a set of points.

A line fit to a set of points (x_i, y_i) 's in neighborhood of size n is defined by the parameters ρ

(normal distance from the origin to the line) and ϕ (direction of a normal to the line) (see Fig. 3.2) which minimize the following error:

$$E_{\text{fit}} = \sum_{i=1}^n (x_i \cos \phi + y_i \sin \phi - \rho)^2. \quad (3.1)$$

A closed-form solution can be derived as the following:

$$\begin{aligned} \phi &= \frac{1}{2} \tan^{-1} \frac{-2S_{xy}}{S_{y^2} - S_{x^2}} \\ \rho &= \bar{x} \cos \phi + \bar{y} \sin \phi, \end{aligned}$$

and

$$\min_{(\phi, \rho)}(E_{\text{fit}}) = \frac{1}{2} \left(S_{x^2} + S_{y^2} - \sqrt{4S_{xy}^2 + (S_{y^2} - S_{x^2})^2} \right),$$

where

$$\begin{aligned} \bar{x} &= \frac{1}{n} \sum_{i=1}^n x_i \\ \bar{y} &= \frac{1}{n} \sum_{i=1}^n y_i \\ S_{x^2} &= \sum_{i=1}^n (x_i - \bar{x})^2 \\ S_{y^2} &= \sum_{i=1}^n (y_i - \bar{y})^2 \\ S_{xy} &= \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}). \end{aligned}$$

Although we fit a line at every sample point, some of the lines should not be considered as tangent lines if the world contour itself is not smooth enough near these points. Specifically, we want to exclude sample points near corners or occlusion boundaries.

Two indicators can help us to recognize these non-smooth regions. First, we check the incidence angle $\theta - \phi$ (θ is the polar angle of the sample point and ϕ is the computed normal direction, see Fig. 3.2). A high incidence angle indicates that either the sensing direction is nearly parallel to a surface so that the range measurements are unreliable, or there is a depth discontinuity (occlusion boundary).

Another value we check is the fitting error $\min(E_{\text{fit}})$ which indicates the co-linearity of the points or the smoothness of the local region. The error is usually small for straight lines or smooth

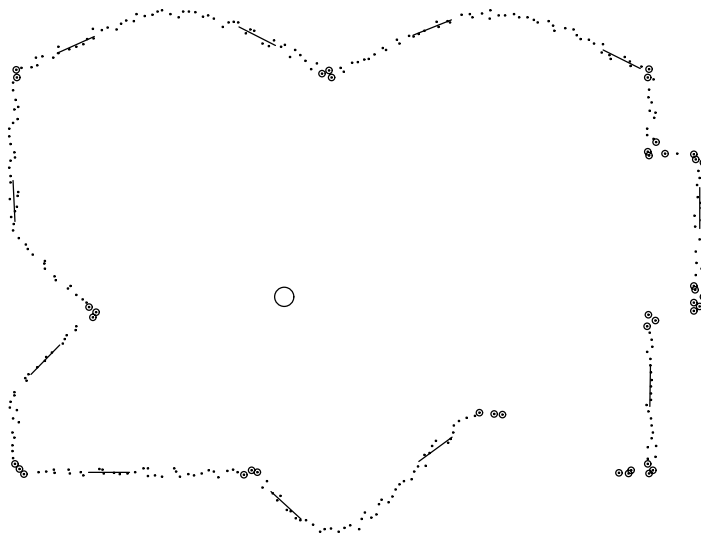


Figure 3.3: The points labeled by circles are discarded. Tangent lines are computed at all the other points. Only some of these tangent lines are shown for clarity.

curves, but large at high-curvature points or highly noisy regions. Therefore, a large fitting error usually means that the tangent line is poorly defined.

We use the above two indicators to reject unreliable fitting results. Only when both values are within predefined limits, we consider the fitted line as the tangent line at the current sample point. Otherwise, we regard the tangent line to be undefined. Fig. 3.3 shows an example of using incidence angle and fitting error to detect points where tangent lines are unreliable. We compute tangent lines at all the other scan points.

It is interesting to note that the points we discard for being unreliable in defining tangent (which are likely corners and occlusion boundaries) can be considered as features. It is possible to do feature-based matching using these points, provided that the correspondences between the features on the two scans can be determined. As the number of identifiable features in the scan is far less than the number of available data points, we believe that the feature-based solution is less accurate and less robust than our method which uses all the data points where a tangent is defined.

3.5 Correspondence Definition

We define the correspondence of scan points and set up an equation based on the values of a pair of corresponding points. For the convenience of analysis, we initially regard the scans as continuous

curves rather than sets of discrete points and we also ignore sensing noise or occlusion, until we instantiate the equation for the actual scan points.

Once we have projected S_{ref} to the pose P'_{new} , we represent the two scans in the same coordinate system (defined by (T_0, θ_0)). The two scans differ only by a rotation ω and a translation $T = (T_x, T_y)^t$.

Let P_1 be a sample point on S_{new} and let P_2 be the true corresponding point on S_{ref} (i.e. they both represent the same physical point in the world). The two points are related by

$$P_2 = R_\omega P_1 + T, \quad (3.2)$$

where $R_\omega = \begin{pmatrix} \cos \omega & -\sin \omega \\ \sin \omega & \cos \omega \end{pmatrix}$ is the rotation matrix. Consider the tangent line defined on S_{new} at P_1 and another tangent line on S_{new} at P_2 . Let the normal directions of the two tangent lines be \vec{n}_1 and \vec{n}_2 , respectively. Then we can derive from Eq. 3.2 that

$$\vec{n}_2 = R_\omega \vec{n}_1 \quad (3.3)$$

$$\vec{n}_2 \cdot P_2 = (R_\omega \vec{n}_1) \cdot (R_\omega P_1) + (R_\omega \vec{n}_1) \cdot T. \quad (3.4)$$

Our strategy is to use Eq. 3.4 to estimate the translation T , given the rotation ω and the two scans. However, we note that for a point P_1 on S_{new} , the exact correspondence point P_2 on S_{ref} also depends on T . Therefore, we want to derive an approximated version of Eq. 3.4 which does not use the exact correspondence point P_2 .

We will choose a point P^* on S_{ref} which is close to P_2 . There are many choices of selecting P^* based on $R_\omega P_1$ (which is P_1 after correcting the rotation), such as the closest point from $R_\omega P_1$ or the intersection of the normal line at $R_\omega P_1$ with S_{ref} . For the convenience of searching, we choose P^* as the intersection of the extension of vector $R_\omega P_1$ with scan S_{ref} (see Fig. 3.4 for illustration). Let \vec{n}^* be the normal direction of the tangent line at P^* on S_{ref} . If scan S_{ref} is smooth and if P^* is not too far away from P_2 , we have the approximate relationships:

$$(R_\omega \vec{n}_1) \cdot T \approx (R_\omega \vec{n}_1) \cdot (P^* - R_\omega P_1) \quad (3.5)$$

$$\vec{n}^* \cdot T \approx \vec{n}^* \cdot (P^* - R_\omega P_1). \quad (3.6)$$

The approximation errors of the above relationships are of the order $O(x^2)$ where x is related to $|T|/|P_1|$ (see Appendix A for derivation). Therefore, if $|T| \ll |P_1|$ and if the contour curve is smooth, the approximation error is small. We can combine the above two relationships to form a

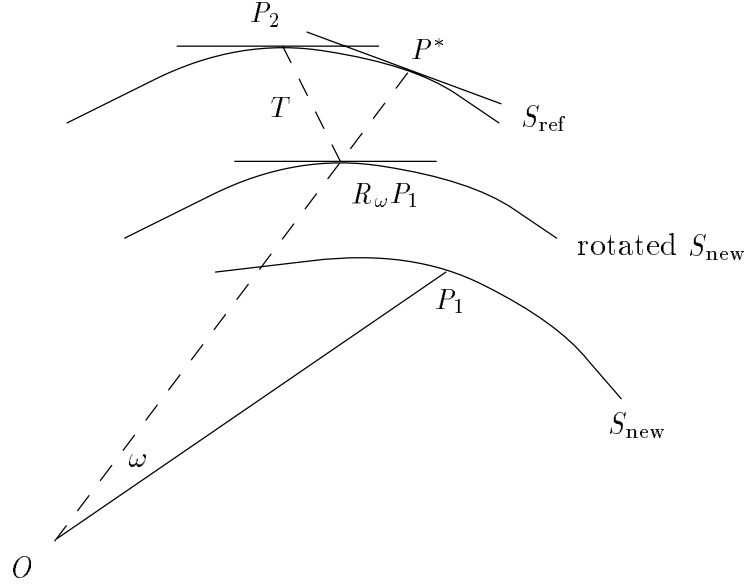


Figure 3.4: Illustration of point relationships.

more accurate approximation whose error is of the order $O(x^3)$ (see Appendix A):

$$(R_\omega \vec{n}_1 + \vec{n}^*) \cdot T \approx (R_\omega \vec{n}_1 + \vec{n}^*) \cdot (P^* - R_\omega P_1). \quad (3.7)$$

Eq. 3.7 is a linear equation about T in the form:

$$C^x T_x + C^y T_y \approx D. \quad (3.8)$$

It is established based on the point P_1 and the rotation angle ω . Notice that, given P_1 and ω , we can determine P^* without using T . The coefficients C^x, C^y, D can be computed from ω and the parameters of the tangent lines.

Now if we instantiate Eq. 3.7 for n_p points on S_{new} , we can define an error function:

$$E(\omega, T) = \sum_{i=1}^{n_p} (C_i^x T_x + C_i^y T_y - D_i)^2. \quad (3.9)$$

This error indicates the “distance” between the two scans as a function of ω and T . For a given ω , we can solve for T which minimizes the above distance, in terms of ω . Moreover, the least-squares error $\min_T E(\omega, T)$ can be considered as a matching distance function which is primarily defined by the rotation ω . Our final solution is based on searching the function for an optimal ω such that the least-squares error is minimum.

3.6 Correspondence Search

Considering the fact that S_{ref} is a discrete set of points, we can obtain the approximate correspondence point P^* for P_1 by interpolation. Let the polar angle of P_1 be θ , then the point P^* is at direction $\theta + \omega$. Note that the points in S_{ref} are ordered by their polar angles. We can locate two consecutive points on S_{ref} whose angles enclose $\theta + \omega$. Then we linearly interpolate for both the range and the normal direction of P^* across the two points. This correspondence search is very efficient. We can implement the search for all correspondence pairs in linear time with respect to the number of points in the two scans.

Correspondence points which differ greatly in their normal directions or positions are considered as outliers and they are not included in the least-squares solution. Typical causes of outliers are occlusion or poor tangent estimation. We empirically choose thresholds α and H_d . A correspondence pair is accepted only if both of the following conditions are met:

$$(R_\omega \vec{n}_1) \cdot \vec{n}^* \geq \cos \alpha; \quad |D_i| \leq H_d. \quad (3.10)$$

Otherwise, we consider the pair as outlier and discard it. We also exclude the points if they are near a corner or a depth discontinuity (as detected by the line fitting procedure).

3.7 Optimization

We define a total matching distance for a given rotation ω based on the least-squares error (Eq. 3.9) and the number of outliers. Let n_p be the number of matching pairs of points and n_o be the number of outliers, the total matching distance is defined as:

$$E_{\text{match}}(\omega) = \frac{1}{n_p + n_o} (\min_T E(\omega, T) + n_o H_d^2), \quad (3.11)$$

where H_d^2 is the constant cost of an outlier (note that H_d is the threshold that detects outliers). The effect of the thresholding and adding fixed cost for outliers to the least-squares error is approximately equivalent to using a truncated quadratic robust estimator to define the distance measure. The advantage of using a robust estimator is that a small number of arbitrarily bad outliers do not lead to an arbitrarily bad estimation [52, 13].

The matching distance in Eq. 3.11 is a function of ω . Implicitly, it is also a function of the translation T (because the least-squares solution T may be difference from the true translation). But the variable ω is dominant. We expect that the distance function should have a global minimum

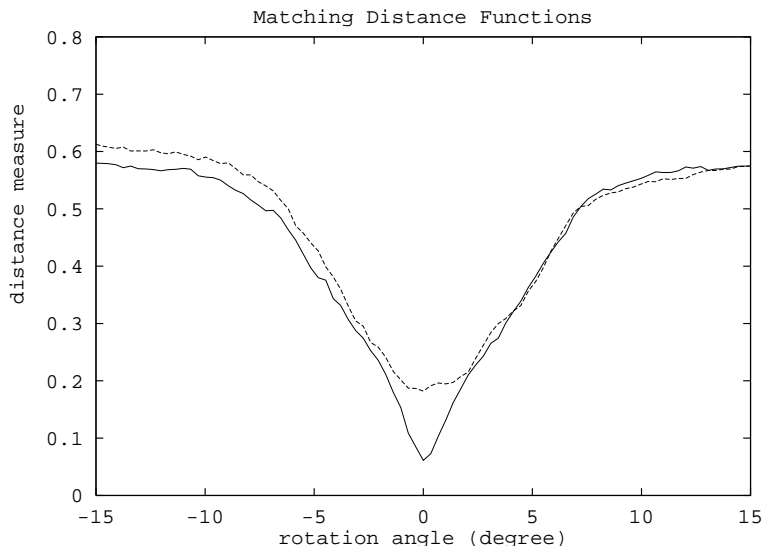


Figure 3.5: Matching distance measure as a function of rotation angle, in an example. The solid curve is the distance function when translation is zero. The dashed curve is the distance function when translation is $(0.2, 0.1)$ meters (comparing with dimension of the environment of 10 meters).

at the true rotation angle and that the function is unimodal near the minimum. From the plot of a typical $E_{\text{match}}(\omega)$ (Figure 3.5), we can see that the distance function has a single prominent valley. Furthermore, if there is little translation, it has a well-defined lowest point.

We use the search by golden section method [60] to find the minimum along the rotation dimension in the distance function. We choose this method because of its efficiency in terms of required function evaluations (considering that the distance function is not differentiable and therefore gradient-based search methods are not applicable). The search by the golden section method is stated as follows. Assume that the global minimum is enclosed in an interval $[\omega_1, \omega_2]$. The first trial point is set at $x_1 = \omega_1 + 0.618(\omega_2 - \omega_1)$. At every step, a new point is chosen at $x_2 = \omega_1 + \omega_2 - x_1$. Then depending on whether the function has a lower value at x_1 or x_2 , the better point becomes the new x_1 and the worse one replaces ω_1 or ω_2 to form a new interval which still encloses the minimum. After sufficient iterations, the final interval will be narrow enough to localize the minimum point.

Due to the residue in the translation, the optimal rotation found by the one-dimensional search may be biased. To address this problem and reduce the bias, we correct the translation by the least-squares solution every time we evaluate the distance function. As the search narrows down the rotation, the translation residue is also getting smaller and so is the bias.

From the plot, we can see that the valley of the curve is within an interval of about 0.5 radians (30 degrees) in width. Beyond this interval, the curve is mostly flat. We require this initial interval to start the search procedure. However, in case the initial rotation error is very large or completely unknown, we can determine an initial interval by coarsely sampling the function at some rotation values. For example, we can sample the function at every 15 degrees (resulting in a total of 24 samples) and choose two adjacent points which have the lowest function values to form an interval. This effectively allows our method to handle arbitrarily large rotations.

The amount of translational error handled by our method depends on the threshold H_d . Usually the algorithm is good enough to handle the residuals of the odometry estimates.

3.8 Discussion

The rotation search/least-squares algorithm we presented in this chapter has the following properties. First of all, the algorithm directly matches points from one scan to points on the another. Thus it allows robot pose estimation in unknown environments. This is an improvement over Cox's method [29] which requires known environments. Our algorithm does not use an explicit linear model. Thus smooth curved shapes can be aligned in the same way as linear shapes. The algorithm is capable of handling a certain amount of sensing noise and occlusion.

An important point of the algorithm is that it uses a search procedure to minimize the distance function with respect to the rotation angle. An alternative approach would be to define a distance function (e.g. a sum of squared distances from point to tangent lines) with respect to both rotation and translation, and iteratively find least-squares solutions to all the variables. Because the distance function may not be smooth or convex, such an iterative algorithm may converge to wrong local minimum or not converge at all if the initial guess is poor. By excluding the most non-linear variable, the rotation angle, from the distance function, the convexness of the function near global minimum is much improved. Then, by formulating it as a search problem rather than relying on fixed-point iterations, our algorithm is more robust in finding the global optimum (we can sample the search space if necessary). On the other hand, we only search for the optimal rotation but solve for the translation in an embedded least-squares procedure. This is much more efficient than searching in all variables.

By including the costs for outliers in the distance function, our search procedure is able to determine a matching based on non-overlapping. For example, in a circular environment where part of the boundary is missing (Fig. 3.6), we can align two scans based on the missing region.

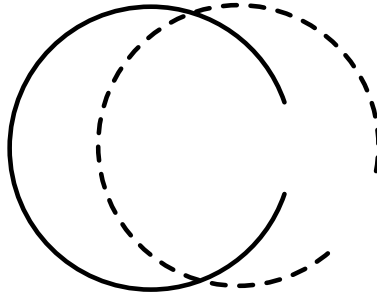


Figure 3.6: In an incomplete circular environment, it is possible to recover the rotation by aligning the two arcs using our search algorithm. But a least-squares method which does not use the non-overlapping information cannot solve this rotation because of symmetry.

However, an iterative least-square method which uses overlapping correspondence only cannot recover the rotation component because of the rotational symmetry.

We also mention that our method directly uses low-level sensor data instead of localized features in the matching process. Therefore, some problems with feature-based matching methods, such as unreliable feature extraction, insufficient number of features, or high complexity in feature correspondence, are avoided. As a price to pay, the algorithm requires an initial estimate in the relative translation between two scans. This initial estimate is typically available from odometry or may be derived from some global registration procedure.

Extensive experiments of matching range scans using this algorithm will be presented in Chapter 5. Both real range data and simulated scans will be used in the experiments.

Chapter 4

Point Correspondence Based Matching Algorithm

4.1 General Approach

In the previous chapter, we presented an algorithm for estimating the new robot pose with respect to a reference pose by matching the range scans taken at these two poses. We now present another method to solve the same problem. We call this method the point correspondence based matching algorithm as it uses point to point correspondences to align the scans. In Chapter 5, we will compare the two algorithms and combine them into a single two-staged method.

The point-based matching algorithm is iterative and is based on associating point to point correspondence. The idea of the method is the following. For each point P_i on S_{new} , we use a simple rule (independent of the actual rotation and translation) to determine a corresponding point P'_i on S_{ref} . Then from all the correspondence pairs of points, we compute a least-squares solution of the relative rotation and translation. This solution is applied to reduce the pose error between the two scans. We repeat this process until it converges.

The least-squares solution is derived by minimizing the following distance function which is defined on n pairs of correspondence points:

$$E_{\text{dist}}(\omega, T) = \sum_{i=1}^n |R_{\omega}P_i + T - P'_i|^2. \quad (4.1)$$

We can derive a closed-form solutions for ω and T as the following:

$$\omega = \tan^{-1} \frac{S_{xy'} - S_{yx'}}{S_{xx'} + S_{yy'}}$$

$$T = \bar{P}' - R_\omega \bar{P}$$

where

$$\begin{aligned} \bar{P} &= \frac{1}{n} \sum_{i=1}^n P_i & \bar{P}' &= \frac{1}{n} \sum_{i=1}^n P'_i \\ \begin{pmatrix} S_{xx'} & S_{xy'} \\ S_{yx'} & S_{yy'} \end{pmatrix} &= \sum_{i=1}^n (P_i - \bar{P})(P'_i - \bar{P}')^t. \end{aligned}$$

The central issue of the algorithm is then to define a sensible rule to determine correspondences without knowing the actual rotation and translation.

4.2 Rules for Correspondence

We describe the correspondence rules in the following sections. For convenience, we will regard the reference scan S_{ref} as a continuous curve and refer to it as the model. We will consider the correspondence search on a discrete scan later.

4.2.1 Closest-Point Rule

A commonly used rule is to choose the closest point on the model as the correspondence for a data point. We refer to this rule as the *closest-point rule*. Fig. 4.1(a) shows an example of finding the correspondences for a set of points in an elliptic model, using the closest-point rule. Besl and McKay described a general-purpose iterative closest point (ICP) algorithm for shape registration based on this rule and they proved that the ICP algorithm always converges monotonically to a local minimum with respect to the least-squares distance function [12]. We observe from experiments that, if the rotation is small, the ICP algorithm is good at solving the translation.

One disadvantage of the ICP algorithm is that it converges very slowly. Especially, when the model is curved, the correspondences found by the closest-point rule may contain little information about the rotation. As seen from Fig. 4.1(a), the vectors joining the correspondence pairs have very inconsistent directions and they tend to cancel out each other when used together to solve for the rotation. Moreover, regardless of the type of the model, the convergence speed of the algorithm is always very slow when the distance function approaches a local minimum.

To accelerate the ICP algorithm, Besl and McKay used a line search method to heuristically determine the transformation variables based on their values in two or three recent iterations. Although this improves the convergence speed near a local minimum, the problem of obtaining a

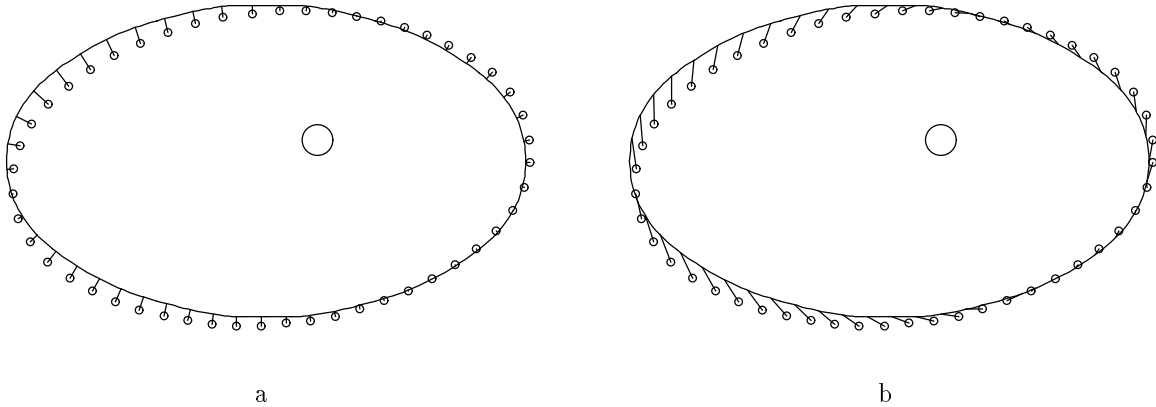


Figure 4.1: Use rules to determine correspondences (which are joined by line segments in the figure). (a) Use the closest-point rule; (b) use the matching-range-point rule. The model is the ellipse. The data points are labeled by small circles. The big circle in the center is the origin.

poor solution for the rotation component still exists. Therefore, the improvement in convergence speed is limited. Moreover, in order to apply the line search method, an implicit assumption is made about the smoothness of the least-square distance function. But this is typically not true if the number of correspondence pairs changes during the iterations (as a result of rejecting outliers).

4.2.2 Matching-Range-Point Rule

We propose a different rule which finds the correspondences in such a way that they significantly reveal the rotation component.

Consider a data point P and its corresponding point $P' = R_\omega P + T$. If we ignore the translation, we have $|P'| \approx |P|$. On the other hand, the polar angle θ of P and the polar angle $\hat{\theta}$ of P' are related by $\hat{\theta} \approx \theta + \omega$. This implies that the correspondence of P under a rotation is a point which has the same polar range as that of P , and the polar angles of the corresponding points differ by the rotation angle ω . Now in the presence of a small translation, we can still expect that the point P' with the same range as that of P is possibly a good approximation to the true correspondence of P , and this approximate correspondence provides rich information about the rotation angle ω .

To ensure that the rule finds a unique and reliable correspondence, we only search for the matching-range point within a local region of the model near P . Suppose that we can estimate a bound B_ω for the rotation ω , i.e. $|\omega| \leq B_\omega$. We have: $\hat{\theta} \in [\theta - B_\omega, \theta + B_\omega]$. This means that P' should lie within the sector bounded by $\theta \pm B_\omega$. Therefore, we propose the *matching-range-point rule* as the following: *For a data point P , its corresponding point is P' on the model where P'*

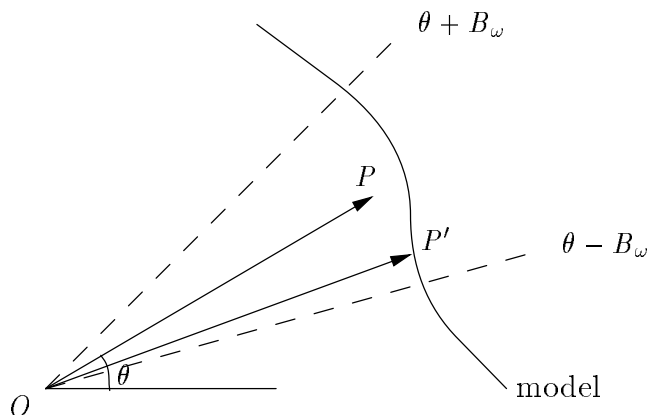


Figure 4.2: Matching-range-point rule: For a point P , the corresponding point P' on the scan lies within the sector and $|OP'|$ is closest to $|OP|$.

satisfies $|\hat{\theta} - \theta| \leq B_\omega$ and $|P'|$ is closest to $|P|$. The rule is illustrated in Fig. 4.2.

Fig. 4.1(b) shows an example of using the matching-range-point rule to find correspondences in the elliptic model. We can clearly see that the vectors joining the correspondence pairs consistently indicate the rotation direction. Therefore, the least-squares solution should be a good approximation to the true transformation, especially in the rotation component.

Based on this new rule, we design an iterative matching-range-point (IMRP) algorithm. In this algorithm, the parameter B_ω controls the size of the neighborhood to be searched for a correspondence and also the maximum rotation possibly solved in one iteration. Thus it will be best to choose B_ω to be close to the rotation residual at every iteration. We empirically generate B_ω using an exponentially decreasing function of the form: $B_\omega(t) = B_\omega(0)e^{-\alpha t}$. It is also possible to use the least-squares solution of ω in the current iteration to determine the value of B_ω for the next iteration.

A comparison of the performance of the ICP algorithm and the IMRP algorithm is illustrated in Fig. 4.3. We notice that the IMRP algorithm converges faster than the ICP algorithm in estimating the rotation. For the translation part, the IMRP algorithm is initially slower but it eventually converges faster than the ICP algorithm.

The reason that the translation residuals from the IMRP algorithm are initially reduced slowly is that the matching-range-point rule only tends to influence the translation component when the sector width B_ω becomes small enough. In the early iteration when B_ω is large, the algorithm tends to explain the displacement between the correspondence pairs by rotation rather than translation.

This phenomenon may present a potential problem to the stability of the algorithm if we need to dynamically reject outliers using a threshold, as good correspondences may be falsely rejected as outliers due to the incorrect translation. We will study more about outlier detection later.

4.2.3 Combining the Two Rules

It is desirable to combine the two rules in order to achieve both the convergence speed of the matching-range-point rule and the stability of the closest-point rule. We propose an *iterative dual correspondence* (IDC) algorithm which uses both rules, as the following:

1. In each iteration do the following steps:
2. For each data point P_i ,
 - (a) apply the closest-point rule to determine a correspondence point P'_i for P_i .
 - (b) apply the matching-range-point rule to determine a correspondence point P''_i for P_i .
3. Compute the least-squares solution (ω_1, T_1) from the set of correspondence pairs (P_i, P'_i) , $i = 1, \dots, n$ (which are obtained using the closest-point rule).
4. Compute the least-squares solution (ω_2, T_2) from the set of correspondence pairs (P_i, P''_i) , $i = 1, \dots, n$ (which are obtained using the matching-range point rule).
5. Form (ω_2, T_1) as the solution for the transformation in the current iteration.

The basic idea of the above algorithm is to take the translation component from the closest-point rule solution and the rotation component from the matching-range-point rule solution to form the current solution for the transformation.

The combination of the two rules appears to achieve significantly better results than each of the individual rules. It does not only ensure the stability of the iterations, but also increases the convergence speed significantly. In fact, the two rules reinforce each other by reducing different components of the transformation, so that each of them can be more effective in the following iteration. We notice that the IDC algorithm is insensitive to the choices of parameter B_ω .

Fig. 4.3 illustrates the residuals of the transformation components during the iterations of the three algorithms. The model and the data points in this example are the ones given in Fig. 4.1, where the initial rotation is -6 degrees; the initial translation is $(-5, 5)$ units (the width of the

elliptic model is 1000 units). Clearly, the IDC algorithm reduces the residuals much more quickly than the other two single-rule algorithms.

We experimentally estimate the rate of convergence for each of the three algorithms using the above example. The iterative closest point algorithm appears to have a sublinear convergence rate. The ratio of error residuals ($C = e_{i+1}/e_i$) gets larger and larger and it approaches to 1. In fact, after 30 iterations, C is at 0.998. For both the iterative matching-range point algorithm and the iterative dual correspondence algorithm, the rate of convergence seems to be linear. The error ratio for the iterative matching-range point algorithm is $C = 0.875$; the error ratio for the iterative dual correspondence algorithm is $C = 0.685$.

4.3 Matching Scans

We will use the IDC algorithm to register a new scan S_{new} to the reference scan (model) S_{ref} . Considering that S_{ref} is discrete, we need to interpolate in order to locate the correspondence point for each of the two rules.

First, we consider the matching-range-point rule. Let $P(r, \theta)$ be a point on S_{new} . We want to find another point $P'(\hat{r}, \hat{\theta})$ on S_{ref} according to the matching-range rule. Since the points in scan S_{ref} are parameterized in a polar coordinate system, we choose to linearly interpolate $1/r$ from θ between two points.

Let $P_1(\theta_1, r_1)$, $P_2(\theta_2, r_2)$ be two adjacent points, the interpolation function $\hat{r}(\hat{\theta})$ is:

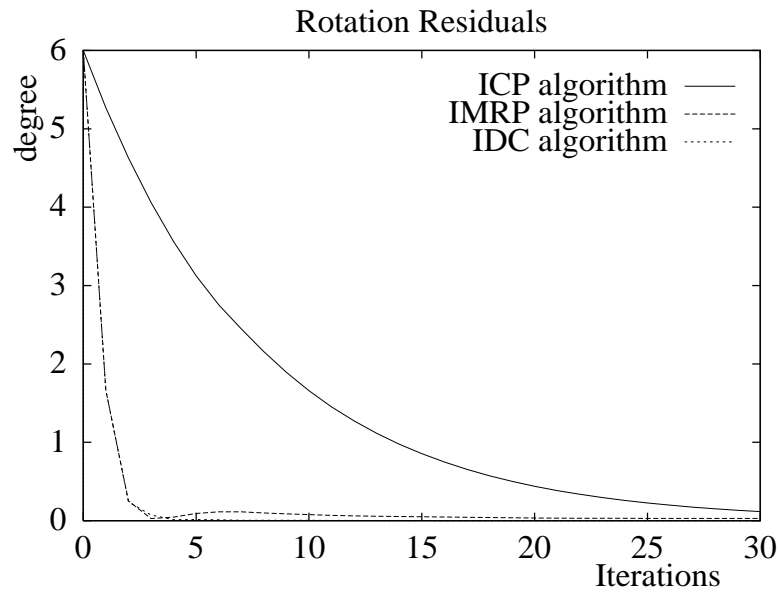
$$\begin{aligned} \hat{r} &= 1 / \left(\frac{1}{r_1} + \frac{\hat{\theta} - \theta_1}{\theta_2 - \theta_1} \left(\frac{1}{r_2} - \frac{1}{r_1} \right) \right) \\ &= \frac{r_1 r_2 (\theta_2 - \theta_1)}{r_1 (\hat{\theta} - \theta_1) + r_2 (\theta_2 - \hat{\theta})} \end{aligned} \quad (4.2)$$

This interpolation scheme is approximately equivalent to connecting the two points with a line segment if $|\theta_2 - \theta_1|$ is small. This can be seen by comparing Equation 4.2 with the equation of a straight line passing through P_1 and P_2 :

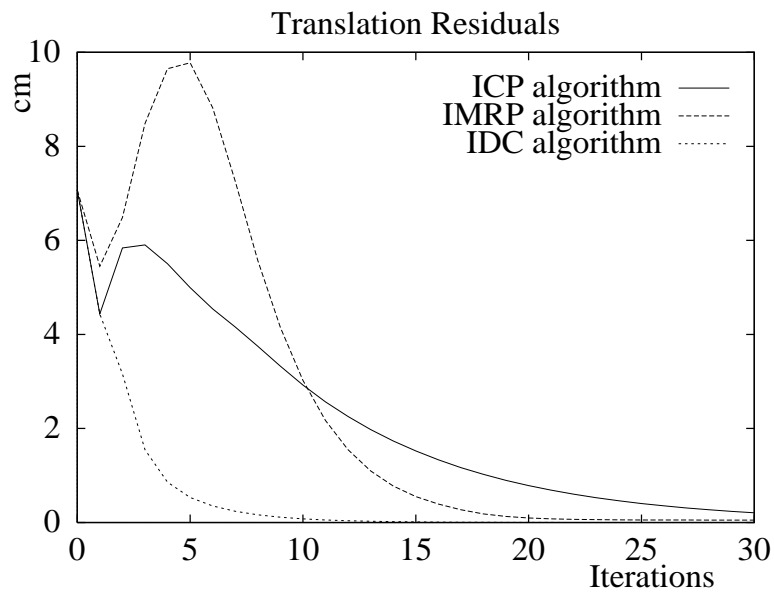
$$\hat{r} = \frac{r_1 r_2 \sin(\theta_2 - \theta_1)}{r_1 \sin(\hat{\theta} - \theta_1) + r_2 \sin(\theta_2 - \hat{\theta})}. \quad (4.3)$$

The interpolation in Equation 4.2 also has the advantage that the interpolated \hat{r} is a monotonic function of $\hat{\theta}$; it is easy to inversely compute $\hat{\theta}$ for a given \hat{r} .

The region where we need to search for P' consists of pairs of points whose angular intervals overlap with $[\theta - B_\omega, \theta + B_\omega]$. Let $[\theta_1, \theta_2]$ be one such interval. We find the intersection $[\hat{\theta}_1, \hat{\theta}_2] =$



a



b

Figure 4.3: Comparison of iterative algorithms. (a) Rotation residuals from the three algorithms over 30 iterations (note the curve for the IDC algorithm is barely visible as it is almost zero after three iterations). The initial rotation angle is 6 degrees. (b) Magnitudes of translation residuals. The magnitude of initial translation is 7.07 units. The model and data set are given in Figure 4.1.

$[\theta_1, \theta_2] \cap [\theta - B_\omega, \theta + B_\omega]$. Then we use Eq. 4.2 to interpolate two points $P'_1(\hat{\theta}_1, \hat{r}_1)$ and $P'_2(\hat{\theta}_2, \hat{r}_2)$ at the ends of the new interval. If $\hat{r}_1 \leq r$ and $\hat{r}_2 \leq r$, or $\hat{r}_1 \geq r$ and $\hat{r}_2 \geq r$, one of P'_1 and P'_2 (whose polar range is closer to r) is a candidate for P' . In this case, we still need to check other intervals for a potentially better candidate. But if $\hat{r}_1 \leq r \leq \hat{r}_2$, or $\hat{r}_1 \geq r \geq \hat{r}_2$, we can (inversely) interpolate for a P' which has the same range as P .

Now we consider the closest-point rule. For consistency with the matching-range rule, we slightly modify the closest-point rule such that it chooses the closest point P' within the sector $[\theta - B_\omega, \theta + B_\omega]$ only as the correspondence of P . The interpolation here is simply connecting two adjacent points with a line segment. In a similar way as in the search for a matching-range correspondence, we search all the angular intervals and find a P' by checking the distance from P to the line segments (or end-points).

The search required to determine all the correspondence pairs can be efficiently implemented in linear time with respect to the number of points on the two scans, by considering the fact that all the points in the scans are ordered by their angles.

4.4 Detecting Outliers

Due to occlusion, we may not find appropriate correspondence points for some data points. We need to identify these outliers and exclude them from the least-squares distance function. Assume that we have a bound B_r for the maximum displacement resulting from translation for a pair of corresponding points P and P' , i.e. $||P'| - |P|| \leq B_r$. We use this condition to test for outliers. For a pair of points found by one of the rules, we accept them as a correspondence pair only if the above condition is met. The threshold B_r can be selected as the k th largest distance among all the correspondence pairs, according to a predetermined constant fraction (p-tile).

4.5 Convergence of Iterative Algorithm

The iterative algorithm terminates when the change in the least-squares errors is sufficiently small. In practice, we find it sufficient to fix the number of iterations to about 15–20.

The iterative process in the algorithm resembles fixed-point iterations. Since the zero pose error is a stable fixed-point for the process, we expect that once the iteration converges that the final solution should be very accurate. To ensure convergence a good initial estimate (small initial pose error) is required by the algorithm. Usually, we can use odometry to provide this initial estimate.

Another possibility is to first apply the rotation search/least-squares method which we presented in Chapter 3 to obtain a relatively good registration before applying the point-based method. We find from experiments that this strategy usually guarantees the convergence of the point-based method and the point-based method also usually improves the accuracy of the solution.

4.6 Discussion

We have presented a new iterative dual correspondence (IDC) algorithm which uses both the closest-point rule and the matching-range-point rule to associate correspondences for data points. The cost of applying the matching-range-point rule is about the same as the cost of applying the closest-point rule. Therefore the complexity of one iteration of the IDC algorithm is about twice as that of one iteration of the ICP algorithm. However, the IDC algorithm converges much faster than the ICP algorithm, especially in solving the rotation component.

In next chapter, we will present experiments to demonstrate that IDC algorithm is effective in matching range scans. Later, in Part IV of the thesis, we will also show that the IDC algorithm can be used for registration of range images and planar image shapes.

Chapter 5

Scan Matching Experiments

5.1 Combining The Two Algorithms

We have presented two algorithms, the rotation search/least-squares algorithm and the iterative point correspondence algorithm, for matching range scans. The rotation search/least-squares is more robust as it uses tangent information in aligning the two scans. Using a search procedure instead of fixed-point iterations also allows the algorithm to robustly solve for the transformation even in the presence of a large initial pose error. However, because the matching distance function is not smooth and we only use a simple one-dimensional search procedure to do the minimization, the solution given by this algorithm may not be highly accurate. On the other hand, the iterative point correspondence based algorithm gives a more accurate solution through its fix-point iterations, as long as it converges.

We choose to sequentially combine the two algorithms into a two-staged algorithm. In the first stage, we apply the search/least-squares algorithm to reduce a possibly large initial pose error. In the second stage, the estimation result from the first stage is used as the initial guess for the point correspondence based algorithm. Since the result from the first stage is usually quite good, the convergence in the second stage can be ensured. In the meanwhile, an improvement in estimation accuracy can be achieved from the second stage.

In this chapter, we will present experimental results of matching scans using this two-staged algorithm.

5.2 Sensing Strategy

If we have control over the sensing directions (as is the case with the ARK sensor [96]), we can choose to take a range scan in such a way that the sampling points are evenly distributed in space. In other words, we want the distances between adjacent points to be approximately equal. This is more efficient than a uniform sampling of orientation where the interval width is proportional to the ranges of the points from the sensor.

Assume that the current sample point is $P(\theta, r)$. We will decide the direction of the next shot, $\theta + \Delta\theta$, such that the new point is a fixed distance d away from P . If we approximate the contour curve near P by its tangent line (which has parameter (ρ, ϕ)), the increment in sensing direction is:

$$\Delta\theta = \arctan \frac{d \cos(\theta - \phi)}{r + d \sin(\theta - \phi)}, \quad (5.1)$$

where the angle $\theta - \phi$ can be computed by the following:

$$\theta - \phi = \arctan\left(\frac{1}{r} \frac{dr}{d\theta}\right). \quad (5.2)$$

Note that the derivative $\frac{dr}{d\theta}$ can be estimated from recent range measurements. Combine the above two equations and take a first order approximation (for a small d), we get:

$$\Delta\theta \approx \frac{d}{\sqrt{r^2 + \left(\frac{dr}{d\theta}\right)^2}}. \quad (5.3)$$

To avoid infinitely small increments, we can predefine a minimum value for $\Delta\theta$. Now $\theta + \Delta\theta$ is the direction to take the next measurement.

This sensing strategy is used to generate both simulated and real scan data in our experiments.

5.3 Experiments with Simulated Data

Range measurements are simulated for various environments in order to test the matching algorithms. We model the environments with lines and spline curves. Sensing noises are modeled as additive random variables. For a given environment model, we can simulate a range scan from any robot pose.

We present a series of experiments using simulated range data. In each experiment, we choose an environment model and two poses, P_{ref} and P_{new} , from which to take the scans S_{ref} and S_{new} . The pose error is generated randomly where ω is uniformly distributed in $[-0.25, 0.25]$ radians

($\pm 14.3^\circ$) and T is uniformly distributed in a disk of radius 50 centimeters (comparing with the dimension of the environment of 10 meters). Sensing noise is assumed to be uniformly distributed. We vary the maximum sensing noise in each experiment.

In each experiment setting, the matching process was run 1000 times (with randomly generated initial pose error and sensing noise as described above). We computed the standard deviations (from the theoretical zero means) of the residuals of the variables (ω , T_x , and T_y) resulting from each matching process. The standard deviations of the residuals after the first stage (rotation search algorithm) and after the second stage (iterative point correspondence algorithm) were recorded. In the matching process, the number of iterations was fixed at 15 in each of the two stages.

The results from six experiments are listed in Table 5.1. In each experiment, we list the environment model, the maximum sensing noise, and the standard deviations of the residuals. We also plot all the translation residuals from each of the six sets of experiments (Fig. 5.1). Each translation residual is shown as a dot in the x-y plane. The dots form a cluster around the origin.

We have some observations from the statistical results. (1) The pose error residuals are very small compared with the sensing noise. The standard deviation of translation residuals is about 15% to 25% of the standard deviation of sensing noise. The rotation residuals are well within one degree. With a typical sensor accuracy of 5cm, the rotation residuals are within one tenth of a degree. (2) The algorithm behaved robustly over thousands of runs. (3) The second algorithm (iterative point correspondence algorithm) gave significantly more accurate estimates of rotation than the first algorithm. But the two methods are equally good in finding the translations (the second algorithm is slightly better when the sensing noise is relatively small or when the environment is mostly linear (experiments 1, 2, 3)). (4) The residuals appear to be normally distributed with approximately zero means. (5) When the sensing noise becomes large (experiments 5 and 6), the algorithm degrades gracefully. However, experiment 6 (where the sensing noise is set to 20cm) appears to have reached the limit of the algorithm. We experienced a few failure cases where the iterative algorithm did not converge. There were about 10 such cases among the 1000 runs. The failure is partly caused by the step of rejecting outliers using thresholds. Raising the threshold level may improve convergence. But it may result in bias in the estimation because of outliers. High sensing noise also leads to bad estimation of local contour directions. Thus the correspondence association step (which involves interpolation) is subject to large error.

We plot three typical matching examples from the above experiments. In each example, we show the two scans which are misplaced with respect to each other. Then the same two scans after

alignment are shown. In the figures, the reference scan is plotted with x's. The big X at the center is the actual pose P_{ref} of the reference scan. The new scan to be registered is plotted with circles. The big circles with arrows are the poses of the new scan before and after registration (that is, P'_{new} and P_{new}).

Example 1 (Fig. 5.2) is a typical case from experiment 3. Here the rotation is 0.25 radians (14.3°) and the translation is $(-30, -20)$ centimeters. Notice that there is a substantial portion of the world (about one third) which is visible only in one scan but not in the other. Our algorithm is able to ignore the unmatched parts but successfully perform the matching based on the remaining part of the world which is visible in both scans.

Example 2 (Fig. 5.3) is from experiment 4, in which the environment consists of mostly curves (rotation 0.5 radians (28.6°), translation $(-60, 40)$ centimeters). There also exist occlusions in the scans. Our matching method is successful.

Example 3 (Fig. 5.4) is chosen from experiment 6, in which the maximum sensing noise is as large as ± 20 centimeters. Rotation is -0.2 radians (-11.5 degrees) and translation is $(-40, 30)$ centimeters. The correct alignment is found by the algorithm in this example. Note this level of sensing noise is close to the limit of algorithm capacity.

In the above experiments, the second algorithm (point correspondence based method) is not fairly tested as its input (from the output of the first algorithm) is too good. Here we present another experiment which examines the iterative point correspondence algorithm without first applying the rotation search method. We use the same environment as the one in experiment 2, but set the maximum sensing noise to ± 10 cm. The initial pose error is randomly generated so that ω is uniformly distributed over $[-0.1, 0.1]$ radians and T is uniformly distributed in a disk of radius 20 centimeters. We ran this experiment 1000 times. The standard deviations of the residuals are listed in Table 5.2. For comparison, we also run the rotation search algorithm using the same input set. We can see that the second algorithm is still successful despite of the relatively larger initial pose errors. Again, the second algorithm performs significantly better than the first one in finding the rotation, but only slightly better in finding the translation.

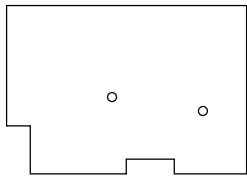
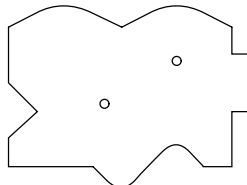
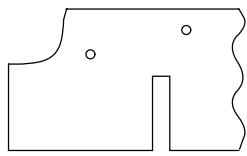
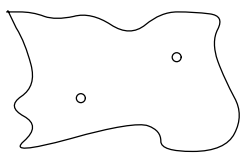
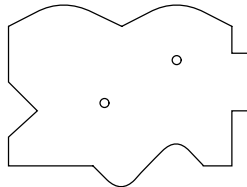
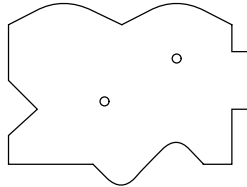
No.	Simulated Environments and Maximum Sensing Noise	Residual Standard Deviations		
			After Stage 1	After Stage 2
1	 noise: $\pm 5\text{cm}$	σ_ω	0.2786 °	0.0547 °
		σ_x	0.3909 cm	0.3418 cm
		σ_y	0.4531 cm	0.2702 cm
2	 noise: $\pm 5\text{cm}$	σ_ω	0.3668 °	0.0754 °
		σ_x	0.4150 cm	0.3592 cm
		σ_y	0.3886 cm	0.3146 cm
3	 noise: $\pm 10\text{cm}$	σ_ω	0.4414 °	0.1876 °
		σ_x	1.3449 cm	1.0436 cm
		σ_y	1.5336 cm	0.8532 cm
4	 noise: $\pm 10\text{cm}$	σ_ω	0.3970 °	0.1824 °
		σ_x	0.8723 cm	0.9535 cm
		σ_y	0.7836 cm	0.8446 cm
5	 noise: $\pm 15\text{cm}$	σ_ω	0.6090 °	0.3027 °
		σ_x	1.2268 cm	1.2604 cm
		σ_y	1.1269 cm	1.1438 cm
6	 noise: $\pm 20\text{cm}$	σ_ω	1.1517 °	0.6230 °
		σ_x	2.2832 cm	2.5478 cm
		σ_y	2.1961 cm	2.1811 cm

Table 5.1: Statistics of experiments in simulated environments. Maximum initial rotation and translation are set at $\pm 14.3^\circ$ and 50cm, respectively.

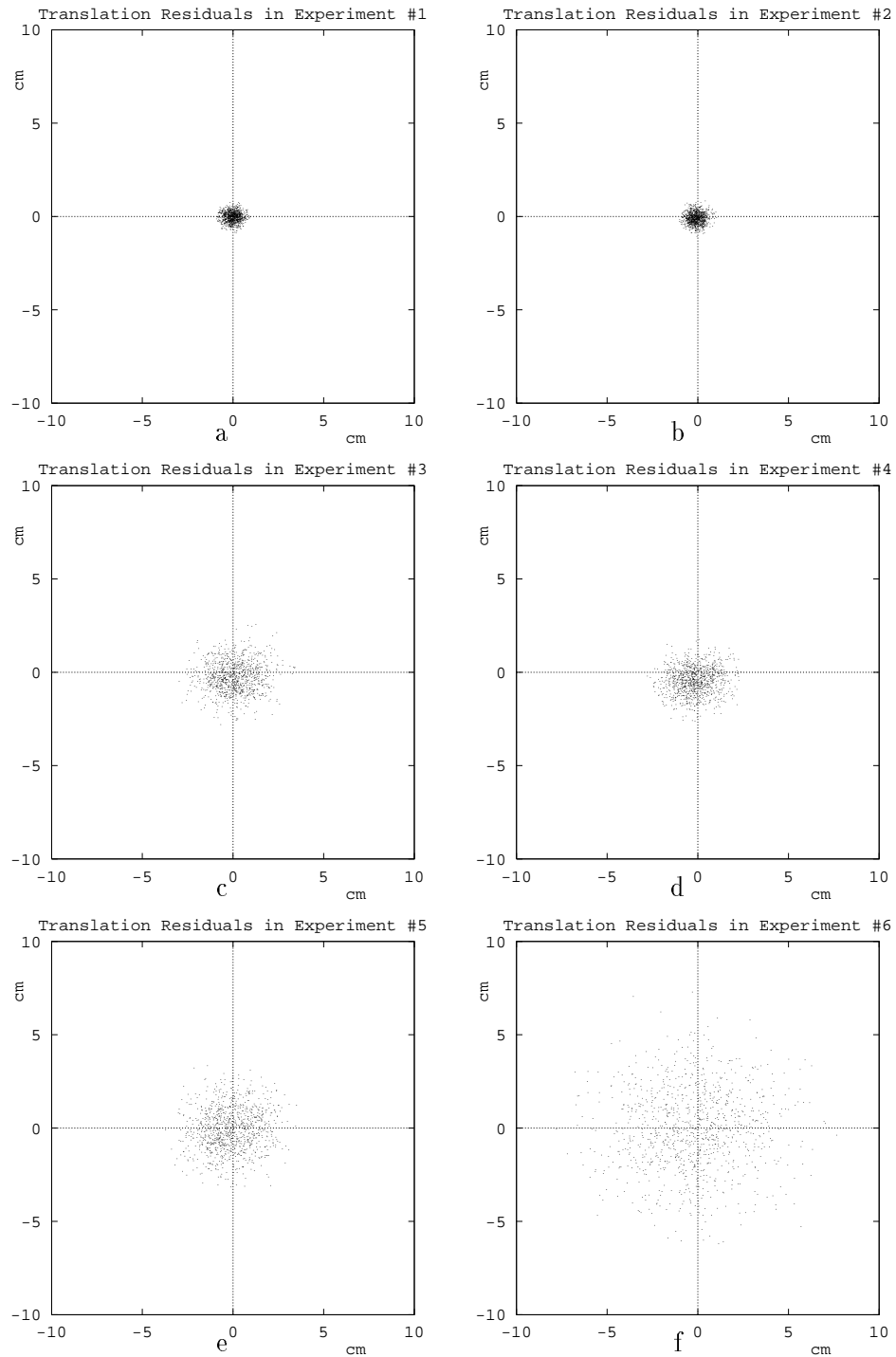


Figure 5.1: Translation residuals after apply the matching algorithm. (a) to (f) correspond to experiments 1 to 6 in table 1. There are 1000 samples in each experiment. Larger spread of dots in the plot corresponds to higher levels of sensing noise.

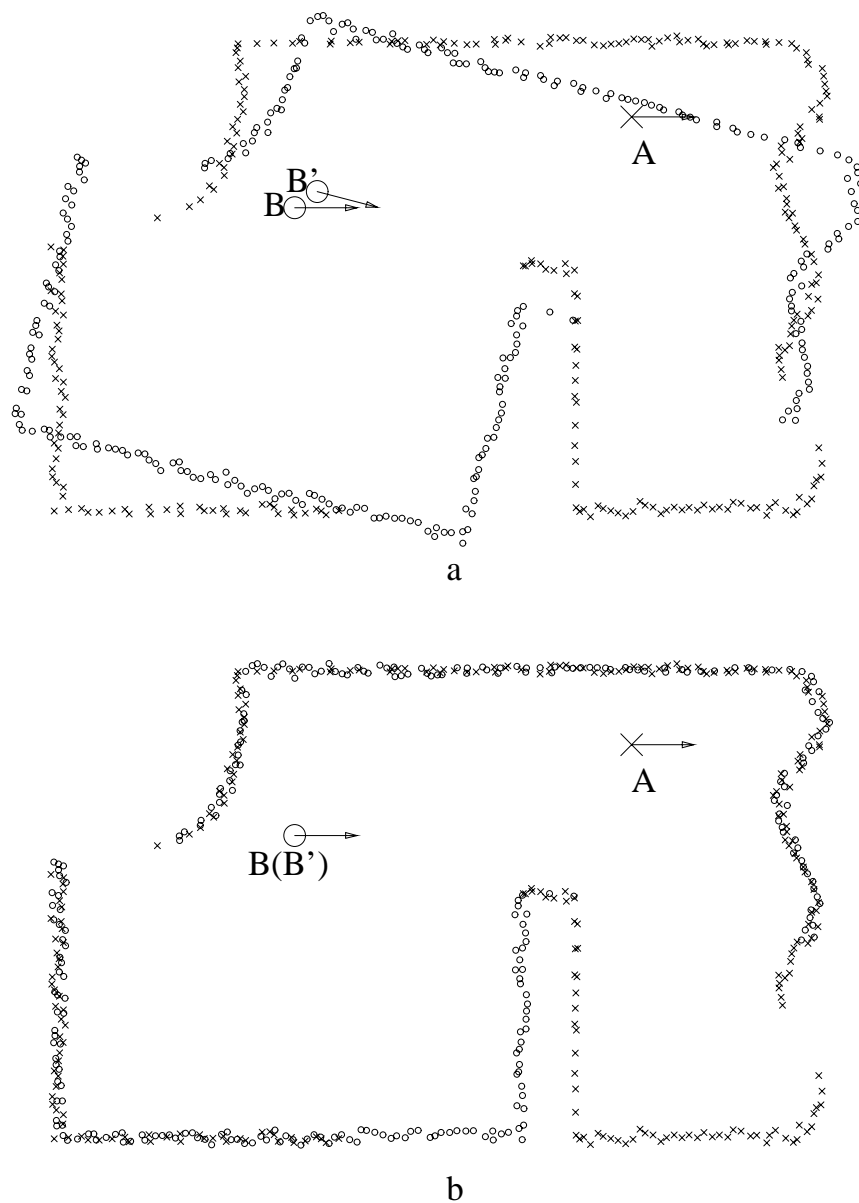


Figure 5.2: Example 1 (from experiment 3, maximum sensing noise is 10cm). The robot went from pose A to pose B in reality, but due to odometry errors, it thinks it went to pose B' . Points on scans from pose A and B are labeled as x's and small circles respectively. The poses are indicated by arrows. Part (a) shows the alignment error due to the difference between pose B' and true pose B . Part (b) shows the result of aligning the two scans. The pose B' is corrected to the true pose B by the same transformation. Notice the large occlusion in the lower right corner.

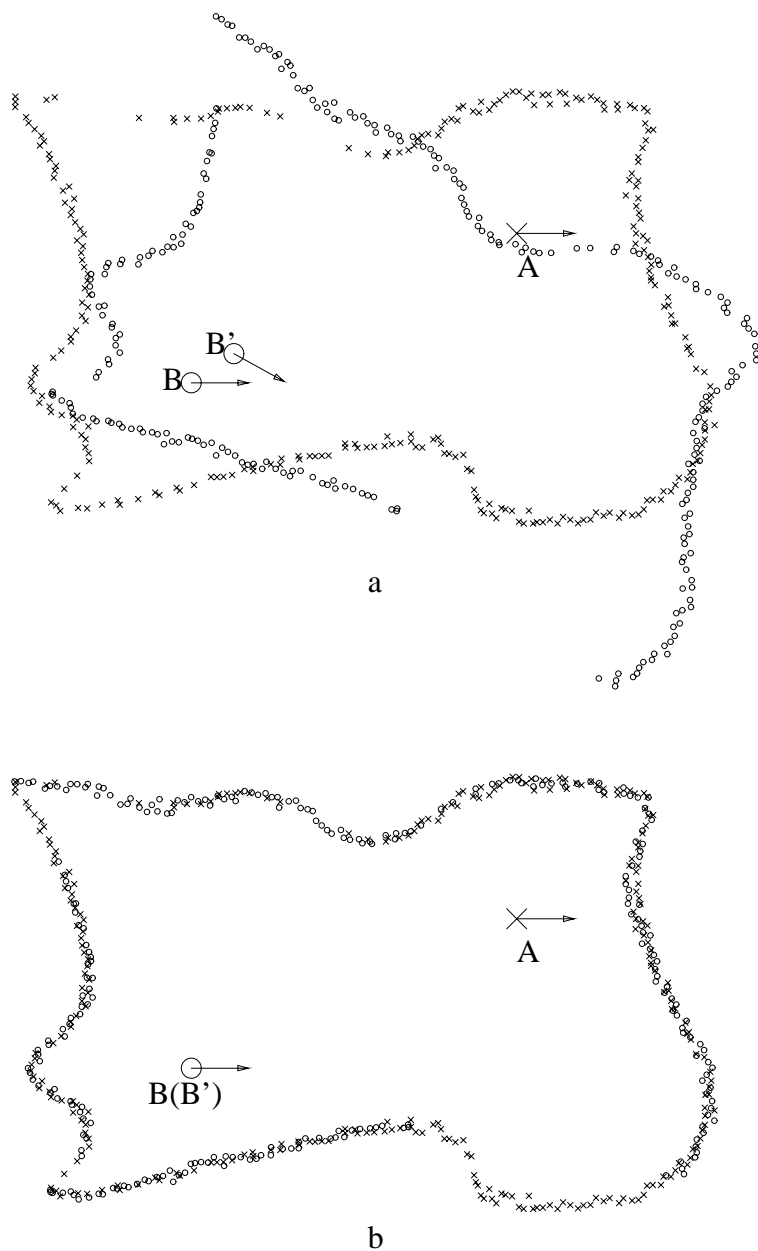


Figure 5.3: Example 2 (from experiment 4, maximum sensing noise is 10cm). Part (a) shows the pose error (indicated by the difference from B' to B) and the resulted alignment error of the scans. Part (b) shows the result of correcting the pose error and aligning the two scans. The large rotation and translation are corrected successfully. There is occlusion in the scan.

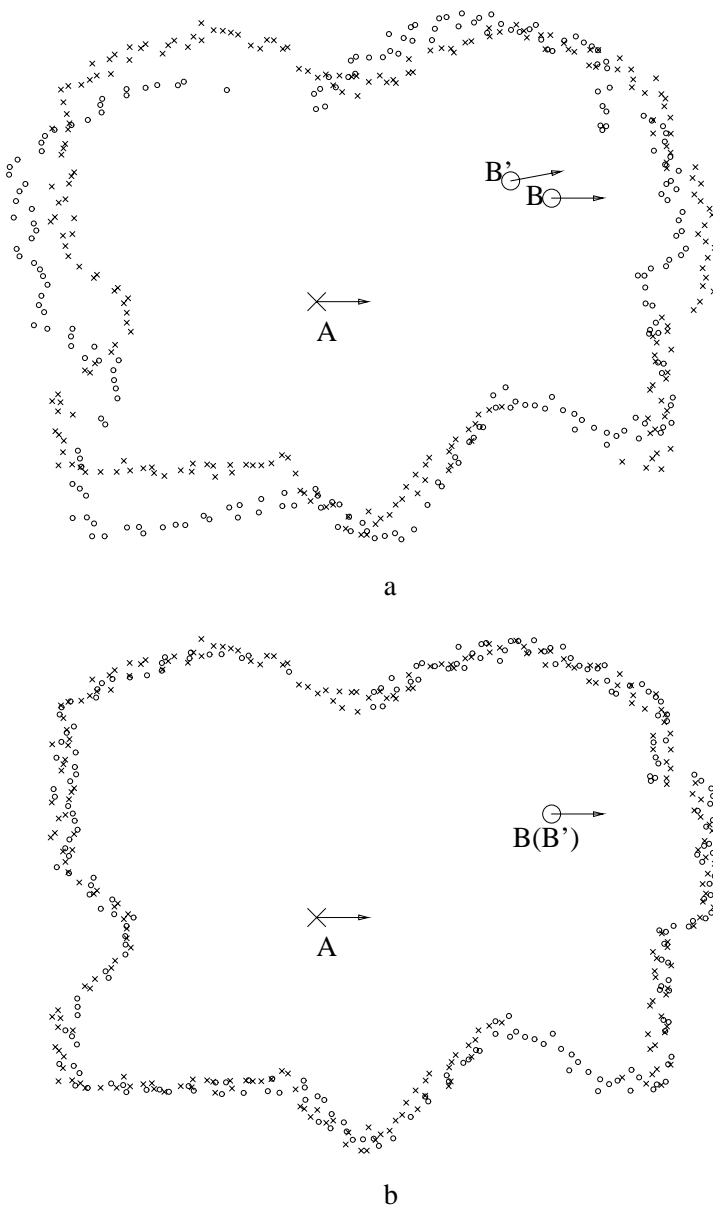


Figure 5.4: Example 3 (from experiment 6, maximum sensing noise is 20cm). Part (a) shows the pose error (indicated by the difference from B' to B) and the resulted alignment error of the scans. Part (b) shows the result of correcting the pose error and aligning the two scans. This level of sensing noise is about the limit of algorithm capacity.

Residual		First Algorithm	Second Algorithm
Rotation	σ_ω	0.5375 °	0.1599 °
Translation x	σ_x	0.7652 cm	0.7827 cm
Translation y	σ_y	0.7998 cm	0.6514 cm

Table 5.2: Standard deviations of the residuals from the two algorithms when they are run independently with the same set of input. Maximum sensing noise is ± 10 cm. Maximum initial rotation and translation are $\pm 5.7^\circ$ and 20cm, respectively.

5.4 Experiments with Real Data

5.4.1 Experiments Using The ARK Robot

The ARK project investigates techniques for robot navigation in structured indoor environments [96]. The ARK robot is equipped with a laser rangefinder (model Optech G150) mounted on a pan and tilt unit on the robot platform. The accuracy of the rangefinder is 4 centimeters. We used the ARK robot to collect a few range scans in the Vision Lab at the Department of Computer Science, University of Toronto. The testing environment is illustrated in Fig. 5.5.

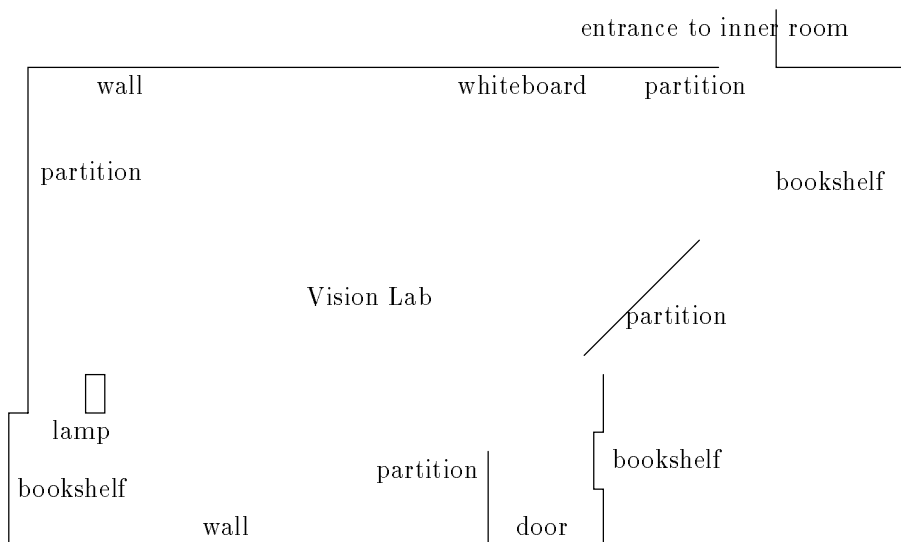


Figure 5.5: Testing environment in the Vision Lab at the Department of Computer Science, University of Toronto.

We present two examples of matching scans from this environment (Figures 5.6 and 5.7). Relatively large pose errors between the two scans were intentionally introduced. Unfortunately, the

ground truth for this experiment is not available as the robot was driven by hand. Our matching algorithm successfully aligned the scans in both examples. Notice that in performing the matching the algorithm is able to ignore the cluttered areas at the right-side wall and the bottom left corner (which are bookshelves). The algorithm also handled the occluded area well.

5.4.2 Experiments at FAW, Germany

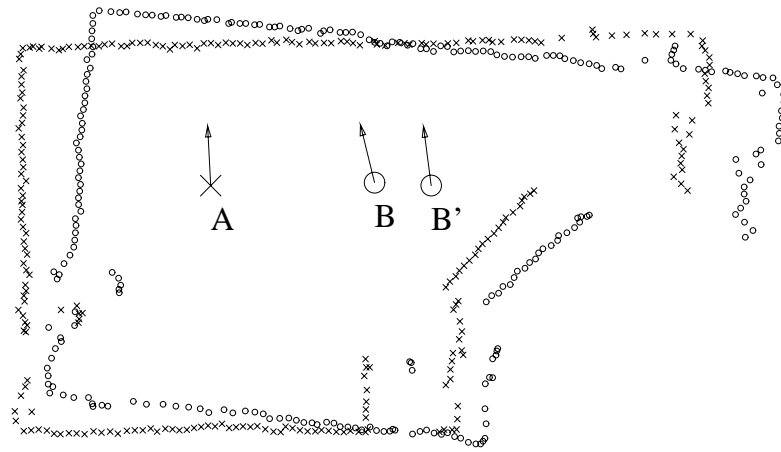
The testing environment was the cafeteria and nearby corridor in FAW, Ulm, Germany. The laser rangefinder used was a Ladar 2D IBEO Lasertechnik mounted on the AMOS robot. The laser sensor has a maximum viewing angle of 220 degrees, and an accuracy of $\pm 20mm$. We obtained 30 scans which were taken by the robot along its path at a pace of approximately 2 meters between scan poses.¹ We then selected 84 pairs of scans which have sufficient overlap and applied the scan matching algorithm to each pair. Our matching algorithm successfully aligned the scans in all 84 cases. All of these scan matching results will be used by a global scan registration algorithm which is discussed in Chapter 6. An illustration of the testing environment and the a plot of all the scans are given in Chapter 7.

Three examples of the matching results are shown in Figures 5.8 to 5.10. In all examples, we exaggerated the pose errors for testing the algorithm. The odometry error from the actual robot is much smaller. Note that this robot samples orientation uniformly when taking the scans, as opposed to our early strategy of making the sample points uniformly dense. But the matching algorithm still works well with this kind of scan.

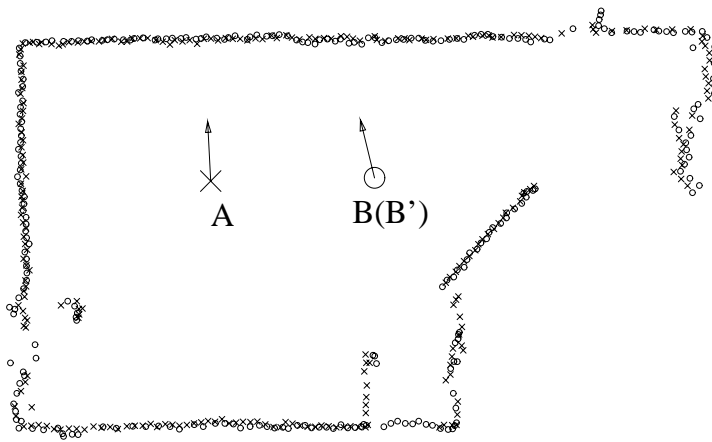
We also include another example using real range data from a hallway (Figure 5.11). Notice that the constraints along the hallway direction is weak (the only reference are several shallow doorways). Our algorithm is able to derive the correct matching.

It appears that the real sensor data are more accurate than our typical simulated data. After thoroughly testing our algorithm with simulations, we believe that it should also work well in real indoor environments of this type. Moreover, in a typical indoor robot course where range measurements are taken at short discrete steps, the pose errors are usually much smaller than the ones we simulated. As a conclusion from the experiments, we believe that our algorithm should work robustly in real-life robotic applications.

¹The author would like to thank Steffen Gutmann, Joerg Illmann, Thomas Kaempke, Manfred Knick, Erwin Prassler, and Christian Schlegel from FAW, Ulm for collecting range scans and making the data available for our experiments.

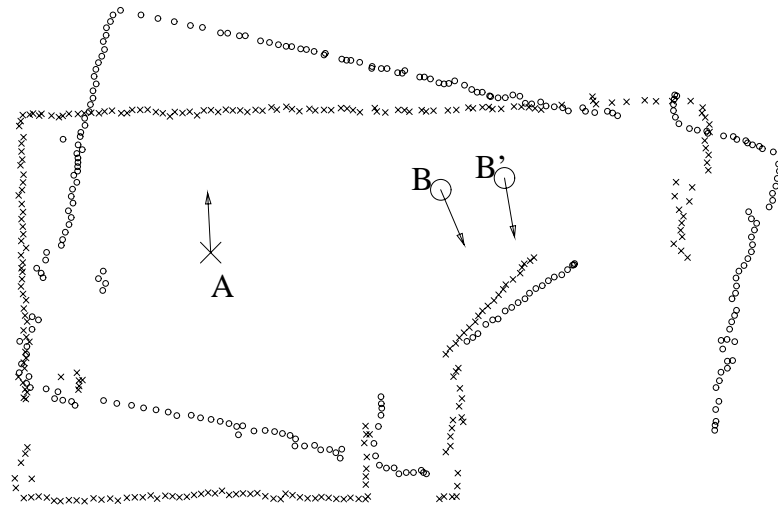


a

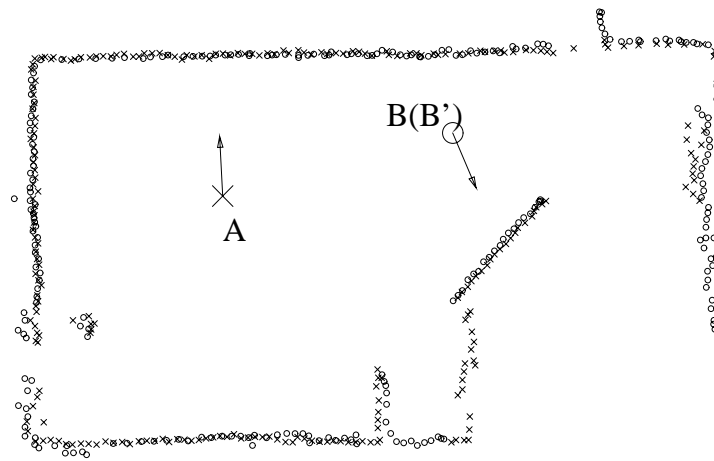


b

Figure 5.6: Example of matching scans taken by the ARK robot at U of T.



a



b

Figure 5.7: Another example of matching scans taken by the ARK robot.

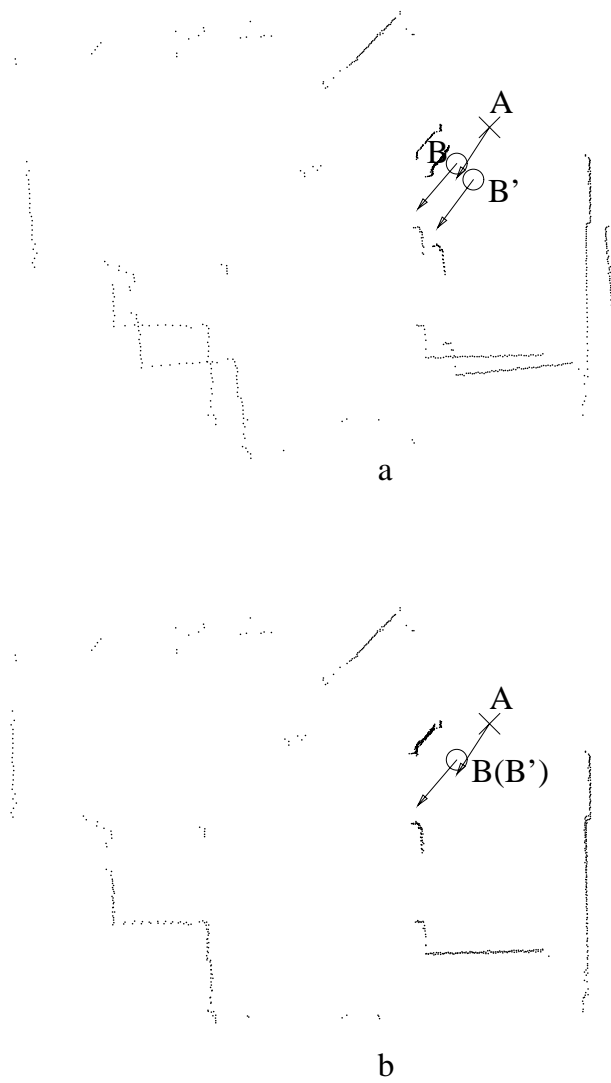


Figure 5.8: Example of matching real scans at FAW, Ulm, Germany. The scans are shown in dots. Part (a) shows the pose error (indicated by the difference from B' to B) and the resulted alignment error of the scans. Part (b) shows the result of correcting the pose error and aligning the two scans.

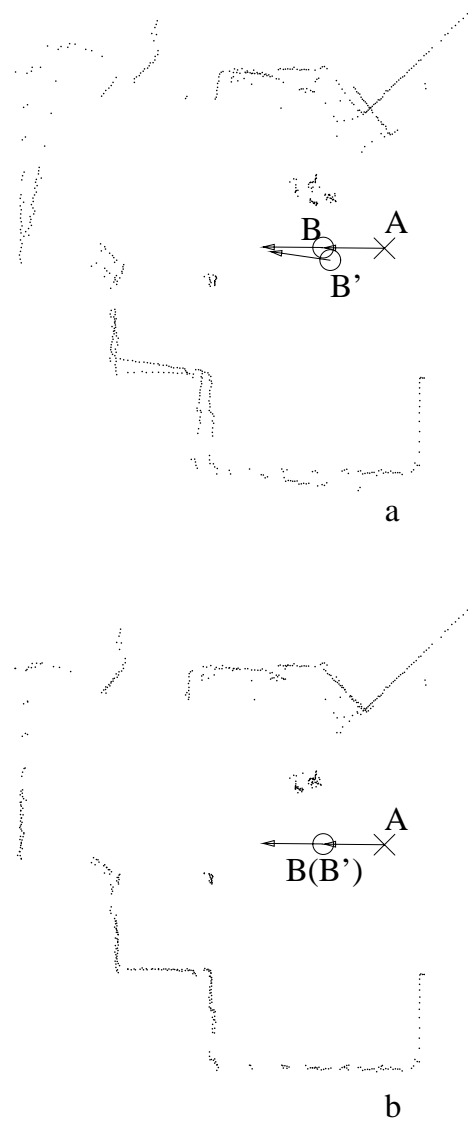


Figure 5.9: Another example of matching real scans at FAW, Ulm, Germany.

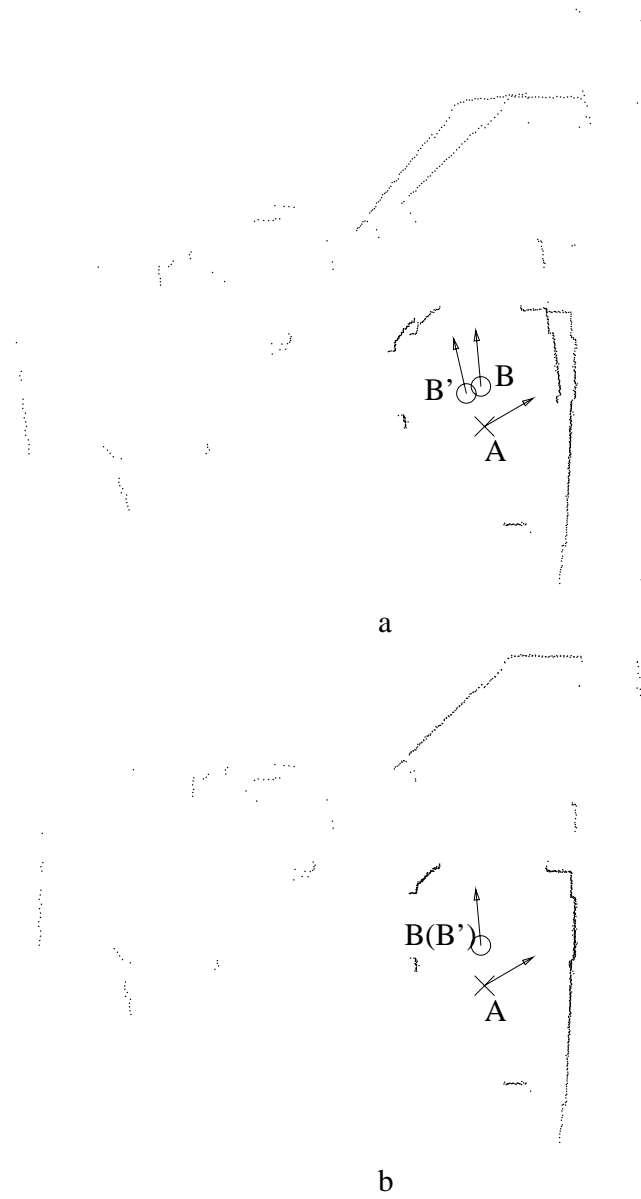


Figure 5.10: Another example of matching real scans at FAW, Ulm, Germany.

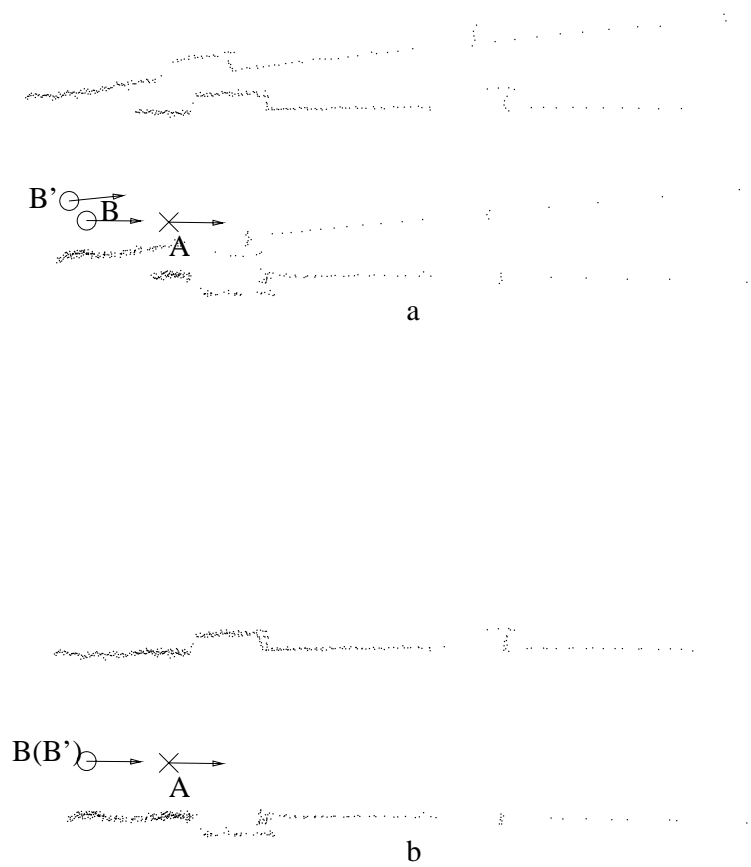


Figure 5.11: Example using real data from a hallway. Notice that the constraints are weak along the hallway direction. Alignment still can be made using our algorithm.

5.5 Comparison with Other Methods

In this section, we present experiments to demonstrate another algorithm which is based on an extension to the method used by Cox [29]. We will comment on its performance as compared to our algorithms.

The original Cox's method assumes a known piecewise linear model of the environment. We first summarize the algorithm as the following.

Given a set of data points and a model which consists of a set of line segments. Assume that the data points is misplaced with respect to the model by a small pose error (consisting of a rotation $R(\theta)$ and a translation $T = (T_x, T_y)^t$). For every point \mathbf{v}_i , the line segment m_i in the model which is closest to \mathbf{v}_i is selected as the corresponding target. Let \mathbf{u}_i be a point on m_i and let \mathbf{n}_i be the normal direction of m_i . The normal distance from a transformed point to its target line is given by:

$$d_i = |(R(\theta)\mathbf{v}_i + T - \mathbf{u}_i) \cdot \mathbf{n}_i|. \quad (5.4)$$

For all the correspondence pairs, a fitting error is defined as the sum of the squared normal distances:

$$E = \sum_i [(R(\theta)\mathbf{v}_i + T - \mathbf{u}_i) \cdot \mathbf{n}_i]^2. \quad (5.5)$$

We want to solve for the variables θ and T which minimize the error function. If the rotation angle θ is small, the trigonometric functions in $R(\theta)$ can be approximated as

$$\sin \theta \approx \theta; \quad \cos \theta \approx 1.$$

After this approximation, d_i becomes a linear function in θ , T_x , T_y and the variables can be easily solved using linear least-squares.

The derived transformation $(R(\theta), T)$ is applied to the points so that they are better aligned with the model. This procedure is iterated until convergence.

To make it robust against outliers, the point-target correspondence pairs are checked against a threshold. If the distance from a point to its target segment is too large, the term from this correspondence is not included in the summation in Eq. 5.5.

The above algorithm is not directly applicable to pose estimation in unknown environments since we only have a scan of data points as model instead of a set of line segments. In an effort to extend the algorithm for matching two scans, we first derive a set of line segments by fitting to the points in the reference scan. The following adaptive fitting procedure is used:

1. Divide the scan of points into subsets at “breaking points” which are near “corners” or depth discontinuities. Fit a line to each subset of points. (The detection of such breaking points and the procedure of line fitting is discussed in Section 3.4.)
2. For each subset of points, if the fitting error is too large, split the points into two parts. The breaking point is chosen in such a way that the sum of fitting errors in two parts is minimum among all possible ways of dividing the set.
3. Recursively apply step 2 to further subdivide the set of points until the fitting error is small for every subset of points.

The result from the above procedure is an ordered set of line segments which can be considered as a model of the environment. An example of line fitting to a simulated scan of points is shown in Fig. 5.12. Once this model is derived, another scan of points can be matched to the model using Cox’s algorithm.

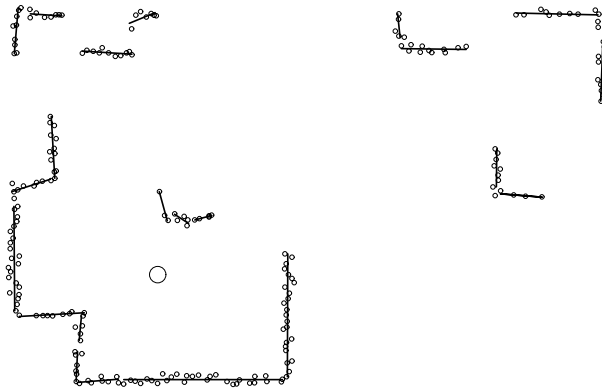


Figure 5.12: Fitting line segments to a set of points.

Note that the derived line segments are oriented as we know their directions based on the ordering of the original points. Some line segments which represent edges facing away from the sensor should not be visible from the current position. They should be deleted from consideration in correspondence association. We can distinguish this type of lines by checking if the polar angles of

the starting and the end points of a line segment is out of order (with respect to the center of current scan). Thus we can avoid associating points on a wall to a line representing the opposite side of the wall. Also note that we can use the information about the ordering sequence of the line segments to speed up the correspondence search. Instead of the normal $O(nl)$ time, we can determine all correspondences in $O(n + l)$ time (n, l being the number of points and lines, respectively).

To test the extended Cox's method, we run experiments using simulated range scans (similar to what we did in Section 5.3). Experimental results show that this algorithm is very effective. First, the fitting procedure generates a good linear approximation of the environment. Although the derived line segments contain errors in position and orientation (because of error in the points), the set of lines as a whole provides adequate constraints to register the new scan points. The iterative algorithm appears to converge very quickly. It typically takes no more than 10 iterations to achieve sufficient accuracy.

A typical example of matching a scan of points to a set of line segments (derived from another scan of points) is shown in Fig 5.13. We repeatedly test the algorithm with random initial pose error and random sensing noise. Several simulated environments are tested. For the environment as in Fig 5.13, we run the matching experiment 1000 times and compute the standard deviations of the residuals in the rotation and translation components (as shown in Table 5.3). The maximum sensing noise is set at ± 10 cm. Initial rotation error is uniformly distributed over $[-0.1, 0.1]$ radians and initial translation error is uniformly distributed in a disk of radius 20 cm (note: the width of the simulated environment is 10 meters).

The standard deviations of residuals indicate a high accuracy in the matching results. The rotation error residual is well under one degree. The translation error residual is about 20% of the maximum sensing noise. Also note that the two scans only partially overlap because of occlusion. The algorithm is able to ignore the non-overlapping regions (outliers).

For comparison, we run our point correspondence based algorithm (discussed in Chapter 4) with the same set of input. The standard deviations of residuals are also computed and listed in Table 5.3. It appears that the two algorithms have very similar accuracy. The computational costs for the two algorithms are also similar (Cox's method is slightly simpler in correspondence search but it requires an extra step of model fitting). This leads us to believe that the two algorithms can be used interchangeably for matching a pair of scans.

It should be noted that the Cox's method requires a good initial estimate to ensure convergence (so does the point correspondence based algorithm). We notice that the Cox's algorithm tolerates

Residual		Point-Line Algo.	Point-Point Algo.
Rotation	σ_ω	0.2277 °	0.1992 °
Translation x	σ_x	1.8281 cm	1.6375 cm
Translation y	σ_y	1.2338 cm	1.3128 cm

Table 5.3: Standard deviations of the residuals from the extended Cox’s algorithm (Point-Line) and point correspondence based algorithm (Point-Point) when they are run with the same set of input (1000 samples). Maximum sensing noise is ± 10 cm. Maximum initial rotation and translation are $\pm 5.7^\circ$ and 20cm, respectively.

about 10° of rotation error (for the testing environment in Fig. 5.13), but a larger rotation error sometimes leads to divergence. The threshold for detecting outliers in correspondence seems to influence the amount of rotation and translation error the algorithm can handle. If larger initial rotation error is possible, our rotation search algorithm (discussed in Chapter 3), which is more robust against large rotation error, can be employed. This algorithm is formulated as a search for the minimum of a distance function rather than fixed-point iterations. A good initial guess can be obtained by sampling the search space. Thus large (unlimited) rotation errors can be handled.

To summarize, the extended Cox’s algorithm is very effective in matching two scans of points. Our point correspondence based algorithm is an equally good alternative. If large initial rotation error is present, the rotation search algorithm can be first applied to find a good starting point.

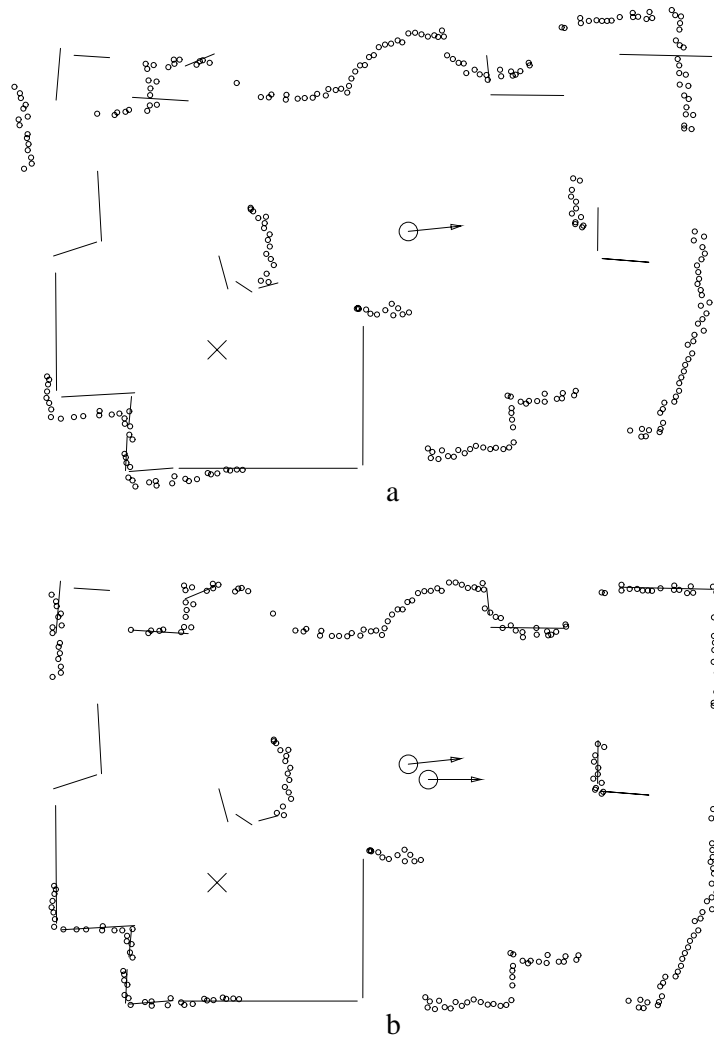


Figure 5.13: Matching a scan of points to line segments which are derived from another scan. (a) Before matching; (b) after matching and correcting pose error.

Part III

Consistent Registration of Multiple Range Scans for Robot Mapping

A robot exploring an unknown environment may need to build a world model from sensor measurements for its navigation task. In order to integrate all the frames of sensor data, it is essential to align the data properly according to the poses of the robot at which the data are collected. In other words, all the data must be consistently registered in a common reference frame. Note that the robot poses (at which the data are collected) are not known exactly and have to be estimated from odometry information and the sensor measurements.

In this part of the thesis, we study the problem of consistent registration of multiple frames of range measurements. Also to be addressed are the related issues of global pose estimation and the representation and manipulation of spatial uncertainties.

In Chapter 6, we formulate a procedure for consistent data registration based on the optimal estimation from a network of measurements in the sense of maximum likelihood criterion. We uniformly model pose constraints from both matching range scans and odometry measurements.

In Chapter 7, we present experiments of consistent data registration with both simulated and real examples. We also discuss some alternative approaches to this problem.

A condensed version of this work has been previously reported in [86].

Chapter 6

Consistent Registration Procedure

6.1 Problem Description and Approach

The general problem we want to solve is to let a mobile robot explore an unknown environment by collecting range measurements and building a map of the environment from sensor data.

Assume that the robot takes scans of range measurements, each representing a partial view of the world, at various locations along its exploration trajectory. The major issue of map building is the registration and integration of all these partial views in a global reference frame (i.e. to place all the scans in the same coordinate system so that they can be merged). This is a difficult problem because it is usually inadequate to use odometry information alone to register scans taken at different locations (as accumulated odometry error grows without bound). On the other hand, we are unable to use pre-mapped external landmarks to correct pose error since the environment is unknown.

A generally employed approach to solving the registration problem is to align the overlapping parts of two adjacent scans to estimate their relative pose, and then use the relative pose to transform scans into the same coordinate system. But this brings up the consistency issue: If one scan can be registered relative to two (or more) previous scans, which registration should be trusted? Furthermore, if one particular registration (or an average of several registrations) is selected, how do we update the previously estimated poses accordingly?

Consider an example as shown in Fig. 6.1. The robot starts at A and returns to a nearby location G after visiting several other places. Now the scan at G can be registered either according to scan F or scan A . Here it will be better to match scan G against scan A as it gives smaller cumulative registration error. But once the pose G is derived from the relative pose GA , the pose

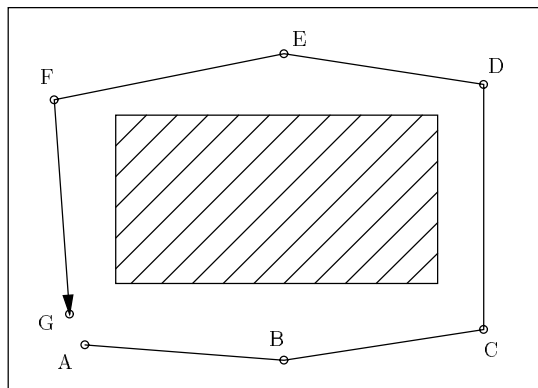


Figure 6.1: The robot starts at A and returns to a nearby location G after visiting other places. By matching the scan taken at G against the scan at A , the error at pose G can be reduced. But the estimates of F, D, \dots should also be updated to maintain consistency.

F should also be updated as otherwise the relative pose FG will be inconsistent with the result of matching scan F and G . The same argument applies to pose E, D, C, B . In general, the result of matching pairwise scans is a complex, and possibly conflicting, network of pose relations. We need a uniform framework to integrate all these relations and resolve the conflicts.

In this chapter, we present such a framework for consistently registering multiple range scans. The idea of our approach is to maintain all the local frames of data as well as a network of spatial relations among the local frames. These spatial relations (or pose constraints) are derived from matching pairs of scans or from odometry measurements. Then our goal is to estimate all the global pose variables from this network of constraints. Consistency among the local frames are ensured as all the spatial relations are taken into account simultaneously.

6.1.1 Pairwise Scan Matching

A procedure for aligning a pair of scans is needed to derive a constraint about their relative pose. We have already studied this problem in previous chapters. In this application, our algorithm takes two scans and a rough initial estimate of their relative pose (from odometry) as input. The output is a much improved estimate of the relative pose. After aligning the two scans, we record a set of corresponding points on the two scans (which is derived in the final iteration of the point correspondence based algorithm). This correspondence set will form a constraint between the two poses.

6.1.2 Constructing a Network of Pose Constraints

Given the pairwise scan matching algorithm, we can form a network of pose constraints by matching all possible scan pairs. In addition, we include odometry measurements in the network. Formally, the network of constraint is defined as a set of nodes and a set of links between pairs of nodes. A node of the network is a pose of the robot on its trajectory at which a range scan is taken. Here a pose is defined as a three dimensional vector $(x, y, \theta)^t$ consisting of a 2D robot position and the home orientation of the rotating range sensor. We then define two types of links between a pair of nodes. First, if two poses are adjacent along the robot path, we say that there is a *weak link* between the two nodes which is the odometry measurement of the relative pose. Second, if the range scans taken at two poses have a sufficient overlap, we say that there is a *strong link* between the two nodes. For each strong link, a constraint on the relative pose of the two nodes is determined by the set of corresponding points on the two scans given by the matching algorithm.

A network of constraints is constructed as the robot travels along a predefined path. First, the robot takes a scan S_0 at its initial pose V_0 . Then for each step of movement from pose V_{i-1} to pose V_i : (1) add a weak link to the network connecting V_{i-1} and V_i to store the odometry measurement of the relative pose; (2) take a scan S_i at the new pose V_i ; (3) match each previous scan $S_j, j = 0, \dots, i - 1$, with the current scan S_i and possibly obtain a set of corresponding pairs of points from S_i and S_j to create a strong link between V_i and V_j .

When we match a previous scan S_j to S_i , we first project S_j to the current pose V_i (by changing the coordinate system of the scan points) and discard the points which are likely not visible from V_i (see Section 3.3). The amount of overlap between S_i and S_j is estimated from the spatial extent of the matching pairs of points on the two scans. A fixed threshold is used to decide if the overlap is significant enough.

6.1.3 Combining Pose Constraints in a Network

The pose constraints in a network is potentially conflicting because the relations are derived independently with possible errors. Our task is to combine all the pose relations and resolve inconsistency among the relations. Smith and Cheeseman [107] addressed a similar problem of combining uncertain relations in a network. Their approach is to define a compounding operation and a merging operation to simplify serially or parallelly connected relations. However, their approach has a serious limitation as not every network can be reduced to a combination of serial or parallel connections; the complex networks can be simplified using their method.

We formulate this problem of combining pose constraints as one of optimally estimating the global poses in the network. We do not deal with the relations directly. Rather, we estimate the poses (which correspond to the nodes in the network) based on all the available constraints. The derived estimation of the nodes defines a consistent set of relations.

An optimization problem is formulated as the following. We define an objective function from the network with all the pose coordinates as variables (except V_0 which defines our reference coordinate system). Every link in the network is translated into a term in the objective function which can be conceived as a spring connecting two nodes. The spring achieves minimum energy when the relative pose between the two nodes equals to the measured value (either from matching two scans or from odometry). Then the objective function represents the total energy in the network. We finally solve for all the pose variables at once by minimizing this total energy function.

In the following sections, we go through the details of forming the energy terms corresponding to the two types of links. But first of all, we formulate the optimization criterion by considering a generic optimal estimation problem. We derive a closed-form solution for the optimal estimates in a linear special case. Later, the robot pose estimation problem will be reduced to the linear case and then the closed-form solution can be applied.

6.2 Optimal Estimation from a Network of Constraints

6.2.1 Problem Definition

We consider the following generic optimal estimation problem. Assume that we are given a network of uncertain measurements about $n + 1$ nodes X_0, X_1, \dots, X_n . Here each node X_i represents a d -dimensional position vector. A link D_{ij} between two nodes X_i and X_j represents a measurable difference of the two positions. Generally, D_{ij} is a (possibly nonlinear) function of X_i and X_j and we refer to this function as the measurement equation. Especially interesting to us is the simple linear case where the measurement equation is $D_{ij} = X_i - X_j$.

We model an observation of D_{ij} as $\bar{D}_{ij} = D_{ij} + \Delta D_{ij}$ where ΔD_{ij} is a random Gaussian error with zero mean and known covariance matrix C_{ij} . Given a set of measurements \bar{D}_{ij} between pairs of nodes and the covariance C_{ij} , our goal is to derive the optimal estimate of the position X_i 's by combining all the measurements. Moreover, we want to derive the covariance matrices of the estimated X_i 's based on the covariance matrices of the measurements.

Our criterion of optimal estimation is based on the *maximum likelihood* or *minimum variance*

concept. The node position X_i 's (and hence the position difference D_{ij} 's) are determined in such a way that the conditional joint probability of the derived D_{ij} 's, given their observations \bar{D}_{ij} 's, is maximized. If we assume that all the observation errors are Gaussian and mutually independent, the criterion is equivalent to minimizing the following Mahalanobis distance (where the summation is over all the given measurements):

$$W = \sum_{(i,j)} (D_{ij} - \bar{D}_{ij})^t C_{ij}^{-1} (D_{ij} - \bar{D}_{ij}). \quad (6.1)$$

Note that this optimal estimation problem is similar to the one addressed by the Kalman filter. However, we specifically deal with a network of measurements with a special type of measurement equation. In our problem, we keep all the poses as state variables, while a typical Kalman filter application only estimates the current pose and “forgets” about the history of poses.

A typical application of the optimal estimation problem is in mobile robot navigation, where we want to estimate the robot pose and its uncertainty in three degrees of freedom (x, y, θ) . As the robot moves from one place to another, its uncertainty about its pose with respect to its initial pose grows. On the other hand, the robot may use sensors (odometry, range sensor etc.) to make relative measurements between its current pose and other poses. We then want to utilize all the available measurements to derive the optimal estimate of the robot poses. Note that in this application, the measurement equation is non-linear because of the θ component in the robot pose.

Next, we study the case when the measurement equation is linear and we derive closed-form solutions for the optimal estimates of the nodes and their covariances. Later, we will solve the non-linear robot pose estimation problem by approximately forming linear measurement equations.

6.2.2 Solution of Optimal Linear Estimation

We consider the special case where the measurement equation is in the simple linear form: $D_{ij} = X_i - X_j$. Here $X_i, i = 0, 1, \dots, n$ are the nodes in the network which are d -dimensional vectors and the D_{ij} 's are the links of the network. Without loss of generality, we assume that there is a link D_{ij} between every pair of nodes X_i, X_j . For each D_{ij} , we have an observation \bar{D}_{ij} which is assumed to have Gaussian distribution with mean value D_{ij} and known covariance C_{ij} . In case the actual link D_{ij} is missing, we can simply let the corresponding C_{ij}^{-1} be 0. Then the criterion for the optimal estimation is to minimize the following Mahalanobis distance:

$$W = \sum_{0 \leq i < j \leq n} (X_i - X_j - \bar{D}_{ij})^t C_{ij}^{-1} (X_i - X_j - \bar{D}_{ij}). \quad (6.2)$$

Note that W is a function of all the position X_i 's. Since we can only solve for relative positions given the relative measurements, we choose one node X_0 as a reference and consider its coordinate as constant. Without loss of generality, we can let $X_0 = 0$ and then X_1, X_2, \dots, X_n will be the relative positions from X_0 .

We can express the measurement equations in a matrix form as

$$\mathbf{D} = \mathbf{H}\mathbf{X} \quad (6.3)$$

where \mathbf{X} is the nd -dimensional vector which is the concatenation of X_1, X_2, \dots, X_n ; \mathbf{D} is the concatenation of all the position differences of the form $D_{ij} = X_i - X_j$; and \mathbf{H} is the incidence matrix with all entries being 1, -1 , or 0. Then the function W can be represented in a matrix form as:

$$W = (\bar{\mathbf{D}} - \mathbf{H}\mathbf{X})^t \mathbf{C}^{-1} (\bar{\mathbf{D}} - \mathbf{H}\mathbf{X}) \quad (6.4)$$

where $\bar{\mathbf{D}}$ is the concatenation of all the observations \bar{D}_{ij} for the corresponding D_{ij} and \mathbf{C} is the covariance of $\bar{\mathbf{D}}$ which is a square matrix with C_{ij} 's in the diagonal and zeros elsewhere (assuming that the observation errors are independent).

Then the solution for \mathbf{X} which minimizes W is given by

$$\mathbf{X} = (\mathbf{H}^t \mathbf{C}^{-1} \mathbf{H})^{-1} \mathbf{H}^t \mathbf{C}^{-1} \bar{\mathbf{D}}. \quad (6.5)$$

The covariance of \mathbf{X} is

$$\mathbf{C}_{\mathbf{X}} = (\mathbf{H}^t \mathbf{C}^{-1} \mathbf{H})^{-1}. \quad (6.6)$$

Denote the $nd \times nd$ matrix $\mathbf{H}^t \mathbf{C}^{-1} \mathbf{H}$ by \mathbf{G} and expand the matrix multiplications. We can obtain the $d \times d$ submatrices of \mathbf{G} as

$$\begin{aligned} G_{ii} &= \sum_{j=0}^n C_{ij}^{-1} \\ G_{ij} &= -C_{ij}. \quad (i \neq j) \end{aligned} \quad (6.7)$$

Denote the nd -dimensional vector $\mathbf{H}^t \mathbf{C}^{-1} \bar{\mathbf{D}}$ by \mathbf{B} . Its d -dimensional subvectors are the following (let $\bar{D}_{ij} = -\bar{D}_{ji}$):

$$B_i = \sum_{j=0; j \neq i}^n C_{ij}^{-1} \bar{D}_{ij}. \quad (6.8)$$

Then the position estimates and covariance can be written as

$$\mathbf{X} = \mathbf{G}^{-1} \mathbf{B}; \quad \mathbf{C}_{\mathbf{X}} = \mathbf{G}^{-1}. \quad (6.9)$$

6.2.3 Special Networks



Figure 6.2: (a) Serial connection; (b) parallel connection.

We will apply the formula in Eq. 6.9 to two interesting simple cases as in Figure 6.2. First, if the network consists of two serially connected links, D_{01} and D_{12} , The derived estimate of X_2 and its covariance matrix are

$$X_2 = D_{01} + D_{12} \quad (6.10)$$

$$C = C_{01} + C_{12} \quad (6.11)$$

Another case to consider is the network which consists of two parallel links D' and D'' between two nodes X_0 and X_1 . If the covariance of the two links are C' and C'' , the estimate of X_1 and its covariance are given by

$$X_1 = (C'^{-1} + C''^{-1})^{-1}(C'^{-1}D' + C''^{-1}D'') \quad (6.12)$$

$$C = (C'^{-1} + C''^{-1})^{-1} \quad (6.13)$$

The solution is equivalent to the Kalman filter formulation. The above two cases correspond to the compounding and merging operations as described by Smith and Cheeseman [107].

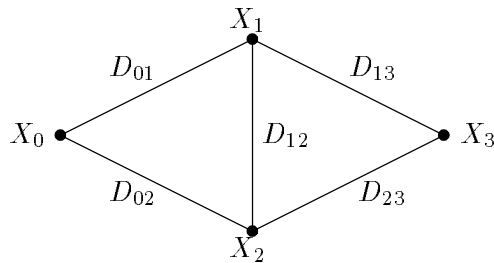


Figure 6.3: A Wheatstone bridge network.

In the next example, we consider a network in the form of a Wheatstone bridge (Fig. 6.3). Note that the estimate of X_3 can not be obtained through compounding and merging operations. There-

fore, the method by Smith and Cheeseman can not be directly applied to simplify this network¹. While in our method, the variables X_1, X_2, X_3 can be solved from the linear system $\mathbf{G}\mathbf{X} = \mathbf{B}$ where

$$\mathbf{G} = \begin{pmatrix} C_{01}^{-1} + C_{12}^{-1} + C_{13}^{-1} & -C_{12}^{-1} & -C_{13}^{-1} \\ -C_{12}^{-1} & C_{02}^{-1} + C_{12}^{-1} + C_{23}^{-1} & -C_{23}^{-1} \\ -C_{13}^{-1} & -C_{23}^{-1} & C_{13}^{-1} + C_{23}^{-1} \end{pmatrix} \quad (6.14)$$

$$\mathbf{B} = \begin{pmatrix} C_{01}^{-1}\bar{D}_{01} + C_{12}^{-1}\bar{D}_{12} + C_{13}^{-1}\bar{D}_{13} \\ C_{02}^{-1}\bar{D}_{02} - C_{12}^{-1}\bar{D}_{12} + C_{23}^{-1}\bar{D}_{23} \\ -C_{13}^{-1}\bar{D}_{13} - C_{23}^{-1}\bar{D}_{23} \end{pmatrix}. \quad (6.15)$$

The covariance matrix for the estimated position X_3 has a nice symmetric form (derived by expanding \mathbf{G}^{-1}):

$$C_3^{-1} = \begin{pmatrix} C_{01}^{-1} & C_{02}^{-1} \end{pmatrix} \begin{pmatrix} C_{01}^{-1} + C_{12}^{-1} + C_{13}^{-1} & -C_{12}^{-1} \\ -C_{12}^{-1} & C_{02}^{-1} + C_{12}^{-1} + C_{23}^{-1} \end{pmatrix}^{-1} \begin{pmatrix} C_{13}^{-1} \\ C_{23}^{-1} \end{pmatrix} \quad (6.16)$$

6.2.4 Analogy to Electrical Circuits

In this section, we make an interesting analogy between a network of measurements and an electrical circuit. We use this analogy to derive the covariance matrix for the estimate of a node state vector. Although the discussion in this section may give us some insight in analyzing a network, its result is already contained in our previous solution in Section 6.2.2.

Consider a network with nodes X_0, X_1, \dots, X_n . We have linear measurements \bar{D}_{ij} about $D_{ij} = X_i - X_j$ on all pairs of nodes X_i, X_j . By using all these measurements, optimal estimates of the state vectors X_1, \dots, X_n can be derived by minimizing the energy function W :

$$W = \sum_{0 \leq i < j \leq n} (X_i - X_j - \bar{D}_{ij})^t C_{ij}^{-1} (X_i - X_j - \bar{D}_{ij}). \quad (6.17)$$

Let these optimal estimates be $\bar{X}_1, \dots, \bar{X}_n$. Here we are interested in finding the covariance matrix of each \bar{X}_i . Without loss of generality, let us only consider \bar{X}_n .

In order to study the error distribution of \bar{X}_n , we introduce a small disturbance ΔX_n to X_n . Then we fix the state vector X_n with value $\bar{X}_n + \Delta X_n$ while minimizing W again with respect to X_1, \dots, X_{n-1} (call them intermediate nodes). After the minimization, there will be some error

¹It is possible to first convert a triangle in the network to an equivalent Y-shaped connection and then the network becomes one with serial and parallel links. However, this Delta-to-Y conversion still can not turn every network into a combination of serial and parallel connections.

ΔX_i at each intermediate node X_i added to its original value \bar{X}_i . By manipulating the partial derivatives of W , we can derive the relationship between the ΔX_i 's as the following:

$$\sum_{j=0, j \neq i}^n C_{ij}^{-1} (\Delta X_i - \Delta X_j) = 0, \quad i = 1, \dots, n-1. \quad (6.18)$$

We regard $f_{ij} = C_{ij}^{-1} (\Delta X_i - \Delta X_j)$ as some kind of flow (call it error flow) from X_i to X_j . Eq. 6.18 tells us that the total net outgoing (or incoming) error flow from/into an intermediate node is zero. Therefore, the total net flow passing through any cut which separates X_0 and X_n must be a constant (independent of the place of the cut). We denote this constant by F and define it as the total error flow from X_n to X_0 caused by disturbance ΔX_n .

We can now make the analogy between a measurement network and an electrical circuit. The error flow corresponds to electric current; the covariance matrix of a link is analogue to a resistor; the error ΔX_i at a node is the counterpart of voltage; and the introduced error ΔX_n resembles a power source. Also Eq. 6.18 has the same form as the Kirchhoff's Law of Current.

Our goal is to combine all the measurements in the network into a single link from X_0 to X_n so as to derive a total covariance C_n of \bar{X}_n . This is similar to computing the total resistance between two nodes in a circuit. Let us consider the reduced network which consists of a single link from X_0 to X_n . This network is equivalent to the original network (in the sense that the two \bar{X}_n 's have the same distribution) when the two networks have the same total energy as resulted from the same disturbance ΔX_n . It can be proved that this happens when the error flow in the reduced network equals to the total error flow in the original network. That is:

$$F = C_n^{-1} (\Delta X_n - \Delta X_0) = C_n^{-1} \Delta X_n. \quad (6.19)$$

Note that the above C_n is the total covariance matrix of \bar{X}_n . To determine C_n , we first compute the "voltage" ΔX_i at each node after adding the "power source" ΔX_n . From the equations in Eq. 6.18, we can solve ΔX_i in terms of ΔX_n . Let an $(n-1)d$ -dimensional vector $\mathbf{\Delta}$ be the concatenation of $\Delta X_1, \dots, \Delta X_{n-1}$. Then

$$\mathbf{\Delta} = \mathbf{G}_n^{-1} \mathbf{S} \Delta X_n, \quad (6.20)$$

where \mathbf{G}_n is a $(n-1)d \times (n-1)d$ matrix, \mathbf{S} is a $(n-1)d \times d$ matrix. Their $d \times d$ submatrices are:

$$G_{ii} = \sum_{j=0, j \neq i}^n C_{ij}^{-1} \quad (6.21)$$

$$G_{ij} = -C_{ij}^{-1} \quad i \neq j \quad (6.22)$$

$$S_i = C_{in}^{-1} \quad (6.23)$$

We can choose any cut of the network to compute the total error flow F . If we use the cut which separates X_0 from the rest of the network, we have

$$F = \sum_{i=1}^n C_{0i}^{-1} \Delta X_i = C_{0n}^{-1} \Delta X_n + \mathbf{T}^t \Delta, \quad (6.24)$$

where \mathbf{T} is a $(n-1)d \times d$ matrix whose $d \times d$ submatrices are

$$T_i = C_{0i}^{-1}. \quad (6.25)$$

Substitute Eq. 6.20 and 6.24 into Eq. 6.19, we have

$$\left(C_{0n}^{-1} + \mathbf{T}^t \mathbf{G}_n^{-1} \mathbf{S} \right) \Delta X_n = C_n^{-1} \Delta X_n. \quad (6.26)$$

Since ΔX_n is an arbitrary vector, C_n must satisfy:

$$C_n = \left(C_{0n}^{-1} + \mathbf{T}^t \mathbf{G}_n^{-1} \mathbf{S} \right)^{-1}. \quad (6.27)$$

Note that we previously derived the covariance matrix $\mathbf{C}_X = \mathbf{G}^{-1}$ for X (concatenation of all the state vectors). C_n is a $d \times d$ submatrix of \mathbf{C}_X and the same result of C_n can be obtained from expanding \mathbf{G}^{-1} using a formula of block matrix inverse.

6.3 Formulation of Constraints for Robot Pose Estimation

Now we come back to the robot pose estimation problem. We would like to formulate it into the generic form of maximum likelihood optimal estimation as in Section 6.2. We would also like to linearize the measurement equation so as to apply the closed-form linear solution to solve for the optimal robot poses.

In the following subsections, we study a constraint of pose difference given by matched scans or odometry measurements. For each constraint, we formulate a term in the form of Mahalanobis distance. For the convenience in discussions of pose measurements, we will first define a pose compounding operation.

6.3.1 Pose Compounding Operation

Assume that the robot starts at a pose $V_b = (x_b, y_b, \theta_b)^t$ and it then changes its pose by $D = (x, y, \theta)^t$ relative to V_b , ending up at a new pose $V_a = (x_a, y_a, \theta_a)^t$. Then we say that pose V_a is the compounding of V_b and D . We denote it as:

$$V_a = V_b \oplus D. \quad (6.28)$$

The coordinates of the poses are related by:

$$x_a = x_b + x \cos \theta_b - y \sin \theta_b \quad (6.29)$$

$$y_a = y_b + x \sin \theta_b + y \cos \theta_b \quad (6.30)$$

$$\theta_a = \theta_b + \theta. \quad (6.31)$$

If we regard that an absolute pose defines a coordinate system (the xy coordinates of the origin and the direction of one axis), and a relative pose defines a change of coordinate system (a translation followed by a rotation), then the compounding operation gives the pose which defines the new coordinate system after the transformation. The compounding operation is not commutative, but it is associative. We can thus define the compounding of a series of poses.

It is also useful to define the inverse of compounding which takes two poses and gives the relative pose:

$$D = V_a \ominus V_b. \quad (6.32)$$

The coordinates are related by the following equations:

$$x = (x_a - x_b) \cos \theta_b + (y_a - y_b) \sin \theta_b \quad (6.33)$$

$$y = -(x_a - x_b) \sin \theta_b + (y_a - y_b) \cos \theta_b \quad (6.34)$$

$$\theta = \theta_a - \theta_b. \quad (6.35)$$

If D_{ab} is the relative pose $V_a \ominus V_b$, the reversed relative pose $D_{ba} = V_b \ominus V_a$ can be obtained from D_{ab} by a unary operation:

$$D_{ba} = \ominus D_{ab} = (0, 0, 0)^t \ominus D_{ab}. \quad (6.36)$$

We can verify that $(\ominus D) \oplus V = V \ominus D$.

We also want to define a compounding operation between a full 3D pose $V_b = (x_b, y_b, \theta_b)$ and a 2D position vector $u = (x, y)^t$. The result is another 2D vector $u' = (x', y')^t$. We denote the operation as

$$u' = V_b \oplus_2 u. \quad (6.37)$$

The coordinates for u' are given by the first two equations of the full 3D pose compounding (Eq. 6.29,6.30). This 2D compounding operation is useful for transforming an non-oriented point (typically from a range sensor) from its local sensor coordinate system to the global coordinate system.

6.3.2 Pose Constraint from Matched Scans

Let V_a and V_b be two nodes in the network and assume there is a strong link connecting the two poses. From the pairwise scan matching algorithm, we get a set of pairs of corresponding points: $u_k^a, u_k^b, k = 1, \dots, m$, where the 2D non-oriented points u_k^a, u_k^b are from scan S_a and S_b , respectively. Each pair (u_k^a, u_k^b) corresponds to the same physical point in the robot's environment while they are represented in different local coordinate systems. If we ignore any sensing or matching errors, a pair of corresponding points are related by:

$$\Delta Z_k = V_a \oplus_2 u_k^a - V_b \oplus_2 u_k^b = 0. \quad (6.38)$$

Then if we take the random observation errors into account, we can regard ΔZ_k as a random variable with zero mean and some unknown covariance C_k^Z . From the correspondence pairs, we can form a constraint on the pose difference by minimizing the following distance function:

$$F_{ab}(V_a, V_b) = \sum_{k=1}^m \|(V_a \oplus_2 u_k^a) - (V_b \oplus_2 u_k^b)\|^2. \quad (6.39)$$

If we notice that a pose change is a rigid transformation under which the squared Euclidean distance $\|\cdot\|^2$ is invariant, we can rewrite the function in an equivalent form:

$$F_{ab}(V_a, V_b) = \sum_{k=1}^m \|((V_a \ominus V_b) \oplus_2 u_k^a) - u_k^b\|^2. \quad (6.40)$$

Thus F_{ab} is a function of $D' = V_a \ominus V_b$. The solution of D' which minimizes F_{ab} can be derived in closed-form (see Section 4.1). The relation $D' = V_a \ominus V_b$ is the measurement equation.

In order to reduce F_{ab} into the Mahalanobis distance form, we linearize each term ΔZ_k . Let $\bar{V}_a = (\bar{x}_a, \bar{y}_a, \bar{\theta}_a)^t$, $\bar{V}_b = (\bar{x}_b, \bar{y}_b, \bar{\theta}_b)^t$ be some close estimates of V_a and V_b . Denote $\Delta V_a = \bar{V}_a - V_a$ and $\Delta V_b = \bar{V}_b - V_b$. Let $u_k = (x_k, y_k)^t = V_a \oplus_2 u_k^a \approx V_b \oplus_2 u_k^b$ (the global coordinates of a pair of matching points). Then for small ΔV_a and ΔV_b , we can derive from Taylor expansion:

$$\begin{aligned} \Delta Z_k &= V_a \oplus_2 u_k^a - V_b \oplus_2 u_k^b = (\bar{V}_a - \Delta V_a) \oplus_2 u_k^a - (\bar{V}_b - \Delta V_b) \oplus_2 u_k^b \\ &\approx (\bar{V}_a \oplus_2 u_k^a - \bar{V}_b \oplus_2 u_k^b) - \left(\begin{pmatrix} 1 & 0 & \bar{y}_a - y_k \\ 0 & 1 & -\bar{x}_a + x_k \end{pmatrix} \Delta V_a - \begin{pmatrix} 1 & 0 & \bar{y}_b - y_k \\ 0 & 1 & -\bar{x}_b + x_k \end{pmatrix} \Delta V_b \right) \\ &= (\bar{V}_a \oplus_2 u_k^a - \bar{V}_b \oplus_2 u_k^b) - \begin{pmatrix} 1 & 0 & -y_k \\ 0 & 1 & x_k \end{pmatrix} (\bar{H}_a \Delta V_a - \bar{H}_b \Delta V_b) \end{aligned} \quad (6.41)$$

where

$$\bar{H}_a = \begin{pmatrix} 1 & 0 & \bar{y}_a \\ 0 & 1 & -\bar{x}_a \\ 0 & 0 & 1 \end{pmatrix}, \quad \bar{H}_b = \begin{pmatrix} 1 & 0 & \bar{y}_b \\ 0 & 1 & -\bar{x}_b \\ 0 & 0 & 1 \end{pmatrix}. \quad (6.42)$$

We can rewrite Eq. 6.41 as

$$\Delta Z_k \approx \bar{Z}_k - M_k D \quad (6.43)$$

where

$$\bar{Z}_k = \bar{V}_a \oplus_2 u_k^a - \bar{V}_b \oplus_2 u_k^b \quad (6.44)$$

$$M_k = \begin{pmatrix} 1 & 0 & -y_k \\ 0 & 1 & x_k \end{pmatrix} \quad (6.45)$$

$$D = (\bar{H}_a \Delta V_a - \bar{H}_b \Delta V_b). \quad (6.46)$$

Thus we can now regard D in Eq. 6.46 as the pose difference measurement equation to replace $D' = V_a \ominus V_b$. For the m correspondence pairs, we can form m equations as in Eq. 6.43. If we concatenate the \bar{Z}_k 's to form a $2m \times 1$ vector \mathbf{Z} , and stack the M_k 's to form a $2m \times 3$ matrix \mathbf{M} , then F_{ab} can be rewritten as a quadratic function of D :

$$F_{ab}(D) = \sum_{k=1}^m (\Delta Z_k)^t (\Delta Z_k) \quad (6.47)$$

$$\approx (\mathbf{Z} - \mathbf{M}D)^t (\mathbf{Z} - \mathbf{M}D). \quad (6.48)$$

We can then solve for the $D = \bar{D}$ which minimizes F_{ab} as

$$\bar{D} = (\mathbf{M}^t \mathbf{M})^{-1} \mathbf{M}^t \mathbf{Z}. \quad (6.49)$$

The criterion of minimizing $F_{ab}(D)$ constitutes a least-squares linear regression. In Eq. 6.43, M_k is known and \bar{Z}_k is observed with an error ΔZ_k having zero mean and unknown covariance C_k^Z . If we assume that all the errors are independent variables having the same Gaussian distribution, and further assume that the error covariance matrices have the form:

$$C_k^Z = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}, \quad (6.50)$$

then the least squares solution \bar{D} has the Gaussian distribution whose mean value is the true underlying value and whose estimated covariance matrix is given by $C_D = s^2 (\mathbf{M}^t \mathbf{M})^{-1}$, where s^2 is the unbiased estimate of σ^2 :

$$s^2 = (\mathbf{Z} - \mathbf{M}\bar{D})^t (\mathbf{Z} - \mathbf{M}\bar{D}) / (2m - 3) = \frac{F_{ab}(\bar{D})}{2m - 3}. \quad (6.51)$$

Moreover, we notice that Eq. 6.48 can be rewritten as

$$F_{ab}(D) \approx (\bar{D} - D)^t (\mathbf{M}^t \mathbf{M}) (\bar{D} - D) + F_{ab}(\bar{D}). \quad (6.52)$$

Then we can define the energy term W_{ab} corresponding to the pose constraint and it is equivalent to a Mahalanobis distance:

$$W_{ab} = (F_{ab}(D) - F_{ab}(\bar{D}))/s^2 \quad (6.53)$$

$$\approx (\bar{D} - D)^t C_D^{-1} (\bar{D} - D) \quad (6.54)$$

where

$$C_D = s^2 (\mathbf{M}^t \mathbf{M})^{-1} \quad (6.55)$$

is the estimated covariance of \bar{D} . Note that D (as given in Eq. 6.46) is the linearized pose difference measurement equation.

6.3.3 Pose Constraints from Odometry

We also form an energy term in the objective function for each weak link. Suppose odometry gives a measurement \bar{D}' of the relative pose D' as the robot travels from pose V_b to pose V_a . The measurement equation is:

$$D' = V_a \ominus V_b. \quad (6.56)$$

We define the energy term in the objective function as the following:

$$W_{ab} = (\bar{D}' - D')^t C'^{-1} (\bar{D}' - D') \quad (6.57)$$

where C' is the covariance of the odometry error in the measurement \bar{D}' .

The covariance of measurement error is estimated as the following. Consider that a cycle of pose change consists of: (1) the robot platform rotation by an angle α to face towards the new target position; (2) the robot translation by a distance L to arrive at the new position; (3) the sensor rotation by a total cumulative angle β (usually 360°) to take a scan of measurements while the platform is stationary. We model the deviations σ_α , σ_L , σ_β , of the errors in the variables α , L , and β as proportional to their corresponding values, while the constant ratios are determined empirically. The 3D pose change $D' = (x, y, \theta)^t$ is derived as:

$$x = L \cos \alpha; \quad y = L \sin \alpha; \quad \theta = \alpha + \beta. \quad (6.58)$$

Then the covariance C' of the pose change D' can be approximated as:

$$C' = J \begin{pmatrix} \sigma_\alpha^2 & 0 & 0 \\ 0 & \sigma_L^2 & 0 \\ 0 & 0 & \sigma_\beta^2 \end{pmatrix} J^t \quad (6.59)$$

where J is the Jacobian matrix consisting of the partial derivatives of $(x, y, \theta)^t$ with respect to $(\alpha, L, \beta)^t$:

$$J = \begin{pmatrix} -L \sin \alpha & \cos \alpha & 0 \\ L \cos \alpha & \sin \alpha & 0 \\ 1 & 0 & 1 \end{pmatrix}. \quad (6.60)$$

We would also like to linearize and transform the measurement equation of D' to make the pose difference representation for odometry measurements consistent with that for matched sensing data. Consider the observation error $\Delta D' = \bar{D}' - D'$ of odometry. Let $\bar{V}_a = (\bar{x}_a, \bar{y}_a, \bar{\theta}_a)^t$, $\bar{V}_b = (\bar{x}_b, \bar{y}_b, \bar{\theta}_b)^t$ be some close estimates of V_a and V_b . Denote $\Delta V_a = \bar{V}_a - V_a$ and $\Delta V_b = \bar{V}_b - V_b$. Then through Taylor expansion, the observation error $\Delta D'$ becomes:

$$\Delta D' = \bar{D}' - D' = \bar{D}' - (V_a \ominus V_b) \quad (6.61)$$

$$= \bar{D}' - ((\bar{V}_a - \Delta V_a) \ominus (\bar{V}_b - \Delta V_b)) \quad (6.62)$$

$$\approx \bar{D}' - (\bar{V}_a \ominus \bar{V}_b) + \bar{K}_b^{-1}(\Delta V_a - \bar{H}_{ab}\Delta V_b) \quad (6.63)$$

where

$$\bar{K}_b^{-1} = \begin{pmatrix} \cos \bar{\theta}_b & \sin \bar{\theta}_b & 0 \\ -\sin \bar{\theta}_b & \cos \bar{\theta}_b & 0 \\ 0 & 0 & 1 \end{pmatrix}; \quad \bar{H}_{ab} = \begin{pmatrix} 1 & 0 & -\bar{y}_a + \bar{y}_b \\ 0 & 1 & \bar{x}_a - \bar{x}_b \\ 0 & 0 & 1 \end{pmatrix}. \quad (6.64)$$

Notice that $\bar{H}_{ab} = \bar{H}_a^{-1}\bar{H}_b$ where \bar{H}_a and \bar{H}_b are defined in Eq. 6.42. If we define a new observation error $\Delta D = -\bar{H}_a\bar{K}_b\Delta D'$, then we can rewrite Eq. 6.63 as

$$\Delta D = \bar{D} - (\bar{H}_a\Delta V_a - \bar{H}_b\Delta V_b) = \bar{D} - D \quad (6.65)$$

where we denote

$$\bar{D} = \bar{H}_a\bar{K}_b((\bar{V}_a \ominus \bar{V}_b) - \bar{D}') \quad (6.66)$$

$$D = \bar{H}_a\Delta V_a - \bar{H}_b\Delta V_b. \quad (6.67)$$

Notice that now we are dealing with the measurement equation for D which is consistent with that for matched sensing data. \bar{D} can be considered as an observation of D . The covariance C of \bar{D} can be computed from the covariance C' of \bar{D}' as:

$$C = \bar{H}_a\bar{K}_bC'\bar{K}_b^t\bar{H}_a^t. \quad (6.68)$$

The energy term in the objective function now becomes:

$$W_{ab} \approx (\bar{D} - D)^t C^{-1} (\bar{D} - D). \quad (6.69)$$

6.4 Optimal Pose Estimation

Once we have uniformly formulated the two types of measurements, we can form the total energy function for all the measurements:

$$W = \sum_{(i,j)} (\bar{D}_{ij} - D_{ij})^t C_{ij}^{-1} (\bar{D}_{ij} - D_{ij}) \quad (6.70)$$

where D_{ij} is the linearized pose difference between V_j and V_i :

$$D_{ij} = \bar{H}_i \Delta V_i - \bar{H}_j \Delta V_j \quad (6.71)$$

and \bar{D}_{ij} is an observation of D_{ij} (\bar{D}_{ij} is derived from the true observations, either range data or odometry measurements). The covariance C_{ij} is also known.

By regarding $X_i = \bar{H}_i \Delta V_i$ as the state vector corresponding to a node of the network as in Section 6.2.2, we can directly apply the closed-form linear solution to solve for the X_i 's as well as their covariance C_i^X . The formulas are in Eq. 6.5 to Eq. 6.9. Then the pose V_i and its covariance C_i can be updated as:

$$V_i = \bar{V}_i - \bar{H}_i^{-1} X_i, \quad C_i = (\bar{H}_i^{-1}) C_i^X (\bar{H}_i^{-1})^t. \quad (6.72)$$

Note that the pose estimate V_i and the covariance C_i is given based on the assumption that the reference pose $V_0 = 0$. If, in fact, $V_0 = (x_0, y_0, \theta_0)^t$ is non-zero, the solution should be transformed to

$$V_i' = V_0 \oplus V_i; \quad C_i' = K_0 C_i K_0^t \quad (6.73)$$

where

$$K_0 = \begin{pmatrix} \cos \theta_0 & -\sin \theta_0 & 0 \\ \sin \theta_0 & \cos \theta_0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (6.74)$$

Also note that when we derive the solution of linear estimation, we have an assumption that the errors on different links are all independent. Here we need to assume that the errors from the scan matching procedure or odometry measurements are independent.

6.5 Algorithm Implementation

6.5.1 Estimation Procedure

The implementation of the estimation algorithm is straightforward. After building the network, we obtain the initial pose estimates $\bar{V}_1, \dots, \bar{V}_n$ by compounding the odometry measurements. Then

for each link, we compute a measurement vector \bar{D}_{ij} and a covariance matrix C_{ij} according to Eq. 6.49, 6.55 or Eq. 6.66, 6.68. Finally, we form a large linear system $\mathbf{GX} = \mathbf{B}$ as explained in Section 6.2.2 and solve for the pose variables \mathbf{X} .

The components needed to build \mathbf{G} and \mathbf{B} are C_{ij}^{-1} and $C_{ij}^{-1}\bar{D}_{ij}$. In the case of a strong link (from matching a pair of scans), these components can be readily computed as $C_{ij}^{-1} = (\mathbf{M}^t\mathbf{M})/s^2$; $C_{ij}^{-1}\bar{D}_{ij} = (\mathbf{M}^t\mathbf{Z})/s^2$ which can be expanded into simple summations by noting the regularity in the matrix \mathbf{M} . In the case of a weak link (from odometry), these components can be computed by small matrix multiplications. The most expensive operation in the estimation process is to compute the inverse of a $3n \times 3n$ matrix \mathbf{G} which gives the covariance of \mathbf{X} .

The network is stored as a list of links and a list of nodes. Each link contains the following information: type of link, labels of the two nodes, the computed measurement (relative pose), and the covariance matrix of the measurement. Each node contains a range scan.

6.5.2 Iterative Process

We made linear approximations in the measurement equations in formulating the optimization criterion. The first order approximation error is proportional to the error in the initial pose estimate. Clearly, if we employ the newly derived pose estimate to formulate the linear algorithm again, a even more accurate pose estimate can be obtained. In fact, we can iterate the entire process to remove the approximation errors.

This strategy turns out to be very successful. We learned from experiments that the iterative process converges very fast. Typically, the first iteration corrects well over 90% of the total pose error correctable by iterating the process. It usually takes four or five iterations to converge to the limit of machine accuracy.

6.6 Discussion

We have formulated the problem of consistent range data registration as one of optimal estimation from a network of constraints. The main idea is to treat the history of robot poses as state variables and consider sensor measurements as observations about these variables. By solving the system using all the constraints at the same time, global consistency among the poses is enforced. It appears that this is a much more convenient way of maintaining consistency than Durrant-Whyte's approach of explicitly updating the estimates along each loop of the network [39].

Smith and Cheeseman also formulated spatial uncertainty in three degrees of freedom (x, y, θ) using mean vectors and covariance matrices [107]. They proposed compounding and merging operations for combining uncertainties in a network of measurements. However, their method has a limitation that only a certain type of network can be handled (i.e. the ones which are reducible to a combination of serial and parallel connections). Our method has a significant advantage over their work that we can estimate uncertain spatial relations from an arbitrary network.

Another contribution of our work is that we studied two types of sensor measurements: range scans and odometry readings, and uniformly represented pose uncertainties from each type of measurement using covariance matrices.

Chapter 7

Experiments and Alternative Approaches

7.1 Introduction

In the previous chapter we proposed a procedure of optimal estimation of global robot poses from a network of measurements, for consistent alignment of multiple range scans. We now present experiments applying this algorithm in registering simulated and real range scan data. We first show experiments using simulated environments and measurements. Then an example using real data is presented. Later in the chapter, we discuss an alternative formulation of the algorithm that does not require a linear approximation step. Finally, we discuss how this method can be adapted as a sequential estimation procedure to continuously register new sensor data.

7.2 Experiments with Simulated Measurements

In the first example, we simulate a rectangular environment with a width of 10 units. The robot travels around a central object and forms a loop in the path. There are 13 poses along the path at which simulated range scans are generated (with random measurement errors). We also simulate a random odometry error (which is the difference between a pose change the robot thinks it made and the actual pose change) at each leg of the trajectory. The magnitude of the accumulated odometry error is typically in the range of 0.5 units.

We apply our global pose estimation algorithm to correct the pose errors. In Fig. 7.2(a), we show all the scans recorded in the initial coordinate system where the pose of each scan is obtained by compounding odometry measurements. Due to the accumulation of odometry error the scan data

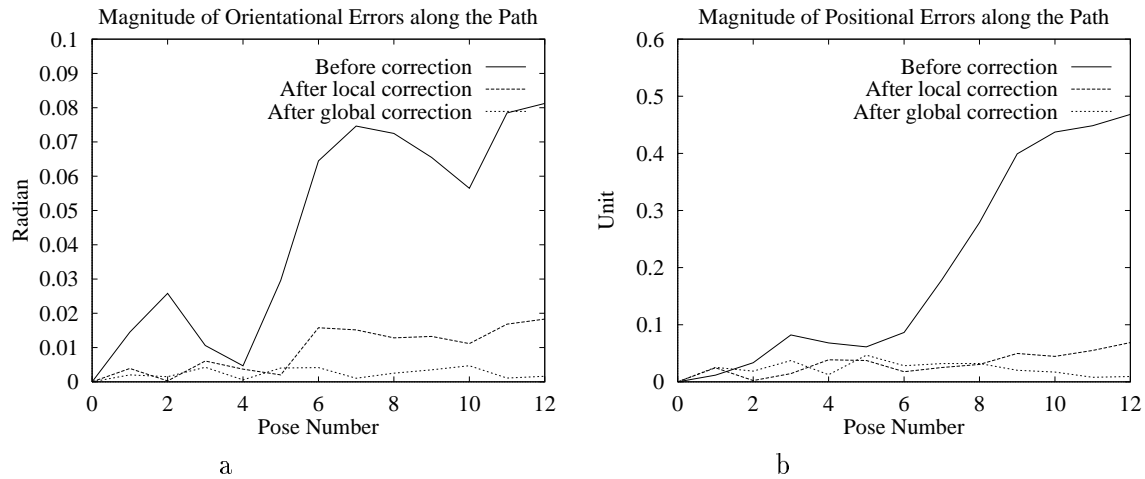


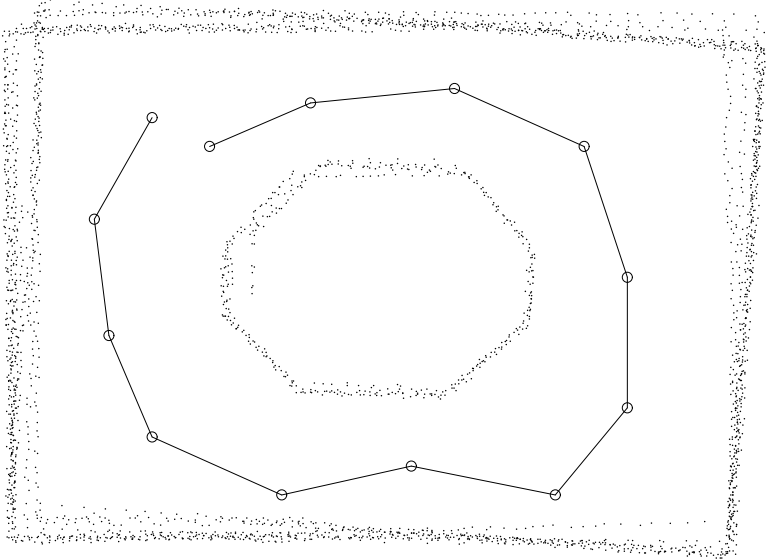
Figure 7.1: Pose errors along the path, before correction, after local correction, and after global correction. (a) Orientational errors; (b) positional errors.

are aligned poorly. In Fig. 7.2(b), we show the result of correcting the pose errors and realigning the scan data. Each line segment (either dashed or solid) in the figure represents a strong link obtained from matching two scans. In addition, the solid lines show the robot path which corresponds to the weak links. A plot of orientational and positional errors of the poses along the path, both before and after the global correction, is given in Fig. 7.1. Pose errors are accumulated along the path while the corrected pose errors are bounded. For comparison, we also apply a local registration procedure which matches one scan only to the previous scan. The pose errors along the path after this local correction is also shown in Fig. 7.1. Although pose errors are also significantly reduced after local corrections, they can still potentially grow without bound. In this example, global registration produces more accurate results than local correction.

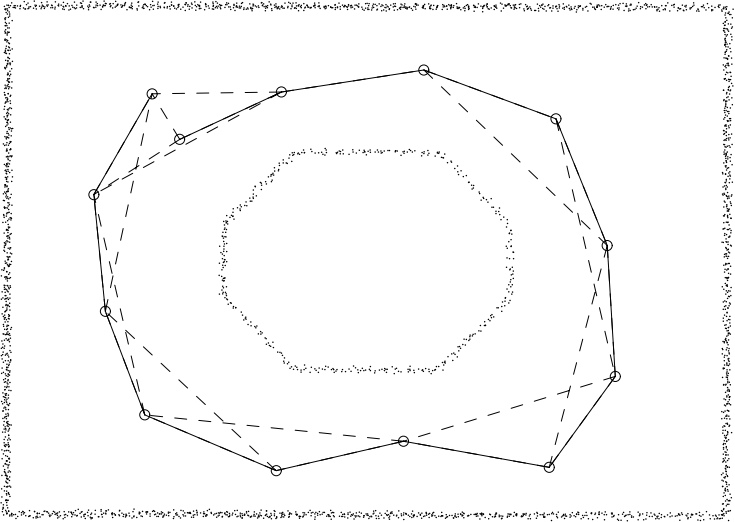
In the second example, the environment is a rectangle containing two blocks in the center. The robot goes around the blocks and forms a couple of loops in its path. There are 29 poses along the path. Between pairwise poses, 70 pose constraints are obtained from matching the scans. Fig. 7.3 shows the simulated range scans before registration and after registration. The pose constraints and the robot path are also shown.

The environment in the third example contains three objects of different shapes. Range scans are simulated at the 16 poses along the path. Fig. 7.4 shows the scans before and after global registration using the pose estimation algorithm.

Our simulated examples demonstrate that the pose estimation algorithm is effective in consis-



a



b

Figure 7.2: Global registration of multiple scans using simulated scan data. (a) scans recorded in a global coordinate system where the pose of each scan is obtained from compounding odometry measurements. The scans align poorly because of accumulation of odometry error. (b) the result of correcting pose errors. Both the dashed lines and solid lines show the constraints from matching scan pairs. The solid lines also give the robot path and odometry constraints.

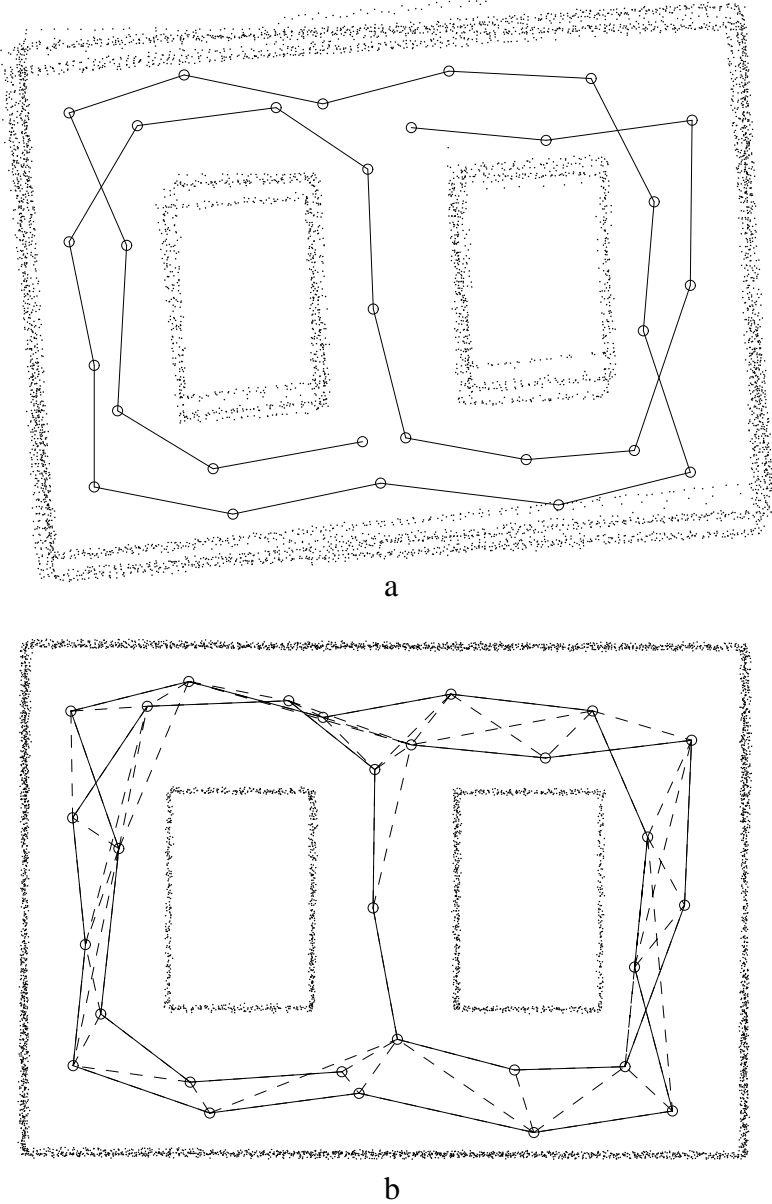
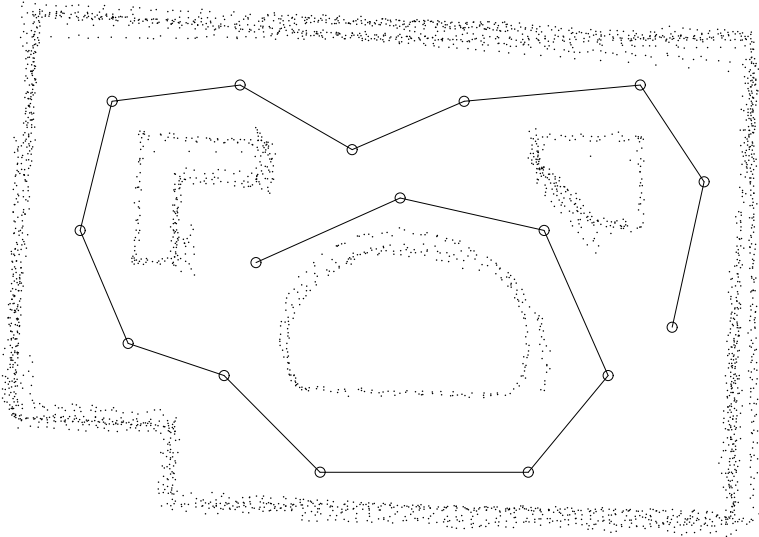
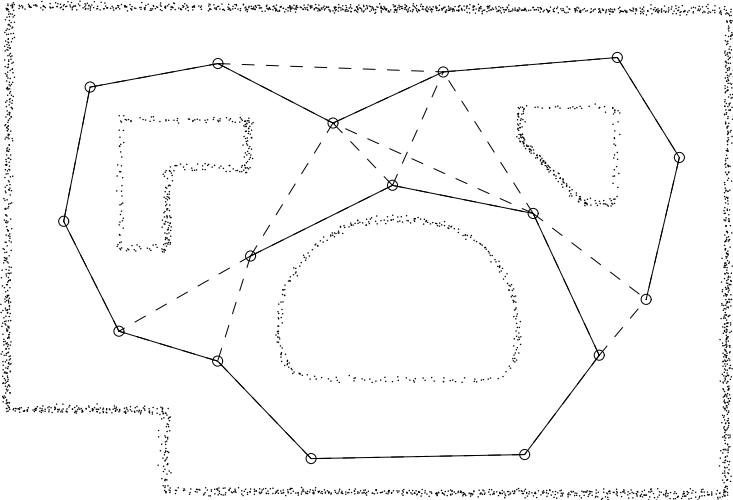


Figure 7.3: Global registration of multiple scans using simulated scan data. (a) scans before registration. (b) scans after registration. Both the dashed lines and solid lines show the constraints from matching scan pairs. The solid lines also give the robot path. There are 29 poses and 70 constraint links.



a



b

Figure 7.4: Another example of global registration of multiple scans using simulated scan data. (a) scans before registration. (b) scans after registration.

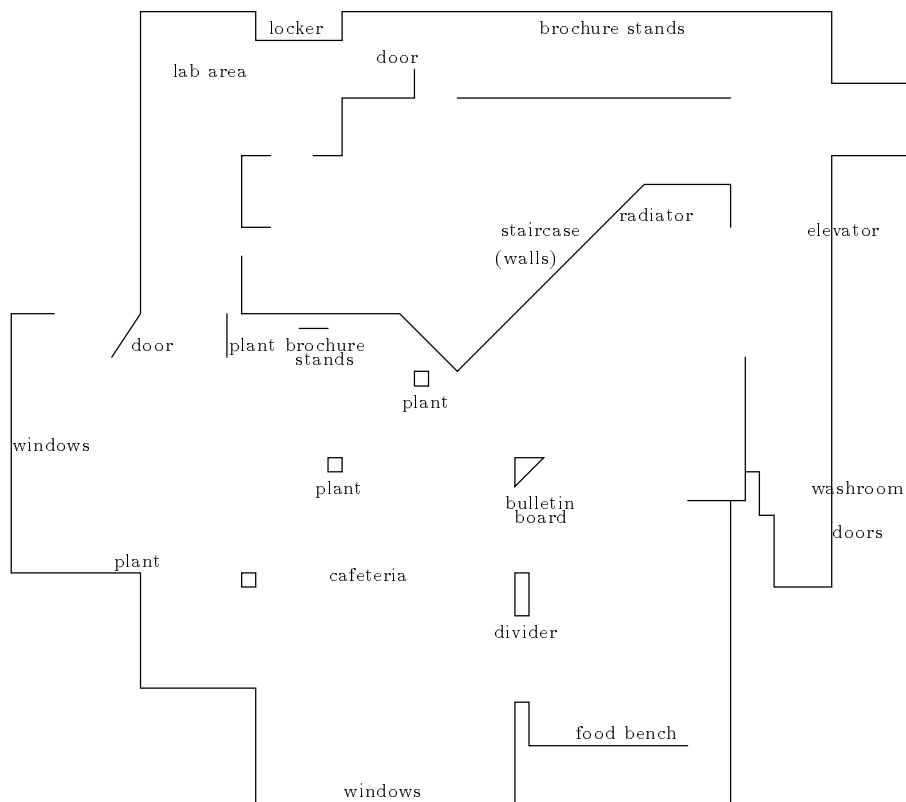


Figure 7.5: Testing environment in the cafeteria at FAW, Ulm, Germany.

tently aligning range scans. Another notable thing is that our pairwise scan matching procedure, which has been called numerous times in constructing the network of constraints, appears to be very effective and reliable.

7.3 Experiments with Real Data

We present another experiment with real range scans and odometry data. The testing environment is the cafeteria and nearby corridor in FAW, Ulm, Germany. An illustration of the environment is given in Fig. 7.5. The robot travels through the environment according to a given path. A sequence of 30 scans which were taken by the robot with an interval of about 2 meters between scan poses were obtained.¹ The laser sensor is a Ladar 2D IBEO Lasertechnik which is mounted on the AMOS robot. This laser sensor has a maximum viewing angle of 220 degrees. Thus having only the 2D

¹The author would like to thank Steffen Gutmann, Joerg Illmann, Thomas Kaempke, Manfred Knick, Erwin Prassler, and Christian Schlegel from FAW, Ulm for collecting the range scans for our experiments.

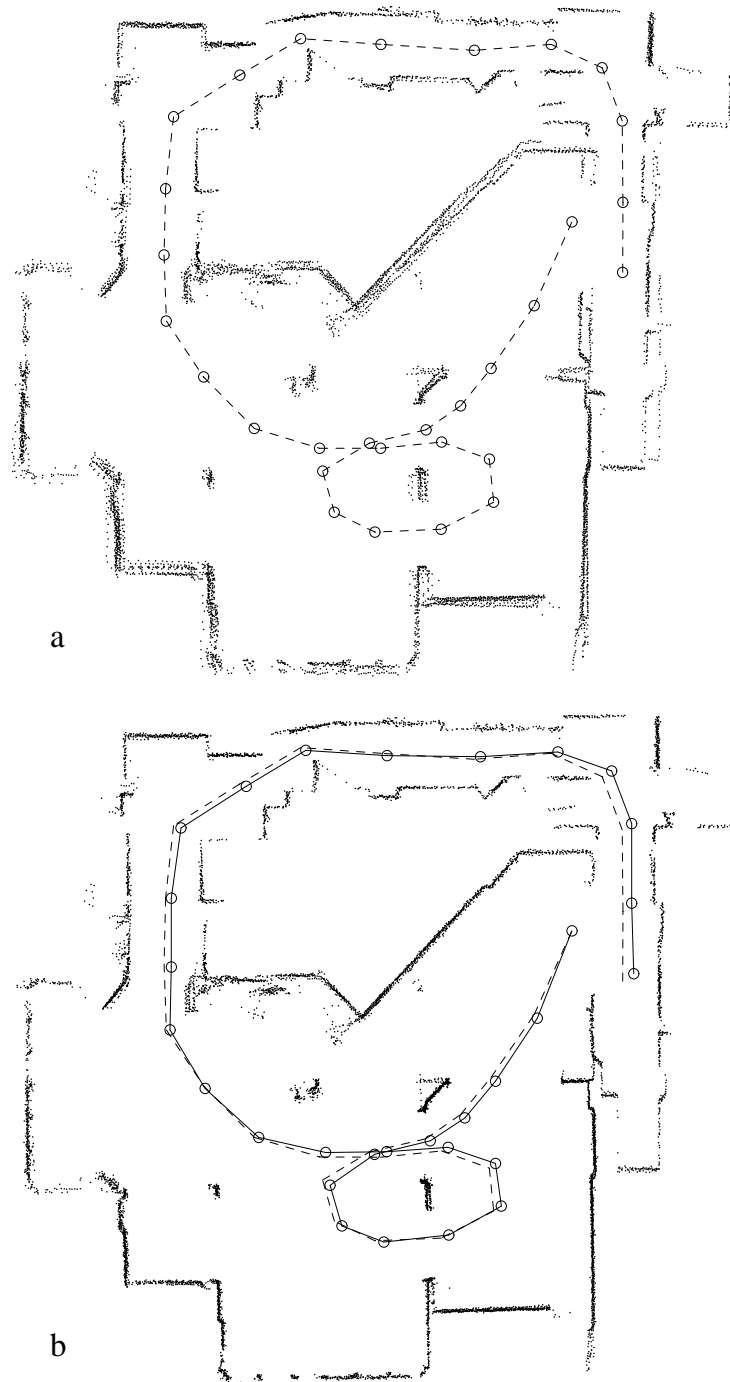


Figure 7.6: Consistent global registration of 30 real range scans which are collected by a robot at FAW, Ulm, Germany. (a) unregistered scans whose poses are subject to large odometry errors. (b) registered scans after correcting the pose errors. The robot path estimated from odometry is shown in dashed lines. The corrected path is shown in solid lines.

positions of two poses close together does not necessarily ensure a sufficient overlap between the scans taken at the two poses; we also need the sensor heading directions to be similar. Among the 30 scans, 84 links from matching overlapping scan pairs are constructed. Our scan matching algorithm is successful in finding the relative poses in all 84 calls. Some of these pairwise scan matching results have been shown in Chapter 5. After constructing the network, the estimation algorithm successfully derived the global poses of the scans. In Fig. 7.6, we show the unregistered scans (where the odometry errors were exaggerated) in part (a) and the globally registered scans in part (b). It is apparent that scans align much better after the registration.

7.4 An Alternative Solution without Linear Approximation

Note that linear approximations for the measurement equations are of theoretical interests as they help to formulate the maximum likelihood criterion and to derive the covariance of the estimations. But to just solve for the optimal estimates, the linear approximation step is unnecessary. Here we propose an alternative method to solve the same pose estimation problem by directly minimizing a non-quadratic objective function without explicitly linearizing the measurement equations.

The objective function W is a sum of the terms corresponding to the links of the network. The term for a strong link is the following:

$$W_{ab} = (F_{ab}(D') - F_{ab}(\bar{D}'))/s^2 \quad (7.1)$$

where $F_{ab}(D')$ is defined in Eq. 6.40; D' is the original pose difference $D' = V_a \ominus V_b$; $F_{ab}(\bar{D}')$ is the minimum residual of F_{ab} (can be omitted without affecting optimization result). The term for a weak link is:

$$W_{ab} = (D' - \bar{D}')^t C'^{-1} (D' - \bar{D}') \quad (7.2)$$

where \bar{D}' is the direct pose difference measured by odometry and C' is the covariance matrix given in Eq. 6.59. The objective function is the same as the one in the Mahalanobis distance form except we undo the linear approximations to the measurement equations.

We minimize W using a non-linear optimization procedure which implements Newton's method coupled with a line search to ensure convergence. We provide the first and second derivatives of W in closed-form. Since the objective function is a well-conditioned smooth function (which is near quadratic), the optimization process converges very easily. Typically, it takes about five to six iterations to converge to the limit of machine accuracy.

Another interesting result is that the covariance matrix of the pose variables can be obtained from the Hessian matrix (second derivative of W) evaluated at the minimum point of W . Let \mathbf{V} be the $3n \times 1$ vector containing the solutions of all the variables in V_1, V_2, \dots, V_n . The covariance matrix \mathbf{C}_V is:

$$\mathbf{C}_V = 2 \left(\frac{\partial^2 W}{\partial \mathbf{V}^2} \right)^{-1}. \quad (7.3)$$

This can be quickly derived by observing the objective function in matrix form (Eq. 6.4) and the closed-form covariance matrix (Eq. 6.6).

Comparing our two optimization methods, we notice that the “linear measurement method” converges slightly faster than the alternative optimization method (with Newton’s iterations). Especially in the first one or two iterations, the solution of the “linear measurement method” is considerably more accurate than the corresponding solution from the Newton’s iterations. But if we allow both methods to run until convergence, the two solutions are essentially identical.

7.5 Sequential Estimation

The estimation algorithm we previously discussed is a one-step procedure which solves all the pose variables at the same time. The algorithm is to be applied only after collecting all the measurements. Yet it will be more practically useful if we have a sequential algorithm which continuously provides estimates about the current or past pose variables after getting each new measurement. Here we will derive such a sequential procedure.

Our sequential algorithm maintains the current best estimate about the poses of previously visited places. At each step, a new location is visited and measurements about the new location as well as the previous locations are gathered. By using these new measurements, the current pose can be estimated while the previous poses can be updated.

Let X_1, \dots, X_{n-1} be the pose vectors which we have previously estimated and let X_n be the current new pose which we are about to measure. Let \mathbf{X} represent the concatenation of X_1, \dots, X_{n-1}, X_n . Assume that we currently have an estimate \mathbf{X}_0 of \mathbf{X} whose inverse covariance matrix is $\mathbf{C}_{X_0}^{-1}$. Because we have no knowledge about X_n yet, the X_n component in \mathbf{X}_0 contains arbitrary value and the matrix $\mathbf{C}_{X_0}^{-1}$ has all zeros in the last d rows and d columns (here $d = 3$). Now consider the addition of a set of new measurements relating X_n to some of the past pose variables. Let the measurement equation, in matrix form, be $\mathbf{D} = \mathbf{H}\mathbf{X}$ (\mathbf{H} is a constant matrix). Assume that the set of measurements is $\bar{\mathbf{D}}$ which is an unbiased observation of \mathbf{D} whose error has Gaussian distribution

with covariance matrix \mathbf{C}_D . Now the updated estimate of \mathbf{X} after using the new measurements is the one which minimizes the following function (using the maximum likelihood criterion, assuming independent errors):

$$W = (\mathbf{X} - \mathbf{X}_0)^t \mathbf{C}_{X_0}^{-1} (\mathbf{X} - \mathbf{X}_0) + (\bar{\mathbf{D}} - \mathbf{H}\mathbf{X})^t \mathbf{C}_D^{-1} (\bar{\mathbf{D}} - \mathbf{H}\mathbf{X}). \quad (7.4)$$

The solution can be derived as

$$\mathbf{X} = (\mathbf{C}_{X_0}^{-1} + \mathbf{H}^t \mathbf{C}_D^{-1} \mathbf{H})^{-1} (\mathbf{C}_{X_0}^{-1} \mathbf{X}_0 + \mathbf{H}^t \mathbf{C}_D^{-1} \bar{\mathbf{D}}) \quad (7.5)$$

and the new covariance of \mathbf{X} is

$$\mathbf{C}_X = (\mathbf{C}_{X_0}^{-1} + \mathbf{H}^t \mathbf{C}_D^{-1} \mathbf{H})^{-1}. \quad (7.6)$$

A convenient way of updating \mathbf{X} and \mathbf{C}_X is to maintain a matrix $\mathbf{G} = \sum \mathbf{H}^t \mathbf{C}_D^{-1} \mathbf{H}$ and a vector $\mathbf{B} = \sum \mathbf{H}^t \mathbf{C}_D^{-1} \bar{\mathbf{D}}$ (the summation is over different sets of measurements). Then at each step, the updating algorithm is the following: First increase the dimensions of \mathbf{G} and \mathbf{B} to make room for the new pose X_n . Update \mathbf{G} and \mathbf{B} as

$$\mathbf{G} \leftarrow \mathbf{G} + \mathbf{H}^t \mathbf{C}_D^{-1} \mathbf{H} \quad (7.7)$$

$$\mathbf{B} \leftarrow \mathbf{B} + \mathbf{H}^t \mathbf{C}_D^{-1} \bar{\mathbf{D}}. \quad (7.8)$$

Then the new \mathbf{X} and \mathbf{C}_X are given by

$$\mathbf{X} = \mathbf{G}^{-1} \mathbf{B}; \quad \mathbf{C}_X = \mathbf{G}^{-1}. \quad (7.9)$$

There is one potential problem with the above sequential updating procedure that the state variable \mathbf{X} keeps expanding as it is augmented by a new state at each step. In case the robot path is very long, the variable size may become too large, causing storage or performance problems. A possible solution is to delete some of the old variables while adding the new ones.

We propose a strategy of reducing the size of the state variables as the following. In order to choose a pose to be deleted, we check all pairs of poses and find a pair (X_i, X_j) where the correlation between the two poses is the strongest. We then force the relative pose between X_i and X_j to be fixed as a constant. Then X_i can be deleted from the state variables as it can be obtained from X_j . When deleting the node X_i from the network, we transform any link (X_i, X_k) into a link from X_j to X_k . Note that the covariance matrix \mathbf{C}_X contains all the pairwise covariance between any two poses. A correlation ratio between two poses can be computed from the covariance and variance.

By only fixing some relative poses, the basic structure in the network is still maintained. Thus we are still able to consistently update all the pose variables once given new measurements. This strategy is more flexible than the simple strategy of fixing selected absolute poses as constants.

7.6 Discussion

The experiments demonstrated that our estimation procedure is effective in maintaining consistency among multiple range scans, provided that there are enough overlap in the robot path to form loops in the network of constraints. We have shown that the optimal estimation procedure can be conveniently implemented. The most expensive operation, besides pairwise scan matching, is to compute the inverse of an $3n \times 3n$ matrix. Although the number of poses n may be large for a long robot path, we may heuristically limit it to speed up the computation. The sequential procedure enables the robot to continuously maintain the optimal registration result.

Part IV

Other Shape Registration Applications

In Part II of the thesis, we proposed two algorithms for matching range scans for robot pose estimation. If we regard that a range scan describes a shape of the environment contour, the registration algorithms can be essentially considered as techniques for aligning shapes. In fact, these techniques can also be applied to other types of sensor data or application domains. Especially, they are suitable for matching free-form curved shapes. In this part of the thesis, we will examine our algorithms as well as other methods for shape registration in various other domains. Part of this work has been previously reported in [82].

In Chapter 8, we consider the registration of range image which describe a 3D surface.

In Chapter 9, we study the registration of planar image shapes.

Chapter 8

Registration of Range Images

8.1 Introduction

The problem of range image registration is to determine a 3D rigid transformation (including a 3D rotation and a 3D translation) to apply to the surface described by the range image such that, after the transformation, the data surface aligns with a model surface. The degree of alignment or similarity between the two surfaces is usually expressed in terms of a least-squares error measure.

Some of the methods for 3D shape registration have been reviewed in Section 2.4.3. Here we are particularly interested in the type of approach based on iterative optimization which are suitable for matching free-form curved shapes. Typically, these methods iteratively establish correspondences of low level primitives (usually raw data points or oriented points) to targets on the model and then solve a least-squares problem to minimize the distance between the data shape and the model. The advantage of this approach is that it does not require the extraction or association of localized high level features, while the disadvantage is that these methods require a good initial estimate of the transformation to converge to the optimal solution.

We study several iterative registration methods in the following sections. These methods can be categorized into two types according to whether tangent information is used.

8.2 Point Correspondence Based Methods

We first discuss registrations methods which only use a set of correspondence points, but not tangent information, to solve for the relative transformation. The methods in this category are the ICP algorithm and IDC algorithm we discussed in chapter 4. Here we consider the 3D versions of these algorithms.

8.2.1 Least-Squares Solution

Assume that we have n pairs of 3D correspondence points (P_i, P'_i) . In each pair, P_i and P'_i differ from each other by a 3D rigid motion. Let \mathbf{R} be the 3D rotation matrix and T be the 3D translation vector. The relationship between a pair of points is:

$$P'_i = \mathbf{R}P_i + T. \quad (8.1)$$

The variables \mathbf{R} and T can be solve in least-squares sense by minimizing the following distance function:

$$E_3(\mathbf{R}, T) = \sum_{i=1}^n |\mathbf{R}P_i + T - P'_i|^2. \quad (8.2)$$

The solution of \mathbf{R} using quaternions is given in [12]. This formulation requires to solve the eigenvector corresponding to the maximum eigenvalue of a 4x4 matrix. The solution of T can be obtained in terms of \mathbf{R} . Another solution to this problem is based on singular value decomposition [65].

We derive the solution again as the following. Let \mathbf{M} be the cross-covariance matrix:

$$\mathbf{M} = \sum_{i=1}^n P'_i P_i^t - n \bar{P}' \bar{P}^t. \quad (8.3)$$

By using Lagrange multipliers for the constraint $\mathbf{R}\mathbf{R}^t = \mathbf{I}$ in minimizing $E_3(\mathbf{R}, T)$, we can derive that

$$\mathbf{R}\mathbf{S} = \mathbf{M} \quad (8.4)$$

for some symmetric matrix \mathbf{S} . Then \mathbf{R} can be obtained from factorizing \mathbf{M} into a rotation matrix \mathbf{R} and a symmetric matrix \mathbf{S} . This type of factorization is called polar decomposition [112]. It can be solved using the result of singular value decomposition of \mathbf{M} . Let

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (8.5)$$

where \mathbf{U} , \mathbf{V} are orthogonal matrices, $\mathbf{\Sigma}$ is a diagonal matrix. Then the rotation matrix \mathbf{R} is given by

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T. \quad (8.6)$$

We provide an alternative method to solve for the rotation matrix \mathbf{R} . It is based on iteratively symmetrizing the matrix \mathbf{M} . Let $\mathbf{M}_0 = \mathbf{M}$. We make a series of updates to the matrix as $\mathbf{M}_k = \mathbf{Q}_k \mathbf{M}_{k-1}$, where each \mathbf{Q}_k is a rotation matrix. At step k , we select a pair of off-diagonal elements m_{ij} and m_{ji} of \mathbf{M}_{k-1} such that the value $(m_{ij} - m_{ji})/(m_{ii} + m_{jj})$ is largest among all possible

choices of (i, j) . The matrix \mathbf{Q}_k , which represents a rotation along an axis of the coordinate system, is formed by replacing four entries of an identity matrix with the following:

$$q_{ii} = q_{jj} = \cos \theta_k, \quad (8.7)$$

$$q_{ji} = -q_{ij} = \sin \theta_k. \quad (8.8)$$

We will determine the rotation angle θ_k such that in the updated matrix \mathbf{M}_k , the ij element is equal to the ji element. The angle θ_k can be solved as:

$$\theta_k = \tan^{-1} \left(\frac{m_{ij} - m_{ji}}{m_{ii} + m_{jj}} \right). \quad (8.9)$$

Note that the two off-diagonal elements made equal at one step may become non-equal after another step. But the total “asymmetricalness” of the matrix is decreasing. After some steps, the resulting matrix \mathbf{M}_k is an approximation to \mathbf{S} and the total rotation $\mathbf{Q} = \mathbf{Q}_k \mathbf{Q}_{k-1} \cdots \mathbf{Q}_1$ is an approximation to \mathbf{R}^{-1} .

The convergence rate of the above algorithm appears to be quadratic. Experiments show that it typically takes about 9 iterations (3 rounds of sweeps of the off-diagonal elements) to reduce the relative error to within 10^{-7} . This algorithm is flexible in that it can easily trade running time (number of iterations) for accuracy. The algorithm can also be easily extended to higher dimensions.

Once we determine the rotation matrix \mathbf{R} , the translation vector T is given as:

$$T = \bar{P}' - \mathbf{R}\bar{P}. \quad (8.10)$$

8.2.2 Iterative Algorithms

We discussed the 2D versions of the iterative closest point (ICP) algorithm and iterative dual correspondence (IDC) algorithm in Chapter 4. We now discuss possible adaptation of the algorithms to the 3D case.

The ICP algorithm finds the closest point on the model as the correspondence for a data point. This is still the same in the 3D case.

In the IDC algorithm, in addition to finding the closest point for a data point, it requires another correspondence point which has a similar range as that of the data point. A straightforward extension of the matching-range-point rule to 3D is the following: For a data point P , its corresponding point is P' on the model such that: $\arccos \frac{P' \cdot P}{|P'| |P|} \leq B_R$ and $|P'|$ is closest to $|P|$. In other

words, the search region is the part of the surface model within a cone centered at vector P . The maximum angle between a pair of corresponding points is given by a bound B_R of the rotation.

There is a disadvantage of the above 3D matching-range-point rule in that there may be many choices of P' satisfying the rule for one data point P . In fact, these points form a segment of the curve (within the cone) which is the intersection of the model surface with a sphere. To avoid the ambiguity, we state the rule in the following form: The correspondence P' for a data point P is the point on the model surface which minimizes the objective function $(|P'| - |P|)^2 + W(\theta, B_R)$. Here the penalty function W is monotonical in $\theta = \arccos \frac{P' \cdot P}{|P'| |P|}$. The function value is small when $\theta \leq B_R$ and becomes very large otherwise. A typical choice of W is a power function.

For the iterative algorithm, we need to select the parameter value B_R for every iteration. In the same way as we did for the 2D algorithm, we choose a series of exponentially decreasing values for B_R .

If the model surface is in analytical form, the correspondence points can be easily determined by minimizing the objective functions using a standard optimization procedure (such as the Newton's algorithm).

We present some experiments to compare the 3D IDC algorithm with the ICP algorithm. In the first example, the 3D surface model is an ellipsoid: $(x/a)^2 + (y/b)^2 + (z/c)^2 = 1$, where $a = 5, b = 4, c = 3$. We randomly choose 100 points on the surface as data points which are initially rotated and translated with respect to the model. The rotation axis is $(\frac{2}{3}, -\frac{2}{3}, \frac{1}{3})$. Rotation angle is 9° . Translation is $(0.5, -0.3, -0.1)$. No noise is assumed in this example. We use both the IDC algorithm and the ICP algorithm to correct the rotation and translation errors. The magnitude of rotation and translation residuals from the two algorithms are plotted in Fig. 8.1. It can be seen that, with such a curved model, the ICP algorithm converges very slowly in solving the rotation. Our IDC algorithm significantly improves the convergence speed. The IDC algorithm also converges faster in solving the translation.

In the second example, the surface model consists of the six faces of a rectangular block: $|x| \leq a; |y| \leq b; |z| \leq c$. The choice of points and the initial transformation is the same as in the first example. We again apply both the IDC algorithm and the ICP algorithm to correct the rotation and translation. The magnitude of the residuals are plotted in Fig. 8.2. As now the model is linear, both algorithms perform better than the case of a curved model. While the IDC algorithm still converges much faster than the ICP algorithm in solving the rotation. The two algorithms perform similarly well in solving the translation.

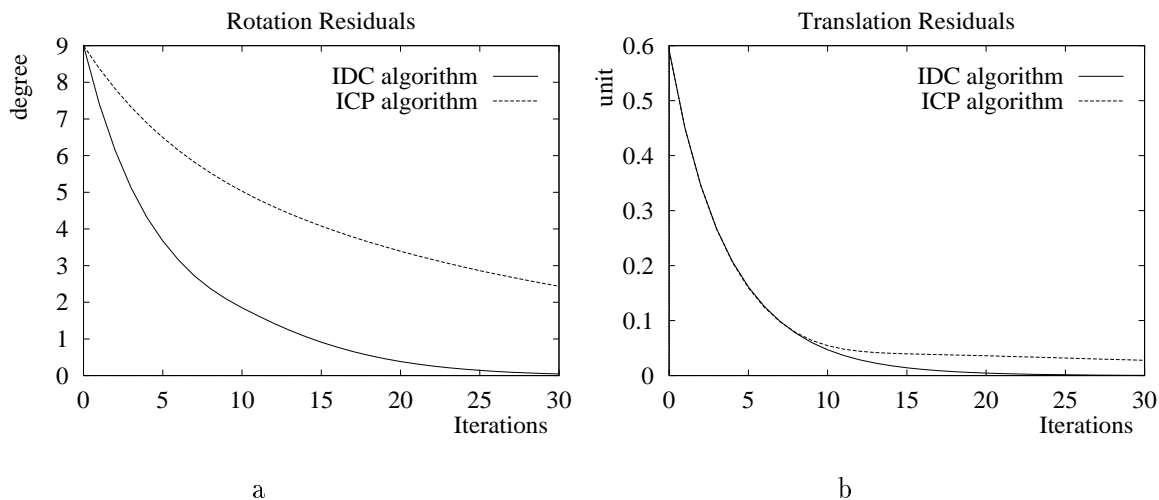


Figure 8.1: Comparison of rotation and translation residuals from ICP and IDC algorithms. (a) Rotation residuals; (b) translation residuals. The model is the surface of an ellipsoid. Notice that the IDC algorithm reduces the rotation residuals much more quickly than the ICP algorithm. The IDC algorithm also reduces the translation residuals more quickly after 10 iterations.

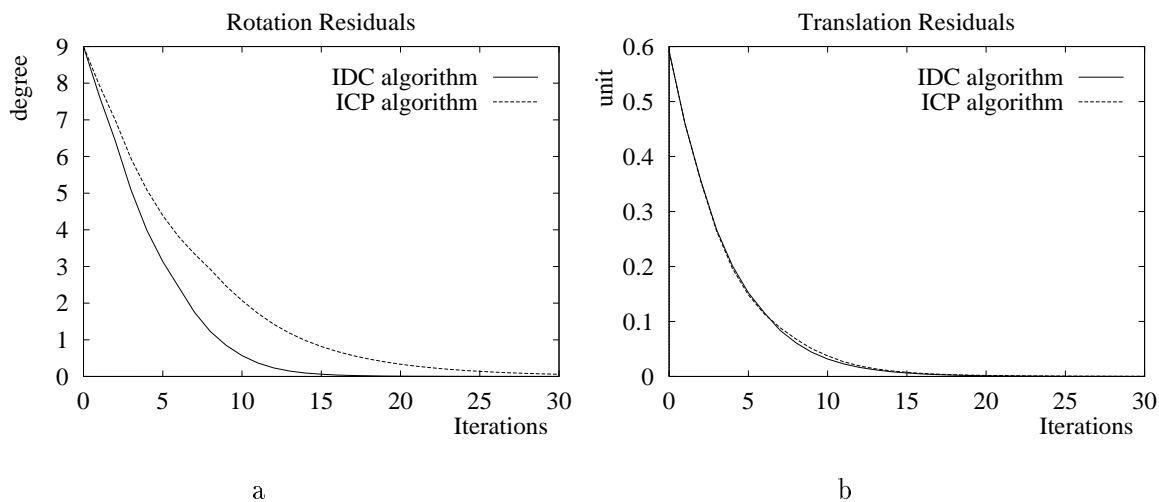


Figure 8.2: Comparison of rotation and translation residuals from ICP and IDC algorithms. (a) Rotation residuals; (b) translation residuals. The model is the surface of a rectangular block. The IDC algorithm still reduces the rotation residuals more quickly than the ICP algorithm. But the translation residuals are reduced in about the same rate in two algorithms.

To summarize, the IDC algorithm solves rotation much more quickly than the ICP algorithm, especially when the model is curved.

If the model surface is not in analytical form, such as in the case of merging partial views where the model itself is a range image, we will be unable to directly use an optimization procedure to locate the correspondence points. To avoid a brute-force search, several approaches can be considered. It has been suggested (in [106]) that a multi-dimensional binary search (proposed by Bentley [11]) may be used to speed up the correspondence search in the ICP algorithm. We believe that it is also possible to represent the range surface in a multi-resolution tree structure to allow a quick reduction in search space during a search for correspondence. Another simple but less efficient way is to sample some directions within the cone and choose an appropriate intersection point of a sample direction with the model as the correspondence point for a data point.

8.3 Methods Using Tangent Information

If the sampling of the range image is dense and if the underlying surface is smooth, surface normal directions may be computed from the data which are very useful to the registration process. Generally, registration algorithms which make use of normal directions are more robust and efficient. Chen and Medioni [26] described an effective method for matching range images of object partial views. The method is based on iteratively minimizing a distance function which is defined from pairs of correspondence points and the surface normal directions at these points. We implemented a similar but slightly improved algorithm and tested on matching real range images. In the following sections, we will discuss this algorithm in detail.

8.3.1 Algorithm Description

Given two views (in terms of range images) of the same surface, the task for the algorithm is to determine a 3D rigid transformation which brings one image into alignment with the other. Let S_p and S_q be the two views of the surface. Let R and T be the rotation matrix and translation vector of the transformation. Then the problem can be defined as finding R and T such that

$$\forall \mathbf{p} \in S_p, \exists \mathbf{q} \in S_q \mid \mathbf{q} = R\mathbf{p} + T. \quad (8.11)$$

The algorithm is summarized as the following:

- Select a set of points (call them control points) \mathbf{p}_i , $i = 1, \dots, N$, from S_p . These points are chosen from smooth areas of the surface. Compute the normal directions \mathbf{n}_i^p of S_p at \mathbf{p}_i .

Transform S_p along with all the control points \mathbf{p}_i and the normal directions \mathbf{n}_i^p by an initial transformation (R_0, T_0) .

- Iterate this step until the process converges. At each iteration k , perform the following operations:
 - For each point \mathbf{p}_i , find the intersection of the normal line defined by \mathbf{p}_i and \mathbf{n}_i^p with the surface S_q . Call this intersection point \mathbf{q}_i . Compute the normal direction \mathbf{n}_i^q of surface S_q at \mathbf{q}_i .
 - Define a distance function (which is the sum of the squared distances from the (transformed) control points to the tangent planes of S_q):

$$E(R_k, T_k) = \sum_{i=1}^N e_i^2 = \sum_{i=1}^N ((R_k \mathbf{p}_i + T_k - \mathbf{q}_i) \cdot \mathbf{n}_i^q)^2. \quad (8.12)$$
 - Find R_k, T_k which minimizes the function $E(R_k, T_k)$.
 - Transform S_p along with all the control points \mathbf{p}_i and the normal directions \mathbf{n}_i^p by the solved transformation (R_k, T_k) .
- The overall transformation (R, T) is the composition of all the transformation $(R_k, T_k), k = 0, 1, \dots$, found in each iteration.

8.3.2 Solution from Minimizing Distance Function

The above algorithm requires minimizing a distance function of the form

$$E(R, T) = \sum_{i=1}^N e_i^2 = \sum_{i=1}^N ((R\mathbf{p}_i + T - \mathbf{q}_i) \cdot \mathbf{n}_i)^2 \quad (8.13)$$

in order to solve for rotation matrix R and translation vector T in an iteration step. The rotation matrix has three degrees of freedom and it can be represented by the three rotation angles, $\theta_x, \theta_y, \theta_z$, around the coordinate axes, as the following:

$$R = \begin{pmatrix} \cos \theta_y \cos \theta_z & \sin \theta_x \sin \theta_y \cos \theta_z - \cos \theta_x \sin \theta_z & \cos \theta_x \sin \theta_y \cos \theta_z + \sin \theta_x \sin \theta_z \\ \cos \theta_y \sin \theta_z & \sin \theta_x \sin \theta_y \sin \theta_z + \cos \theta_x \cos \theta_z & \cos \theta_x \sin \theta_y \sin \theta_z - \sin \theta_x \cos \theta_z \\ -\sin \theta_y & \sin \theta_x \cos \theta_y & \cos \theta_x \cos \theta_y \end{pmatrix}. \quad (8.14)$$

The translation vector T has three coordinate components:

$$T = (t_x, t_y, t_z)^t. \quad (8.15)$$

Chen and Medioni did not explain how the distance function is minimized in [26]. Since the function is continuous and differentiable in the six variables $(\theta_x, \theta_y, \theta_z, t_x, t_y, t_z)$, it is possible to use a standard iterative optimization algorithm such as the Newton's method to search for the values of the variables which minimize the distance function. The difficulty is that it is too tedious to obtain the analytical forms of the second derivatives of the distance function. Search methods which only use first derivatives, such as the conjugate gradient method, can also be used to minimize the function. But they require more iterations. In fact, it is not necessary to accurately find the minimizer because the distance function itself is inaccurate (it is based on a set of approximate correspondence points). We choose to derive a closed-form solution of the variables which only approximately minimize the distance function. This solution should be very efficient to compute while it is as good as the exact solution for improving the correspondence in the next iteration. Thus we only need one level of iteration in the algorithm.

We provide two methods to compute the approximate solution. The first method is based on linearizing each normal distance

$$e_i = (R\mathbf{p}_i + T - \mathbf{q}_i) \cdot \mathbf{n}_i \quad (8.16)$$

in the rotation variables (it is already linear in the translation variables). A first order approximation to the rotation matrix (around $\theta_x = 0, \theta_y = 0, \theta_z = 0$) can be written as:

$$R \approx \begin{pmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{pmatrix}. \quad (8.17)$$

Substituting the approximated rotation matrix into Eq. 8.16, the normal distance can be rewritten in a linear equation of $(\theta_x, \theta_y, \theta_z, t_x, t_y, t_z)$:

$$\begin{aligned} e_i = & (p_{yi}n_{zi} - p_{zi}n_{yi})\theta_x + (p_{zi}n_{xi} - p_{xi}n_{zi})\theta_y + (p_{xi}n_{yi} - p_{yi}n_{xi})\theta_z + n_{xi}t_x + n_{yi}t_y + n_{zi}t_z \\ & - ((q_{xi} - p_{xi})n_{xi} + (q_{yi} - p_{yi})n_{yi} + (q_{zi} - p_{zi})n_{zi}). \end{aligned} \quad (8.18)$$

Then by minimizing $\sum_{i=1}^N e_i^2$ using the least-squares method, the variables $(\theta_x, \theta_y, \theta_z, t_x, t_y, t_z)$ can be solved from a system of linear equations:

$$(\theta_x, \theta_y, \theta_z, t_x, t_y, t_z)^t = \mathbf{A}^{-1}\mathbf{b} \quad (8.19)$$

where

$$\mathbf{A}_{jk} = \sum_{i=1}^N c_{ji}c_{ki}, \quad 1 \leq j, k \leq 6$$

$$\begin{aligned}
\mathbf{b}_j &= \sum_{i=1}^N c_{ji} d_i, \quad 1 \leq j \leq 6 \\
c_{1i} &= p_{yi} n_{zi} - p_{zi} n_{yi} \\
c_{2i} &= p_{zi} n_{xi} - p_{xi} n_{zi} \\
c_{3i} &= p_{xi} n_{yi} - p_{yi} n_{xi} \\
c_{4i} &= n_{xi} \\
c_{5i} &= n_{yi} \\
c_{6i} &= n_{zi} \\
d_i &= (q_{xi} - p_{xi}) n_{xi} + (q_{yi} - p_{yi}) n_{yi} + (q_{zi} - p_{zi}) n_{zi}.
\end{aligned}$$

Although the approximated matrix R in Eq. 8.17 is not a true rotation matrix, we use the true rotation matrix in the form of Eq. 8.14 as defined by the $\theta_x, \theta_y, \theta_z$ to correct the pose error.

We also have a second method of finding a closed-form approximate solution to the transformation variables. It is derived by applying one iteration of the Newton's method to the distance function E as in Eq. 8.13. Let \mathbf{x} be the vector containing the six variables. The initial value \mathbf{x}_0 of the variables is a zero vector (identity transformation). From the Newton's iteration, a new estimate of the transformation variables is given by:

$$\mathbf{x} = \mathbf{x}_0 - \left(\frac{\partial^2 E}{\partial \mathbf{x}^2} \right)^{-1} \left(\frac{\partial E}{\partial \mathbf{x}} \right) \quad (8.20)$$

where the first derivative is a 6-dimensional vector and the second derivative is a 6×6 matrix, both evaluated at $\mathbf{x}_0 = \mathbf{0}$. After computing and evaluating the derivatives, the solution vector \mathbf{x} can be expressed in the following form:

$$\mathbf{x} = (\mathbf{A} + \mathbf{A}')^{-1} \mathbf{b} \quad (8.21)$$

where matrix \mathbf{A} and vector \mathbf{b} are the same as the ones in the least-squares solution. There is an additional 6×6 matrix \mathbf{A}' in the solution whose entries are:

$$\begin{aligned}
\mathbf{A}'_{11} &= \sum_{i=1}^N -e_i (p_{yi} n_{yi} + p_{zi} n_{zi}) \\
\mathbf{A}'_{22} &= \sum_{i=1}^N -e_i (p_{xi} n_{xi} + p_{zi} n_{zi}) \\
\mathbf{A}'_{33} &= \sum_{i=1}^N -e_i (p_{xi} n_{xi} + p_{yi} n_{yi})
\end{aligned}$$

$$\begin{aligned}
\mathbf{A}'_{12} &= \mathbf{A}'_{21} = \sum_{i=1}^N e_i p_{y_i} n_{x_i} \\
\mathbf{A}'_{13} &= \mathbf{A}'_{31} = \sum_{i=1}^N e_i p_{z_i} n_{x_i} \\
\mathbf{A}'_{23} &= \mathbf{A}'_{32} = \sum_{i=1}^N e_i p_{z_i} n_{y_i} \\
\mathbf{A}'_{jk} &= 0, \quad \text{for } j > 3 \text{ or } k > 3.
\end{aligned}$$

That is, only the first 3×3 submatrix of \mathbf{A}' is non-zero.

The new solution is very similar in form to the least-squares solution except the additional coefficient matrix \mathbf{A}' whose values are contributed by the second derivatives with respect to the rotation variables, while this part is ignored in the least-squares solution.

We have described two methods for solving an update to the transformation variables. When used with the iterative matching algorithm, both methods lead to very quick convergence (typically less than 20 iterations to converge to machine accuracy). The second method is slightly better as it typically saves about one third of the iterations required by the first method.

8.3.3 Experiments

We apply the algorithm described in the last two sections to register pairs of range images of human face. These range data are obtained from a database provided by the National Research Council of Canada. Each pair of images consist of a front view and a side view of a human face. The size of each image is 150x256.

Because there is a large difference in viewing angles (approximately 90°) between a pair of images, the overlapping area from the two images is relatively small. The registration task is thus harder than the examples in [26] where a pair of images only differ by a small viewing angle (about 15°). One particular problem is that a region of the surface orthogonal to the viewing direction in one image becomes almost parallel to the viewing direction in another image, introducing depth discontinuities. Normally we connect adjacent grid points in a range image by triangles to form a continuous surface. But this triangle interpolation will be very inaccurate at the depth discontinuities. We try to avoid this problem by leaving the depth discontinuities as “holes” in the surface rather than interpolating across them.

In the algorithm, control points are first selected from the smooth regions in one image. In our experiments, we typically have about 50-100 control points. By using the transformation derived

in the previous iteration, correspondences for the control points can be located by finding the intersection points of the normal lines (defined at the control point on the transformed first image) with the surface in the second image. Finding the intersection point of a line with a range surface is achieved by first searching for a triangle that contains the intersection point and then solving the intersection in closed-form. The correspondence points are tested for outliers by two criteria: the distance between a pair of point and the difference in surface normal directions at the two points. We choose a fixed percentile p (which is 75% in our experiments) and we accept a pair of correspondence points only when both criteria for the pair are among the best p percentile among all correspondence pairs. Otherwise, the pair is discarded. Finally, a correction to the transformation is solved by using the formula in the previous section and the transformation is updated. This process repeats until the correction is small enough. It typically takes about 10-15 iterations for the correction to reach 10^{-10} .

The results of two experiments are presented in Figure 8.3 and 8.4. In each figure, part (a) and (b) are the two views of the surface (rendered with shades). Part (c) is the result of merging two images, as seen from the middle direction between the two views. Part (d) and (e) are the same as part (a) and (b) except that they are seen from the same direction as part (c). Thus part (d) and (e) contain the contributions of the two images to the merged view.

8.3.4 Rotation Search Method

The iterative algorithm we described early is efficient in correcting small registration errors. The derived answer is usually also very accurate. But one limitation of the algorithm is that a good initial estimate of the transformation must be provided to ensure that the algorithm converges. We experimented with the face range images and noticed that the maximum acceptable rotation error is about 3 to 5 degrees. If the initial estimate of transformation has a rotational error larger than that amount, the algorithm usually fails to converge. We explain this failure by the algorithm's inability to provide a correction to the transformation error from slightly inexact correspondences, which are partly due to the facts that the surface is not smooth enough and the linear approximation to the rotation matrix is inadequate.

In Chapter 3, we introduced a rotation search algorithm for matching range scans. In that algorithm, the 2D rotation angle is solved in a search procedure rather than iteratively updated by least-squares solutions. Thus the algorithm has an advantage that it works even without a good initial estimation of the rotation. We naturally hope to extend this algorithm to the 3D case for

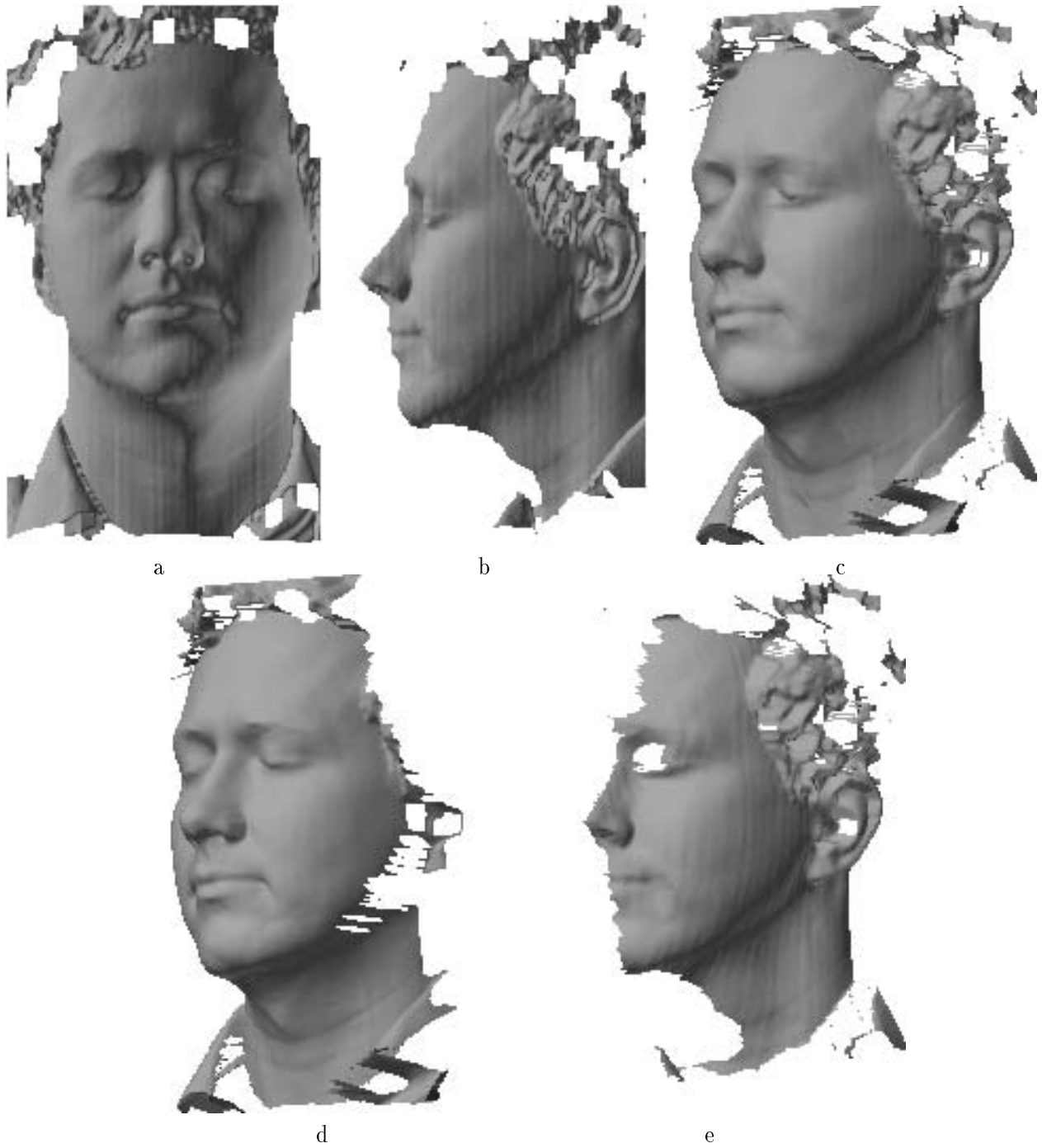


Figure 8.3: Merging two views of a face range surface: (a) Front view; (b) side view; (c) merge of the two views (as seen from a 45° angle); (d) the front view image seen from the same angle as in (c); (e) the side view image seen from the same angle as in (c).

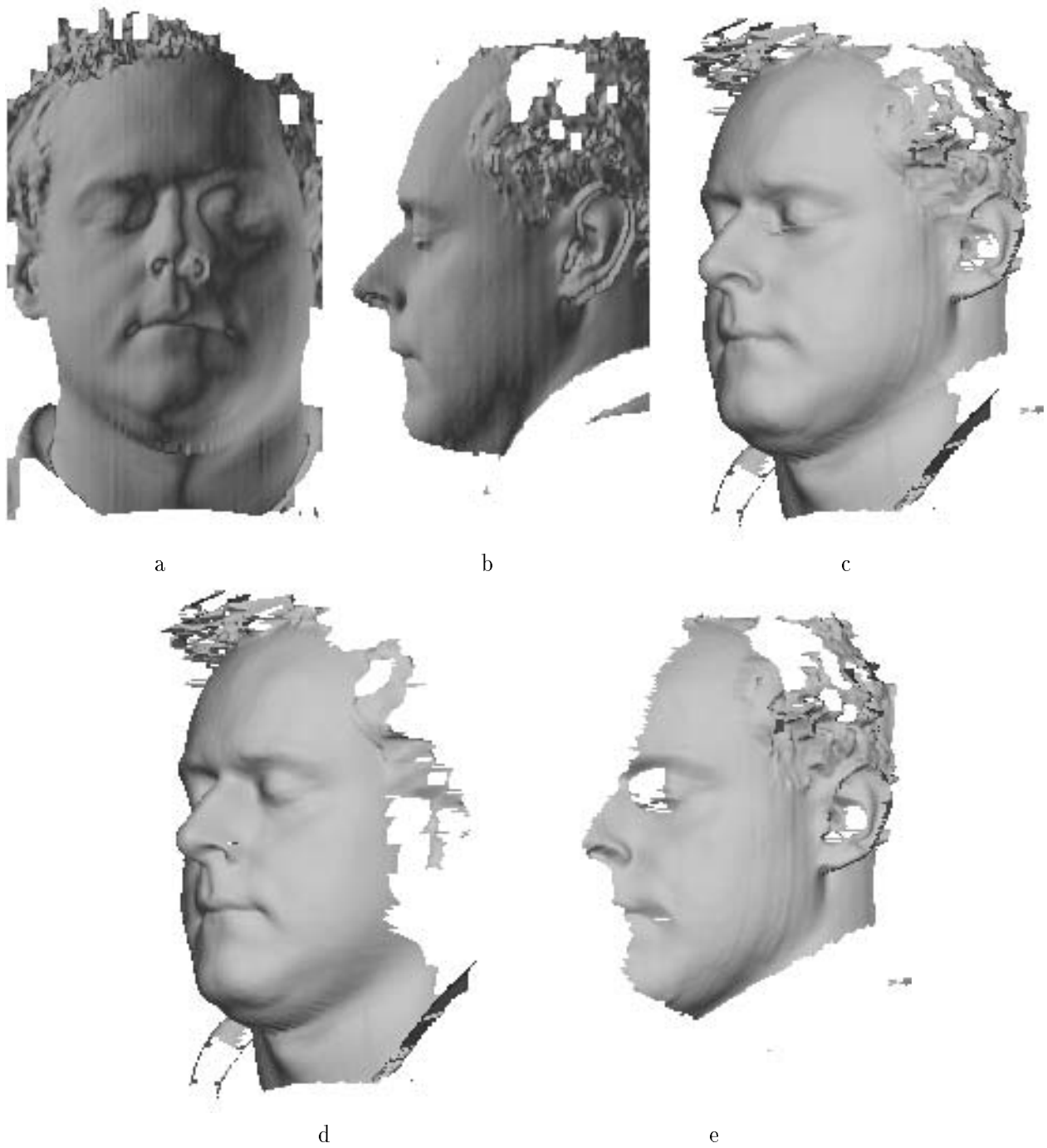


Figure 8.4: Merging two views of a face range surface: (a) Front view; (b) side view; (c) merge of the two views (as seen from a 45° angle); (d) the front view image seen from the same angle as in (c); (e) the side view image seen from the same angle as in (c).

matching range images to lessen the strict requirement of good initial estimate.

If there is a systematic way of searching (guessing) for the 3D rotation, the extension of the algorithm to the 3D case is straightforward. Similar to the 2D case, we can associate correspondences based on a guessed rotation while outliers in correspondences are detected using fixed thresholds. We then set up a linear system to solve for the 3D translation vector. At the same time, a distance measure between the two images can be defined using the least-squares error and the penalty for the outliers. This distance will be used to control the search procedure. However, the major difficulty in the 3D case is that a 3D rotation has three degrees of freedom. Thus the searching procedure will be less efficient comparing with the 2D case where the rotation has only one degree of freedom.

The rotation search method will be readily applicable, however, if the axis of the 3D rotation is known in advance. In this case, the only degree of freedom in the rotation is the rotation magnitude around the known axis and it can be solved easily in the search procedure just like the 2D case. In fact, in many applications, the rotation axis is approximately available as the object (or sensor) is often placed in a turntable (with known configuration) for taking images of partial views. In our face image examples, we know that the rotation between a front view and a side view is approximately around the vertical axis. Then we can apply the search method to determine the rotation angle.

We tested the rotation search algorithm with the face images. Assuming a vertical rotation axis, we try an initial rotation error of 20° . The initial translation is automatically determined by the algorithm by aligning the center of mass of the two images. In the experiments with both pairs of images as in Fig. 8.3 and Fig. 8.4, the search algorithm successfully located the rotation angle in about 15 iterations. The remaining rotation errors are less than 3° (mostly because the rotation axis is not exactly vertical). When the search results are used as initial estimations for the least-squares iterations, the algorithm converges in both examples.

In fact, the search procedure is capable of finding an arbitrarily large initial rotation angle by first sampling all possible rotation angles to determine a starting point, as explained in Chapter 3. When the trial rotation is far away from the true rotation, the distance function must have a very large value because of the incorrect correspondences and the penalties for outliers.

In conclusion, if the axis of the 3D rotation is known (which is often the case in many applications), the rotation search algorithm may be first used to find an approximate registration of two images even if no other knowledge about the relative transformation is available. Then the least-squares iterations may be used to fine tune the solution. The combination gives a flexible, inexpensive, and effective way for registering range images of object partial views.

Chapter 9

Registration of Planar Image Shapes

9.1 Introduction

In the previous chapters, we studied the problem of matching 2D or 3D shapes which are described by range measurements. Here we continue our study on shape registration. But we now focus on planar shapes (curves) from intensity images. The new issues involved in dealing with image shapes are the preprocessing step of extracting features (representing the shape contour) and the scaling factor in the transformation.

The planar image shape registration problem can be stated as one of determining a similarity transformation (consisting of 2D rotation, 2D translation, and uniform scaling) for the data shape such that, after the transformation, the data shape aligns with a model shape. Here the data shape is typically represented by points or lines on the contour curve of the shape as extracted from the image. We have reviewed some of the techniques for 2D shape registration in Section 2.4.2. One common approach is based on searching for correspondences to interpret the features. But if the shape consists of smooth but noisy free-form curve, it can be difficult to interpret features (such as lines) segmented from the curve. Approaches that use arc-length as a reference to associate correspondence will also have problems if the shape is incomplete due to noise or occlusion. Here, we are again interested in the iterative optimization based approach which avoids the use of localized features.

In Part II of the thesis, we presented two such algorithms for matching range scans. We now present new versions of the two algorithms for matching planar image shapes. Here we consider that a model shape is available which is in the form of an analytical curve (splines). The data shape consists of edge points in the image extracted by an edge detector. First, we study the rotation

search/least-squares algorithm where edge points together with their gradient directions are used in defining a distance function. Next, we consider the iterative point to point correspondence algorithm which only requires unoriented edge points in the data shape. Experiments with both algorithms will be presented.

9.2 Rotation Search/Least-Squares Method

9.2.1 Problem Formulation

Given an image of a planar shape and a model of the same shape, the problem is to determine the pose of the shape in the image. Assuming that the shape contour is depicted by the change in image intensity, we can extract the contour points using an edge detector (such as the Canny edge detector). At each extracted edge point, we can also estimate the contour normal direction from the gradient of intensity. Thus the data shape is represented by a set of oriented points.

Assume that we also have an accurate model of the shape which is represented as a parameterized continuous curve. If the extracted edge points indeed belong to the contour curve, then for some similarity transformation (consisting of a rotation of angle ω , a scaling factor s , and a translation vector T), the edge points should be transformed into their corresponding positions on the model. Formally, for each edge point $P_i = (x_i, y_i)^t$ with normal direction \vec{n}_i , there is a point $P(u) = (x(u), y(u))^t$ on the model such that

$$P(u) = sR_\omega P_i + T \quad (9.1)$$

$$\vec{n}(u) = R_\omega \vec{n}_i. \quad (9.2)$$

where R_ω is the rotation matrix, u is the parameter of the model, $\vec{n}(u)$ is the curve normal vector at $P(u)$.

Now we can formulate the matching problem as to determine the transformation (ω, s, T) , such that the above equations hold for all the edge points.

The algorithm is essentially the same as the one presented in Chapter 3. The least-squares error is slightly different because of the additional scaling factor. We also used a different method in deriving approximate correspondence, although the resulting form of correspondence is still the same. We will discuss the correspondence definition and least-squares formulation next.

9.2.2 Correspondence Definition and Least-Squares Matching Distance

We will define the least-squares matching distance between the data set and the model, as a function of the rotation, scaling, and translation variables.

Assume that the model is a closed, continuous curve represented in a local coordinate system whose origin is at the center of the area enclosed by the curve model. We represent the set of data points, P_i , $i = 1, \dots, n$, in an object coordinate system with the origin at the center of the shape. Note that, from the set of data points, if we can exactly locate the same center of shape as that in the model, the translation component in the relative coordinate transformation between the two shapes will be zero. But since we usually do not have a complete set of contour points of the shape and there may be outliers, we can only obtain a rough estimate of the shape center. In this case, there will be a (hopefully small) translation T in Equation 9.3. It is also possible that a better estimate of the transformation is available from another procedure or from *a priori* knowledge.

For a point $P = (x, y)^t$ on a curve shape (either data or model), let the normal vector of the curve at P be $\vec{n} = (\cos \phi, \sin \phi)^t$ where ϕ is the normal angle. Define $\rho = \vec{n} \cdot P$ which is the distance from origin to the tangent line at P (see Fig. 9.1(a) for illustration). Note that ρ is invariant when the curve shape rotates around the origin. The pair (ϕ, ρ) defines the tangent line at P . We also represent a point P on the curve in polar coordinate system as (θ, r) . Thus both the point P and the tangent line parameters (ϕ, ρ) at P can be considered as functions of the polar angle θ . For the curve model, we assume that it uses θ as its parameter.

Now consider an oriented data point $(P_i = (x_i, y_i)^t, \vec{n}_i)$. It is related to its corresponding point $(P(\theta), \vec{n}(\theta))$ on the model curve by:

$$P(\theta) = sR_\omega P_i + T \quad (9.3)$$

$$\vec{n}(\theta) = R_\omega \vec{n}_i. \quad (9.4)$$

We can derive the relationship in terms of ϕ and ρ from the above equations as (note $\rho_i = P_i \cdot \vec{n}_i$):

$$\phi(\theta) = \phi_i + \omega, \quad (9.5)$$

$$\rho(\theta) = s\rho_i + (R_\omega \vec{n}_i) \cdot T. \quad (9.6)$$

From Equation 9.3, we can also derive the relationship of θ and θ_i (the polar angles of P and P_i) as the following (see Fig. 9.1(b) for illustration):

$$\theta = \theta_i + \omega + \Delta\theta_T, \quad (9.7)$$

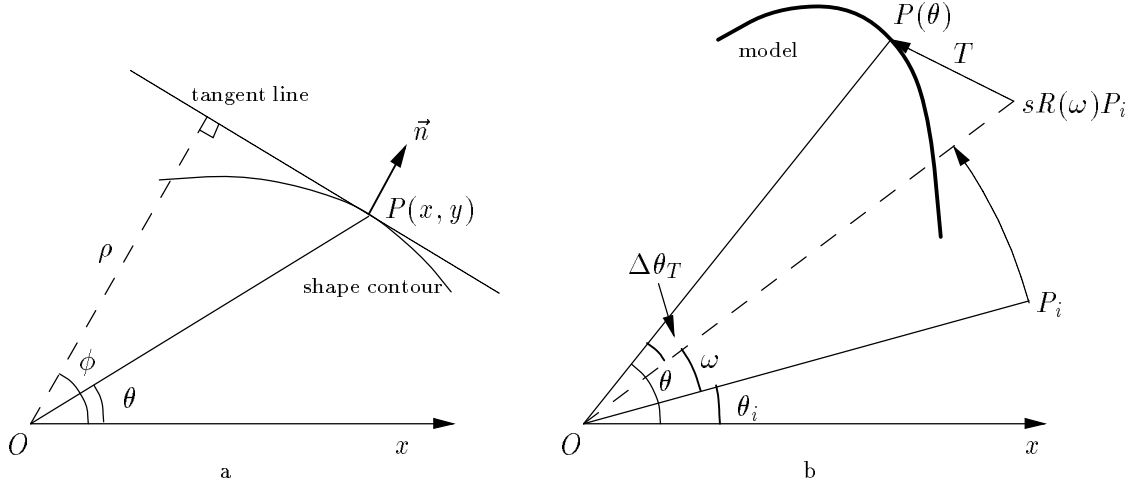


Figure 9.1: (a) The tangent parameters defined at a data point P on the shape contour. (b) Point P_i is mapped to $P(\theta)$ on the model by a similarity transformation.

where $\Delta\theta_T$ is a term introduced by the translation T , which is small when $|T| \ll |P|$.

With θ being the reference parameter, Equation 9.7 indicates that the corresponding point for P_i is a point on the model which has polar angle θ . This relationship depends on many variables (ω , T , and also implicitly s), which is inconvenient for the correspondence search. We plan to combine equation 9.5 and 9.6 to cancel out $\Delta\theta_T$, so that the correspondence depends on ω only.

Considering that $\Delta\theta_T$ is small, we expand $\phi(\theta) = \phi(\theta_i + \omega + \Delta\theta_T)$, and $\rho(\theta) = \rho(\theta_i + \omega + \Delta\theta_T)$ by Taylor series to the first order at $\theta_i + \omega$ in Equation 9.5 and 9.6:

$$\alpha \stackrel{\text{def}}{=} \theta_i + \omega$$

$$\phi(\alpha) + \Delta\theta_T \phi'(\alpha) \approx \phi_i + \omega \quad (9.8)$$

$$\rho(\alpha) + \Delta\theta_T \rho'(\alpha) \approx s\rho_i + (R_\omega \vec{n}_i) \cdot T. \quad (9.9)$$

Then we combine Equation 9.8 and 9.9 to cancel the derivatives $\phi'(\alpha)$ and $\rho'(\alpha)$, by using the following identity (derived in Appendix B):

$$\rho'(\alpha) = \phi'(\alpha) \rho(\alpha) \tan(\alpha - \phi(\alpha)). \quad (9.10)$$

Let $\lambda = \rho(\alpha) \tan(\alpha - \phi(\alpha))$. The linear combination (Eq.9.9) $-\lambda$ (Eq.9.8) should cancel out the terms involving $\phi'(\alpha)$ and $\rho'(\alpha)$. The final equation can be rearranged into the form:

$$C_i^s s + C_i^x T_x + C_i^y T_y = D_i \quad (9.11)$$

with the coefficients

$$C_i^s = \rho_i \quad (9.12)$$

$$C_i^x = \cos(\phi_i + \omega) \quad (9.13)$$

$$C_i^y = \sin(\phi_i + \omega) \quad (9.14)$$

$$D_i = \rho(\alpha) - \lambda(\phi(\alpha) - (\phi_i + \omega)), \quad (9.15)$$

where

$$\lambda = \rho(\alpha) \tan(\alpha - \phi(\alpha))$$

$$\alpha = \theta_i + \omega.$$

Now we have an equation (Eq. 9.11) which relates the data point P_i (which defines $(\theta_i, \phi_i, \rho_i)$), to a point $P(\theta_i + \omega)$ on the model. This correspondence is given by $\theta_i + \omega$ which is determined by the rotation angle ω only.

Equation 9.11 also defines a linear equation with respect to s and T , since all the coefficients C_i, D_i only involve ω and the correspondence pair. For many such equations defined on n correspondence pairs, we can form a distance function as the following:

$$E(\omega, s, T) = \sum_{i=1}^n (C_i^s s + C_i^x T_x + C_i^y T_y - D_i)^2. \quad (9.16)$$

We can then solve for s, T in terms of ω and also obtain a least-squares error $\min_{(s,T)} E(\omega, s, T)$.

9.2.3 Curve Model and Correspondence

We represent the model shape as a closed curve in the form of piecewise splines. A cubic B-spline is used to represent each segment. The whole curve is defined on a parameter u for $0 \leq u < N_s$ (where N_s is the number of segments). The spline model is derived from an image which contains a clean contour of the shape. We fit a spline curve to a list of contour pixels using an optimal, adaptive fitting procedure (which is described in our previous work [83]).

For a given oriented data point P_i which defines $(\theta_i, \phi_i, \rho_i)$, we need to determine a corresponding point on the model which has a polar angle $\alpha = \theta_i + \omega$, as we derived in the previous section. The problem of correspondence search is therefore to find a curve point $(x(u), y(u))$ such that:

$$\frac{y(u)}{x(u)} = \tan \alpha. \quad (9.17)$$

We can easily search the spline segments to find the ones which intersect with the ray of angle α . The parameter u at the intersection points can be computed by solving a cubic equation.

Once we find the correspondence point $P(\alpha) = (x(u), y(u))^t$, the parameters of the tangent line at that point can be computed as:

$$\phi(\alpha) = -\tan^{-1} \frac{x'(u)}{y'(u)} \quad (9.18)$$

$$\rho(\alpha) = x(u) \cos \phi(\alpha) + y(u) \sin \phi(\alpha). \quad (9.19)$$

In case that the model is not a convex curve, there may be more than one point on the curve which has the same angle α . We also need to consider the case of outlier where P_i does not have a corresponding point on the model. We use the following conditions (for some thresholds H_ϕ and H_ρ) to check all the points at direction α :

$$|\phi(\alpha) - (\phi_i + \omega)| < H_\phi; \quad \text{and} \quad |D_i - \rho_i| < H_\rho. \quad (9.20)$$

If more than one point satisfies the conditions, we select the one with the smallest $|D_i - \rho_i|$ value as the correspondence of P_i . If none of them satisfy the conditions, we consider P_i as an outlier and we do not include it in the summation in Equation 9.16. By doing this, we effectively reject outliers in the set of data points.

Note that $D_i = \rho(\alpha) - \lambda(\phi(\alpha) - (\phi_i + \omega))$ can be considered as an adjusted value of $\rho(\alpha)$ to approximate ρ_i when the orientation ϕ_i is slightly biased. Although we derived it as a result of canceling out the derivatives ϕ' and ρ' , this adjusted value also has the potential of reducing noises in the orientations of the data points.

Also note that we are assuming the scaling factor $s \approx 1$ when we test for outliers. Therefore, we need either the actual scaling factor to be close to 1 or the outliers to be far from the model (either in space or in orientation) in order for the outlier detection procedure to work. Similarly, we also require that the translation residual T be relatively small. Sometimes it is also possible to use some global methods (such as moment-based) to estimate the center and the size of the shape.

9.2.4 Search Algorithm

Once we find the correspondences and form the least-squares error (Equation 9.16), we can apply a search routine to solve for the rotation. The rest of the algorithm is the same as in Chapter 3.

We first form a total matching distance function for a given rotation angle ω based on the least-squares error and the number of outliers. Here an outlier is a data point which does not have

a correspondence point on the curve (the conditions in 9.20 are not met). Let n_o be the number of outliers, the total matching distance is defined as:

$$E_{\text{match}}(\omega) = \min_{(s,T)} E(\omega, s, T) + n_o H_\rho^2, \quad (9.21)$$

where H_ρ^2 is the constant cost of an outlier.

We then search the distance function for a minimum along the rotation dimension. At the same time, we also correct the scaling and translation components with the least-squares solution. As the transformation is updated, the distance function is in fact redefined with less significant scaling and translation, and therefore its minimum along the rotation dimension can be more easily determined.

It may be useful to run the matching procedure two or more times with gradually decreasing values in the thresholds H_ϕ and H_ρ . At first, we choose relatively large thresholds to obtain an approximate transformation, which is derived from a larger set of points including some of the outliers. As the solution approaches the correct transformation, we can reduce the thresholds to exclude more outliers, and in turn we obtain a more accurate estimate of the transformation.

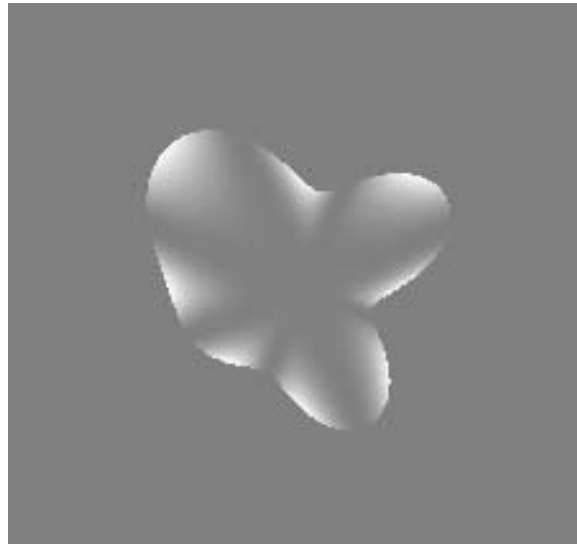
9.2.5 Experimental Results

We present experiments of matching planar shapes which are segmented from synthesized or real images. First, we describe the preprocessing steps in order to extract oriented edge points from an image.

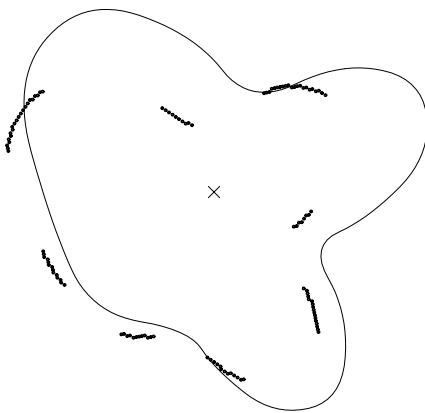
We use a modified Canny edge detector which outputs both the gradient magnitude and orientation at the edge points (whose gradient magnitudes are local maxima). We then threshold the magnitude image to keep only those strong edge points. In the binary image of the remaining edge points, we apply a thinning algorithm to extract a pixel-wide skeleton. Next, we trace each connected segment of the skeleton and smooth the gradient orientation of the points along the segment. We also discard points at which the gradient orientation changes rapidly along the segment (typically these are the corner points on the contour). The resulting set of oriented edge points (x_i, y_i, ϕ_i) are the data set to be matched to the curve model.

Note that since we do not require a complete or connected contour of the shape for matching, we can afford to set a relatively high threshold on the magnitude image to reduce the number of outlier points. Our algorithm is capable of matching a fragmented contour to the model.

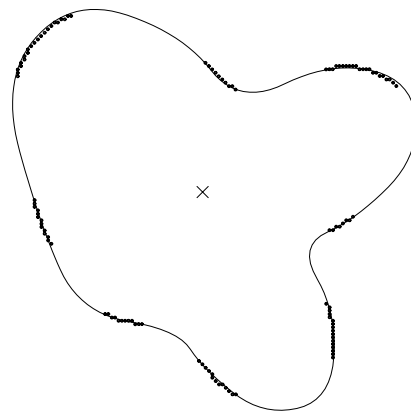
We first present an example using a synthesized image. The image is designed to contain a



a



b



c

Figure 9.2: (a) A planar shape in a synthesized image whose contour is only partially visible. (b) The extracted contour points superimposed on the model. The set of points differ from the model by a similarity transformation. (c) The points are registered to the model as the matching algorithm recovers the relative transformation.

shape whose contour is fragmented and only partially visible (Figure 9.2(a)). We derive a model of the shape from another image. This model consists of 16 spline segments. The initial pose of the image points with respect to the model is shown in (b), where the two shapes differ by a similarity transformation. As shown in (c), the edge points are successfully registered to the model using our matching algorithm. The correctly recovered transformation is: rotation $\omega = 12.0^\circ$, scaling $s = 1.18$, translation $T = (13.0, -6.2)$ pixels (the width of the contour is about 120 pixels).

Next, we show two examples with real images. In example 2, we match the contour curve of a shampoo bottle (Figure 9.3). First, we derive a curve model of the contour from a clean image of the bottle. The derived model consists of 15 spline segments. Then we take another image of the same bottle whose shape is to be registered to the model. The image has a cluttered background (containing another bottle and some blocks). The edge points of the shape, together with many outliers, are then extracted from the image. The image points are initially registered to the model with some pose and scaling error. The final matching result shows that the points are aligned with the model after correcting the pose error.

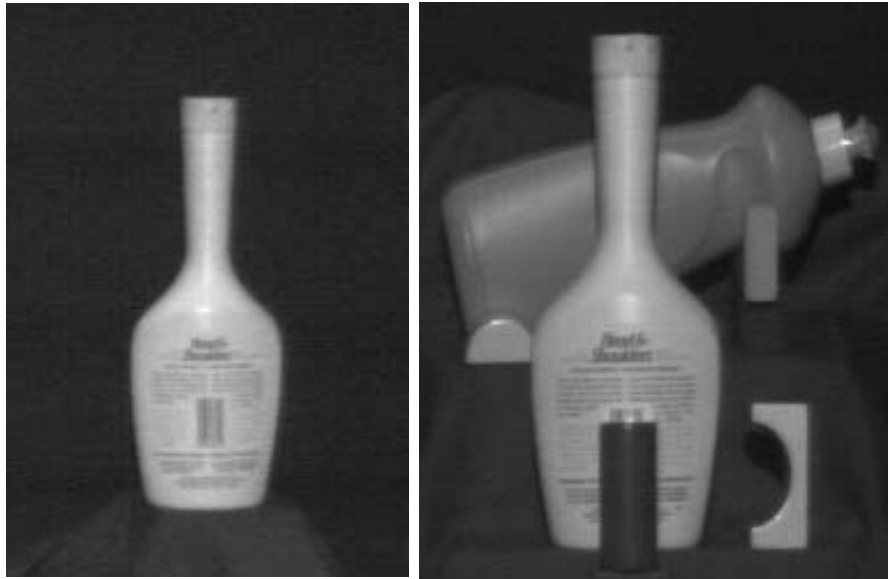
In example 3, we show the result of matching the shape of a maple leaf (Figure 9.4). Again, we derive a spline model of the shape from a clean image. The model consists of 39 segments of splines. The shape to be registered is extracted from an image of the leaf in a very noisy background. This time we try to register the points to the model using a global procedure based on the first and second order moments which are defined on both the set of points and on the curve of the model. Because of the large number of outliers, this global registration is quite inaccurate. However, it is good enough to be a starting point for our matching algorithm.

9.3 Iterative Point Correspondence Based Algorithm

In Chapter 4, we discussed a shape registration method based on iteratively associating point correspondences to derive updates to the transformation. We discussed two rules for associating correspondence, namely the closest point rule and the matching range point rule, and the algorithms based on these rules. Here we apply this method for matching planar image shapes.

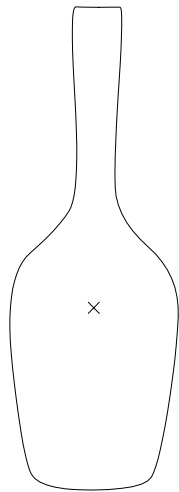
9.3.1 Iterative Least-Squares Procedure

Our task is to determine a relative 2D similarity transformation between a data shape and its model. The data shape is represented by a set of (unoriented) points on the shape contour, while



a

b



c



d

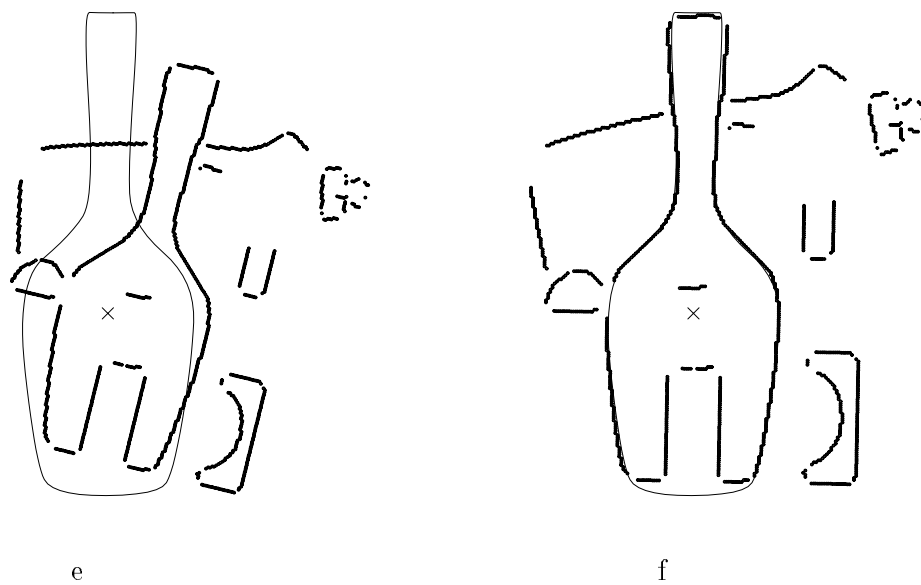
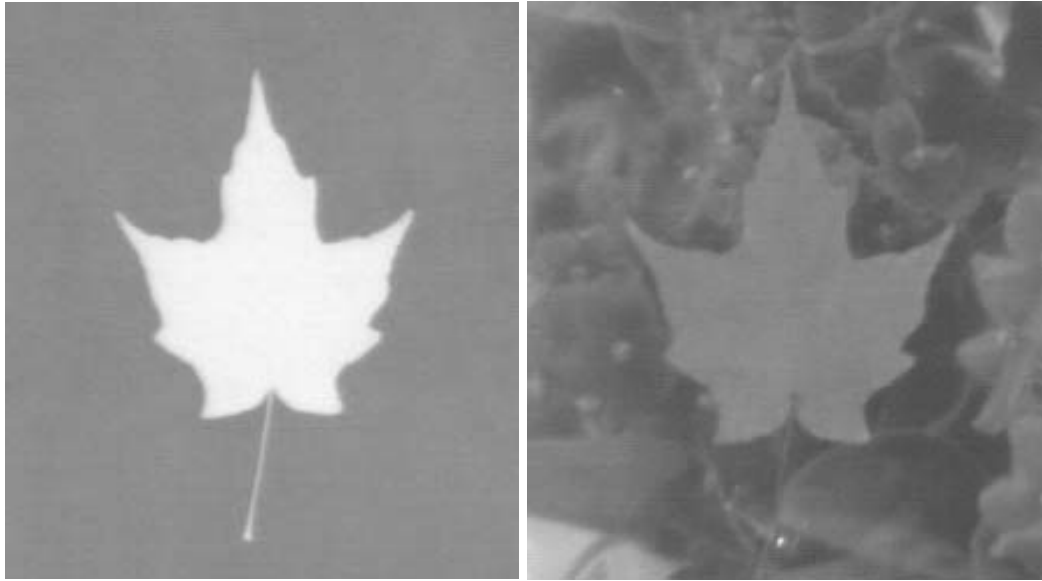
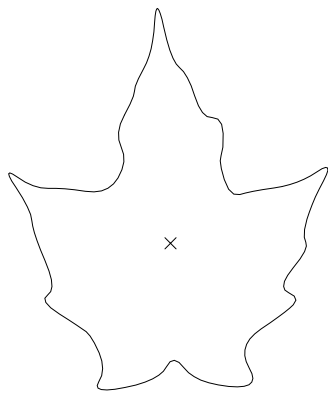


Figure 9.3: (a) An image of a shampoo bottle, used for constructing a curve model. (b) An image of the same bottle, whose contour shape is to be matched to the model, in a scene together with other objects. (c) Spline curve model derived from the image in (a). (d) Edge points extracted from image (b). (e) Initial pose of the points with respect to the model. (f) The points are registered to the model as the matching algorithm recovers the relative transformation.



a

b



c



d



Figure 9.4: (a) An image of a leaf shape from which a contour curve model is derived. (b) An image of the same leaf shape in a cluttered background. (c) The derived curve model from the image in (a). (d) The edge points extracted from the image in (b). (e) The points are initially registered to the model by a moment-based global procedure. (f) The points are matched to the model as the matching algorithm recovers the geometric transformation.

the contour model is assumed to have an analytical curve representation. The transformation consists of a rotation of angle ω , a scaling factor s , and a translation vector $T = (T_x, T_y)^t$.

The algorithm is based on iteratively minimizing the distance from the data points to their corresponding points on the model. Each step of the iterative algorithm consists of the following operations: (1) For each point P_i in the data set, use a rule to determine a corresponding point P'_i on the model; (2) compute a least-squares solution of the relative transformation (rotation, translation, and scaling) based on all the correspondence pairs of points; (3) apply the least-squared solution to transform the data set with respect to the model.

Based on n pairs of corresponding points $(P_i(x_i, y_i), P'_i(x'_i, y'_i)), i = 1, \dots, n$, the least-squared solution for the transformation variables (ω, s, T_x, T_y) can be derived by minimizing the following distance function:

$$E(\omega, s, T) = \sum_{i=1}^n |sR_\omega P_i + T - P'_i|^2, \quad (9.22)$$

where $R_\omega = \begin{pmatrix} \cos \omega & -\sin \omega \\ \sin \omega & \cos \omega \end{pmatrix}$ is the rotation matrix. Closed-form solutions for ω , s , and T can be derived as the following:

$$\begin{aligned} \omega &= \tan^{-1} \frac{S_{xy'} - S_{yx'}}{S_{xx'} + S_{yy'}} \\ s &= \frac{\sqrt{(S_{xy'} - S_{yx'})^2 + (S_{xx'} + S_{yy'})^2}}{S_{x^2} + S_{y^2}} \\ T &= \bar{P}' - sR_\omega \bar{P} \end{aligned}$$

where

$$\begin{aligned} \bar{P} &= \frac{1}{n} \sum_{i=1}^n P_i & \bar{P}' &= \frac{1}{n} \sum_{i=1}^n P'_i \\ \begin{pmatrix} S_{xx'} & S_{xy'} \\ S_{yx'} & S_{yy'} \end{pmatrix} &= \sum_{i=1}^n (P_i - \bar{P})(P'_i - \bar{P}')^t \\ \begin{pmatrix} S_{x^2} & S_{xy} \\ S_{xy} & S_{y^2} \end{pmatrix} &= \sum_{i=1}^n (P_i - \bar{P})(P_i - \bar{P})^t. \end{aligned}$$

We will revisit the two rules and the related algorithms below.

9.3.2 Correspondence Rules and Algorithms

First, we apply the closest point rule. For each data point, its corresponding point is determined as the one on the model which is closest to the data point. We also propose a matching-range point

rule as the following. For a data point P , its corresponding point is P' on the model where P' satisfies: $|\arg P - \arg P'| \leq B_\omega$ and $|P'|$ is closest to $|P|$. The parameter B_ω controls the search area to ensure a reliable correspondence point. Based on the two rules, we have two iterative algorithms (ICP and IMRP).

A better algorithm can be devised by using both the closest point rule and the matching-range point rule. We call it the iterative dual correspondence (IDC) algorithm and it works by obtaining two correspondence points for each data point and solving two sets of transformation variables from the two correspondence sets. Then, the translation and scaling components from the closest-point rule solution and the rotation component from the matching-range-point rule solution are used to form the current solution for the transformation.

The IDC algorithm appears to converge much faster than the ICP algorithm and the IMRP algorithm. We compare the three iterative algorithms using an example. The curve model is an ellipse of width 1000 units. The data points and the model are similar to the ones in Fig. 4.1 in Chapter 4. The initial rotation is -6° ; initial scaling factor is 0.8; and the initial translation is (-5, 5) units. Fig. 9.5 illustrates the residuals of the transformation components during the iterations of the three algorithms. Clearly, the IDC algorithm reduces the residuals much more quickly than the other two single-rule algorithms.

To exclude outliers, we use a threshold B_r for the maximum displacement of a data point caused by translation and scaling, i.e. $||P'| - |P|| \leq B_r$ for a correct correspondence pair P and P' . A pair of points is accepted as a correspondence pair only if the above condition is met. The threshold B_r can be selected as the k th largest distance among all the correspondence pairs, according to a predetermined constant fraction.

9.3.3 Experiments of Planar Shapes Registration

We apply the IDC algorithm to match a planar image shape to a curve model. The data points on the shape contour are extracted from the image using an edge detector. The model for the shape contour is a closed curve represented in piecewise splines. We derive the curve model from a clean set of contour pixels of the shape using an optimal spline fitting procedure (the spline fitting procedure is reported in our previous work [83]). The extraction of image shape and the preparation of curve model are similar to our experiments in Section 9.2.5 except that we do not need the orientation of the points here. The two shapes are initially registered by an approximate transformation which is either given or estimated using a global procedure.

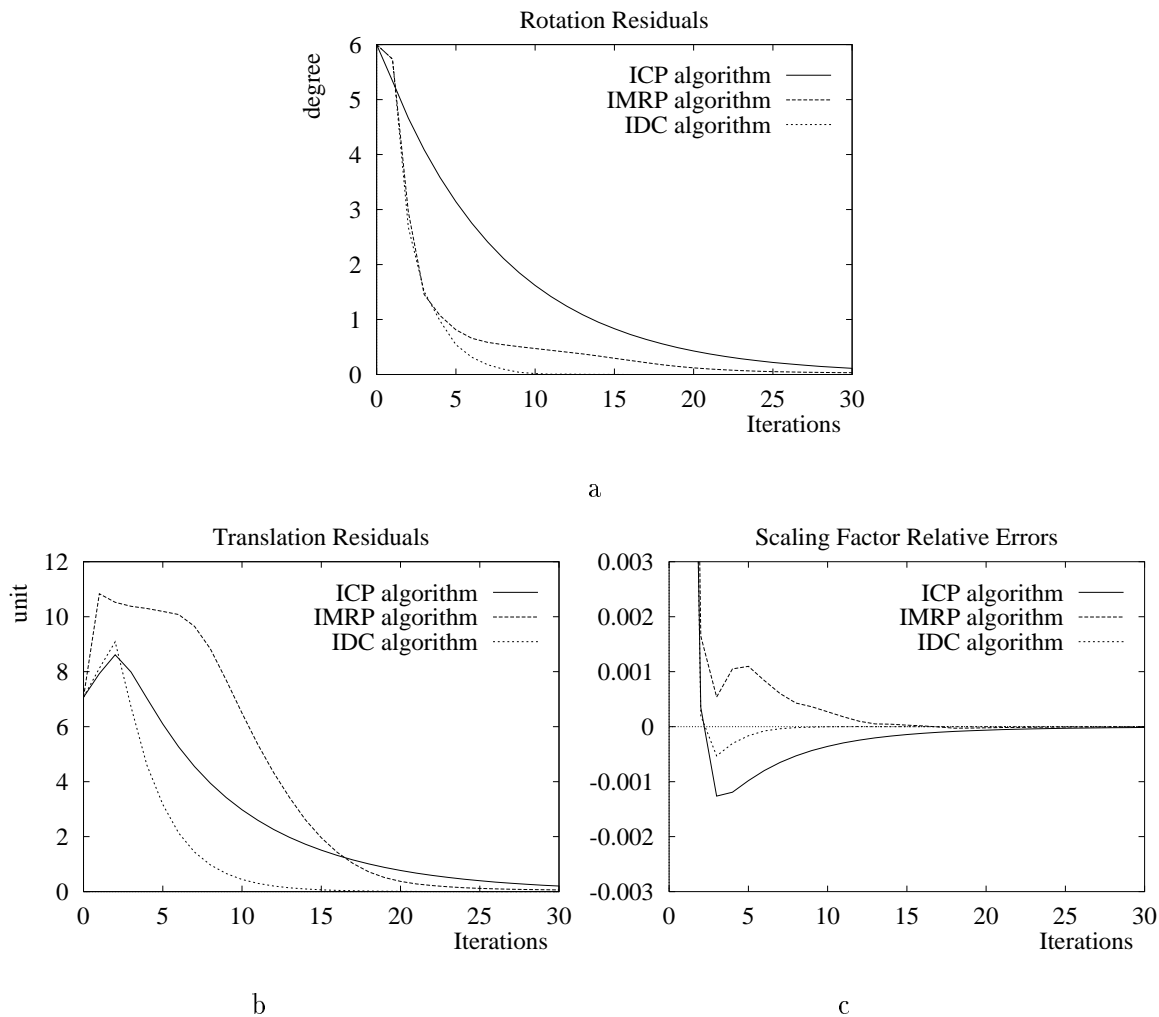


Figure 9.5: Comparison of iterative algorithms in an example. (a) Rotation residuals from the three algorithms over 30 iterations. The initial rotation angle is 6° . (b) Magnitudes of translation residuals. The magnitude of initial translation is 7.07 units. (c) Scaling factor residuals (relative errors of the derived scaling factors). The initial scaling factor is 0.8.

The correspondence between the data set and the model shape is computed as the following. For a data point P , its correspondence point P' according to either rule is determined by searching the curve model and minimizing a distance function $d(P')$, where $d(P') = |P' - P|^2$ in the case of closest-point rule and $d(P') = (|P'| - |P|)^2$ in the case of matching-range point rule. We apply a standard one-dimensional search procedure to minimize the distance function (by considering it as a function of the polar angle of P'). We then update the transformation using the least-squares solutions. This process is repeated as described in the IDC algorithm.

Two examples of registering planar shapes are given below. In the first example, a synthesized image containing a shape with an incomplete contour, is shown in Fig. 9.6(a). The extracted data points are shown in part (b). Part (c) shows an initial registration of the shape to a curve model, where noticeable pose error is present. Part (d) shows the result of registering the shape to the model. A threshold of 10% is used to remove outliers (i.e., 10% of the points which are furthest to model are discarded during the matching process). The recovered transformation variables are: rotation angle -9.2° , scaling factor 0.91, translation $(-5.25, -1.32)$ pixels.

In the second example, we register the shape of a maple leaf. The original image containing the leaf is shown in Fig. 9.7(a). The extracted points on the leaf contour are shown in part (b). Part (c) shows an initial registration of the leaf shape to a curve model. Part (d) shows the final result after registration. The recovered transformation variables are: rotation angle 9.0° , scaling factor 0.96, translation $(-12.5, 15.6)$ pixels. We applied the registration process twice with two different outlier thresholds. First, we used a low threshold (5%) to derive an approximate registration. Then we used a higher threshold (20%) to exclude more outliers and derive a more accurate final registration.

9.4 Discussion

We demonstrated that the two algorithms we initially introduced for matching range scans, the rotation search/least-squares algorithm and the iterative dual correspondence algorithm, can also be used for matching free-form curves obtained from intensity images. The adapted algorithms are able to solve for an scaling factor in addition to the rotation and translation in the transformation.

The rotation search/least-squares algorithm makes use of the tangent (or normal) directions in the shapes and thus it is more robust in associating correspondences and rejecting outliers. This algorithm also has a nice feature that it solves the rotation by a search procedure rather than fix-point iterations. Thus our algorithm would still work even if there is a large initial error in

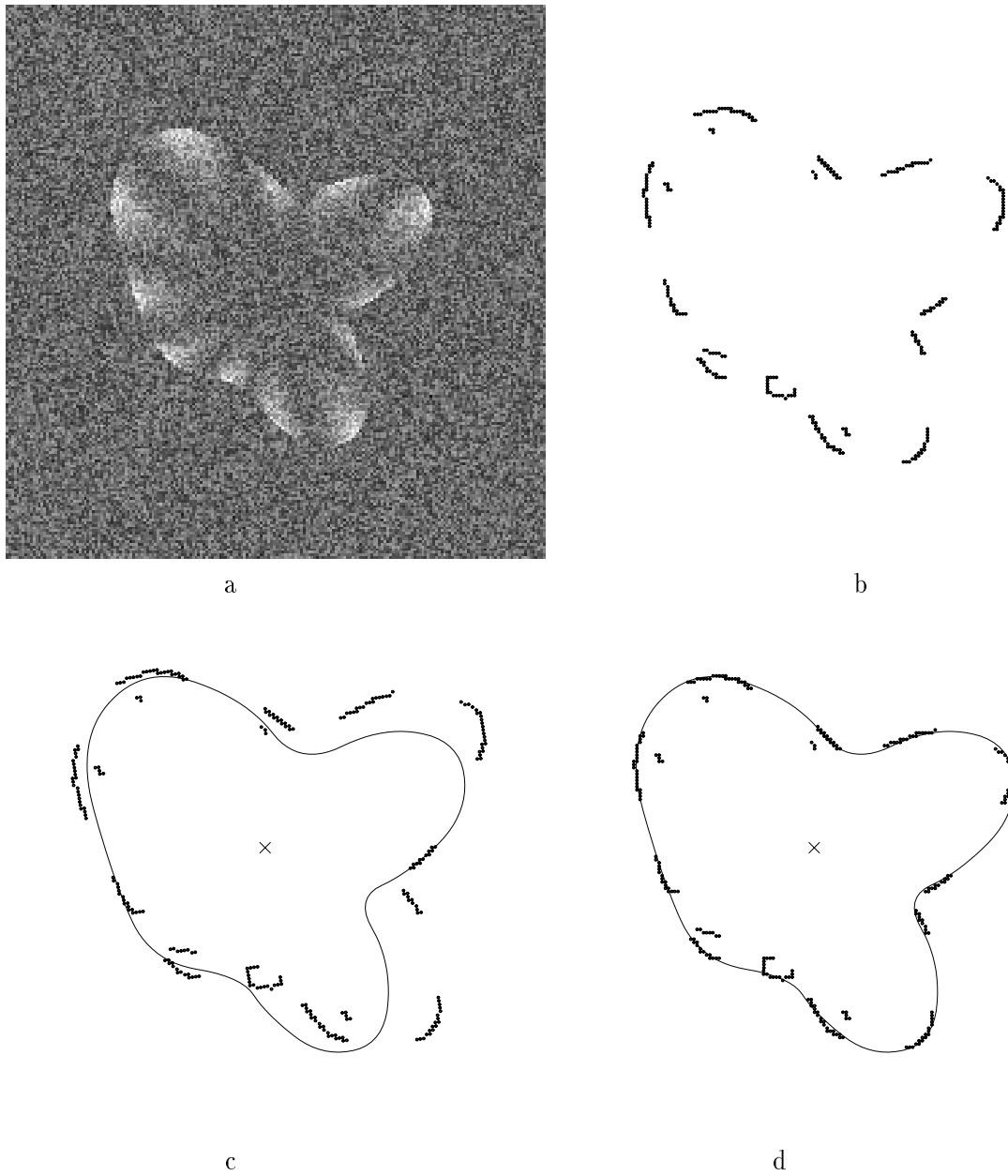


Figure 9.6: Registration of a synthesized shape. (a) Image containing the shape. (b) Extracted data points on the shape contour. (c) Initial registration of the shape to the curve model. (d) Final result of registration.

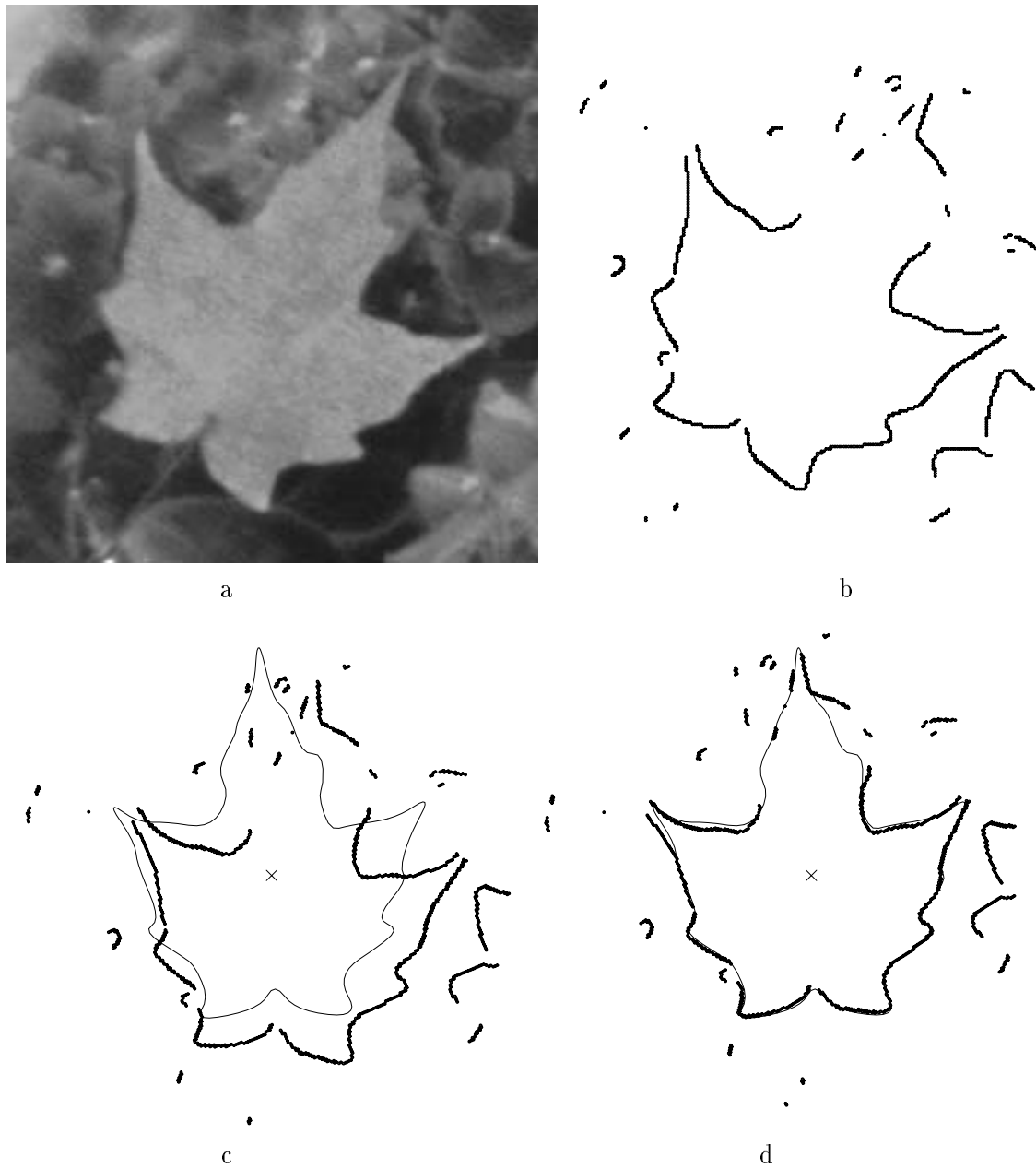


Figure 9.7: Registration of the shape of a maple leaf. (a) Image containing the leaf shape. (b) Extracted data points on the shape contour. (c) Initial registration of the leaf shape to the curve model. (d) Final result of registration.

rotation. A small disadvantage is that the search procedure we use only has a linear convergence rate, while an iterative least-squares method (such as the one described in Section 8.3.1) appears to have a superlinear rate. This is usually not a problem since the accuracy of our solution after a few iterations is already quite good considering that the data are noisy. If necessary, we can apply an iterative least-squares procedure after initially using our search algorithm.

The IDC algorithm based on point to point correspondences appears to be less robust because it does not use tangent information. But this algorithm is still useful if it is inconvenient to obtain the tangent information in the data shape or model. Comparing with the commonly used ICP (iterative closest point) algorithm, our IDC algorithm is more efficient. From the previous discussion about these algorithms in various applications, we can conclude that the IDC algorithm is essentially capable of replacing the ICP algorithm.

One advantage of our algorithms is that they can align curves without distinctive features. However, like any algorithm which is based on iterative optimization rather than feature correspondence, the two algorithms we discussed here all have the limitation that they require an initial estimation of the registration. If the amount of outliers and missing data is relatively small, the initial registration can be estimated from a global procedure. Otherwise, a good initial estimate must be given for these iterative algorithms to work.

Chapter 10

Conclusion

10.1 Summary of Major Results

The theme of the thesis has been shape registration using optimization techniques. We first focused on the shape matching problem for mobile robot applications, namely robot pose estimation and map building. Then we extended our discussion to the registration of 3D surfaces and planar image shapes.

The major contributions of this thesis are summarized as the following.

- We studied the problem of robot navigation in unknown structured environments using laser range sensing. Previously, laser range sensing has mostly been employed for robot localization with respect to a known world model. However, we examined the feasibility of dynamically building a contour model from range scans as well as using this data-made-model for robot pose estimation. Through our experiments, we concluded that using laser range sensing as well as odometry for navigating an unknown structured environment is a feasible approach.
- We comprehensively studied the techniques for matching a range scan with respect to another reference scan (which is the data-made-model) for deriving the relative position and heading of the robot. The difficult issues in this problem are that the scans are noisy, discontinuous, not necessarily linear, and two scans taken at different positions may not completely overlap because of occlusion. We examined several optimization-based scan matching algorithms including two new algorithms that we proposed as well as one that we adapted from an existing method. Extensive experiments with simulated and real data confirmed that these techniques effectively solve the 2D scan matching problem.

- In the study of optimal registration of multiple range scans for mapping an unknown environment, we raised the issue of maintaining spatial consistency in the integrated model. In most previous approaches to dynamic modeling, new sensor data are merged to a cumulative model only based on local registrations, and this may cause inconsistency in the model. We solved this problem by maintaining individual local frames of data as well as a network of spatial relations among data frames, and then optimally combining all available relations to resolve possible conflicts. The algorithm we propose to combine uncertain spatial relations is superior to a previous method by Smith and Cheeseman [107] in that we avoided a restriction on the type of connections of the spatial relations and our formulation is more elegant. We also derived and formulated uncertain pose constraints from two types of sensing data which are commonly used in mobile robot systems: odometry measurements and laser range measurements.
- We proposed two new algorithms for shape registration. These algorithms are based on iterative optimization and they do not require feature extraction or segmentation. The first algorithm is formulated as a search for minimizing a distance function. It is more robust in global optimality than a least-squares iterative algorithm. It is also more efficient than a full three degree-of-freedom search as our algorithm only searches in the nonlinear rotation dimension while it solves for translation in an embedded least-squares procedure. The second algorithm is based on iteratively minimizing a distance defined by point to point correspondences. It is more effective and efficient than the Iterative Closest Point algorithm [12]. Our algorithms are demonstrated to be very effective in matching 2D range scans. We also successfully extended our algorithms to other applications such as matching 3D range surfaces and planar image shapes.

10.2 Directions for Future Research

To conclude our thesis, we point out several possible directions for future research.

- We have proven that it is sufficient to use 2D laser range scans for robot localization in structured indoor environments where the world contour is typically formed by long smooth walls. However, it will be much more difficult to do this in a cluttered industrial workshop where the nice wall structures are hidden by irregular, three-dimensional objects such as pipes, racks, machines, etc. (One such environment is the AECL bay where the ARK robot is

tested [96].) It will be interesting to examine if laser range sensing is still capable of providing localization for the robot. A possible approach is to use 3D range scans (i.e. to allow both pan and tilt of the sensor) to reveal more structures in the environment. Note that this kind of 3D range scan is very different from dense, shallow-depth range images. Registration of 3D scans is considerably more difficult than registration of 2D scans because the range measurements are sparse in the 3D space. One possibility is to use an active approach in collecting the measurements (i.e. to point the laser beam in the directions where objects are observed from previous scans).

- For a mobile robot mapping an unknown environment, once all the scans of range data are registered into a common reference frame, the next task is to extract a concise representation from the data. Although it is an easy problem if the environment is simple, the best way of modeling a more complicated environment from sensor data is unclear, especially considering that there are sensing and registration errors in the data.
- We treat exploration and mapping as separate steps. In fact, the robot may determine where to go or look next based on the currently explored world. The best strategy for the robot depends on the task and the purpose of the exploration mission. The criteria can be to obtain a complete model, to minimize uncertainty in the model, to find an exit as quickly as possible etc.. The best strategy based on each criterion can be studied. Whaite and Ferrie have reported some results of autonomous exploration based on minimizing uncertainty [119].
- An objective function is formed and minimized in each step of an optimization-based matching algorithm. However, this objective function is only defined heuristically. We may study more carefully what the “best” way is in order to define the objective functions.
- It may be interesting to extend the optimization-based matching methods to handle slight deformation in the shape. For example, the surface of a human face may slightly change in different views. A possible approach is to divide the surface into smaller regions and allow slightly different transformations for each region. Some study on this problem based on curvature and motion consistency has been reported in [108].

Appendix A

Error from Approximate Point Correspondence

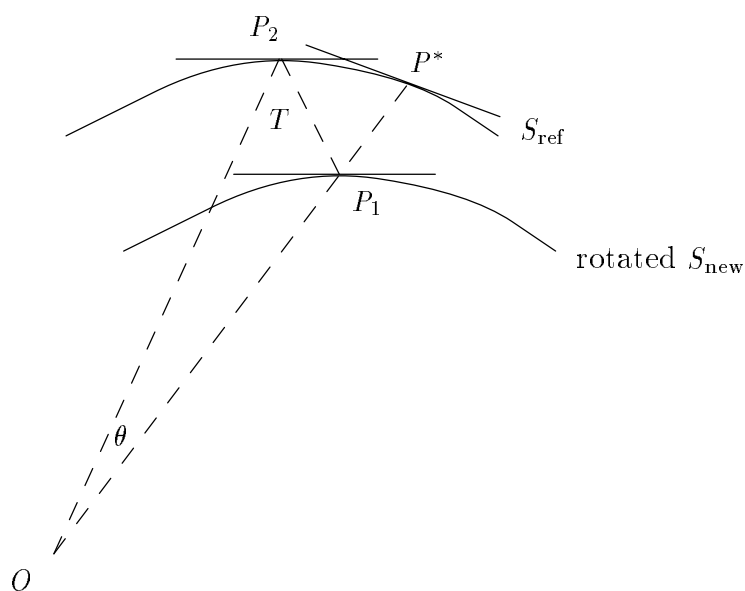


Figure A.1: Illustration of approximate point correspondence.

In Section 3.5, we choose an approximate correspondence P^* to replace P_2 as the corresponding point for P_1 (see Fig A.1). The pair P_1, P^* is used to form a constraint on the translation vector T as (assuming the rotation is already corrected out):

$$\vec{n}_1 \cdot T \approx \vec{n}_1 \cdot (P^* - P_1). \quad (\text{A.1})$$

The error in the above approximation is derived as the following. We regard that the reference scan S_{ref} is parameterized with the angle θ as the index parameter. Let $P_2 = (x(0), y(0))$, $P^* = (x(\theta), y(\theta))$. Then the approximation error is:

$$\begin{aligned}
& (P^* - P_1 - T) \cdot \vec{n}_1 \\
&= (P^* - P_2) \cdot \vec{n}_2 \\
&= (x(\theta) - x(0), y(\theta) - y(0)) \cdot (-y'(0), x'(0)) / \sqrt{x'(0)^2 + y'(0)^2} \\
&= (\theta x'(0) + O(\theta^2), \theta y'(0) + O(\theta^2)) \cdot (-y'(0), x'(0)) / \sqrt{x'(0)^2 + y'(0)^2} \\
&= O(\theta^2).
\end{aligned}$$

Notice that $\theta \leq |T|/|P_1|$. The above error is bounded by $O((|T|/|P_1|)^2)$, which is small when $|T| \ll |P_1|$.

Similarly, we can show that

$$\vec{n}^* \cdot T \approx \vec{n}^* \cdot (P^* - P_1) \quad (\text{A.2})$$

with an approximation error of $O((|T|/|P_1|)^2)$.

Furthermore, we can combine Eq. A.1 and Eq. A.2 and yield a better approximated constraint:

$$(\vec{n}_1 + \vec{n}^*) \cdot T \approx (\vec{n}_1 + \vec{n}^*) \cdot (P^* - P_1) \quad (\text{A.3})$$

whose approximation error can be similarly derived as

$$\begin{aligned}
& (P^* - P_2) \cdot (\vec{n}^* + \vec{n}_2) \\
&= O(\theta^3) = O((|T|/|P_1|)^3).
\end{aligned}$$

Note that in the above derivation, we assume that the curve (reference scan) is smooth.

Appendix B

Derivation of Identity

In this section, we derive the identity in Eq. 9.10 which we rewrite below:

$$\rho' = \phi' \rho \tan(\theta - \phi). \quad (\text{B.1})$$

Let $r(\theta)$ denote a smooth curve in a polar coordinate system. A tangent line $r = \rho / \cos(\theta - \phi)$ is defined at point $(\theta, r(\theta))$, where ϕ is the orientation of the normal and ρ is the distance from the origin. In the following discussion, all the derivatives are with respect to θ .

First from the fact that $r = \rho / \cos(\theta - \phi)$ is a tangent line to $r = r(\theta)$, both the function values of the curve and the line and their first derivatives should agree:

$$r(\theta) = \frac{\rho}{\cos(\theta - \phi)} \quad (\text{B.2})$$

$$r'(\theta) = \left(\frac{\rho}{\cos(\theta - \phi)} \right)' = \frac{\rho \sin(\theta - \phi)}{\cos^2(\theta - \phi)}. \quad (\text{B.3})$$

It follows that

$$\tan(\theta - \phi) = \frac{r'}{r}. \quad (\text{B.4})$$

The parameters ϕ and ρ can then be solved as:

$$\phi = \theta - \tan^{-1} \frac{r'}{r} \quad (\text{B.5})$$

$$\rho = r \cos(\theta - \phi) = \frac{r^2}{\sqrt{r^2 + r'^2}}. \quad (\text{B.6})$$

We now regard ϕ and ρ as functions of θ . Their derivatives can be evaluated directly from the above two equations:

$$\phi'(\theta) = 1 - \frac{\frac{r''}{r} - \frac{r'^2}{r^2}}{1 + \left(\frac{r'}{r}\right)^2} = \frac{r^2 + 2r'^2 - rr''}{r^2 + r'^2} \quad (\text{B.7})$$

$$\rho'(\theta) = \frac{2rr'}{\sqrt{r^2 + r'^2}} - \frac{r^2(2rr' + 2r'r'')}{2\sqrt{(r^2 + r'^2)^3}} = \frac{rr'(r^2 + 2r'^2 - rr'')}{\sqrt{(r^2 + r'^2)^3}}. \quad (\text{B.8})$$

Now from Equation B.4, B.6, B.7 and B.8 we can easily see that

$$\rho' = \phi' \rho \tan(\theta - \phi). \quad (\text{B.9})$$

This is the identity we want.

Bibliography

- [1] P. D. Alevizos, J. D. Boissonnat, and M. Yvinec. An optimal $O(n \log n)$ algorithm for contour reconstruction from rays. *ACM Symposium on Computational Geometry*, pages 162–170, 1987.
- [2] C. S. Andersen, C. B. Madsen, J. J. Sorensen, N. O. S. Kirkeby, J. P. Jones, and H. I. Christensen. Navigation using range images on a mobile robot. *Robotics and Autonomous Systems*, 10:147–160, 1992.
- [3] R. C. Arkin. Motor schema-based mobile robot navigation. *The International Journal of Robotics Research*, 8(4):92–112, 1989.
- [4] R. C. Arkin. Navigational path planning for a vision-based mobile robot. *Robotica*, 7, 1989.
- [5] M. Asada. Map building for a mobile robot from sensory data. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6):1326–1336, 1990.
- [6] N. Ayache and O. D. Faugeras. HYPER: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):44–54, 1986.
- [7] N. Ayache and O. D. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Transactions on Robotics and Automation*, 5(6):804–819, 1989.
- [8] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [9] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
- [10] R. Bauer and W. D. Rencken. Sonar feature based exploration. In *IEEE/RSJ International Conference of Intelligent Robots and Systems*, pages 148–153, 1995.
- [11] J. L. Bentley. Multidimensional binary search trees used for associative search. *Communications of the ACM*, 18(9):509–517, 1975.
- [12] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

- [13] M. J. Black. *Robust Incremental Optical Flow*. PhD thesis, Yale University, September 1992.
- [14] G. Blais and M. D. Levine. Registering multiple range data to create 3d computer objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):820–824, 1995.
- [15] J. Borenstein and L. Feng. Correction of systematic odometry errors in mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 569–574, 1995.
- [16] J. Borenstein and Y. Koren. Obstacle avoidance with ultrasonic sensors. *IEEE Journal of Robotics and Automation*, 4(2):213–218, 1988.
- [17] J. Borenstein and Y. Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1179–1187, 1989.
- [18] R. A. Brooks. Solving the find-path problem by good representation of free space. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(3):190–197, 1983.
- [19] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986.
- [20] R. A. Brooks and T. Lazano-Perez. A subdivision algorithm in configuration space for find-path with rotation. *IEEE Transactions on Systems, Man and Cybernetics*, 15:224–233, 1985.
- [21] J. Canny and B. Donald. Simplified voronoi diagrams. In I. J. Cox and G. T. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, 1990.
- [22] J. Canny and J. Reif. New lower bound techniques for robot motion planning problems. In *27th IEEE Symposium on Foundations of Computer Science*, pages 49–60, 1987.
- [23] T. A. Cass. Parallel computation in model-based recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 879–884, 1988.
- [24] G. Champlébourg, S. Lavalée, R. Szeliski, and L. Brunie. From accurate range imaging sensor calibration to accurate model-based 3D object localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 83–89, 1992.
- [25] R. Chatila and J. P. Laumond. Position referencing and consistent world modeling for mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 138–145, 1985.
- [26] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *International Journal of Image and Vision Computing*, 10(3):145–155, 1992.
- [27] J. H. Connell. *Minimalist Mobile Robotics: A Colony-style Architecture for an Artificial Creature*. Academic Press, 1990.
- [28] I. J. Cox. Blanche: Position estimation for an autonomous robot vehicle. In I. J. Cox and G. T. Wilfong, editors, *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

- [29] I. J. Cox. Blanche: An experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193–204, 1991.
- [30] I. J. Cox and G. T. Wilfong (eds.). *Autonomous Robot Vehicles*. Springer-Verlag, 1990.
- [31] I. J. Cox and J. B. Kruskal. On the congruence of noisy images to line segments models. In *Second International Conference on Computer Vision*, pages 252–258, 1988.
- [32] I. J. Cox and J. J. Leonard. Modeling a dynamic environment using a multiple hypothesis approach. *Journal of Artificial Intelligence*, 66(1):311–344, 1994.
- [33] J. L. Crowley. Navigation for an intelligent mobile robot. *IEEE Journal of Robotics and Automation*, 1(1):31–41, 1985.
- [34] J. L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *IEEE International Conference on Robotics and Automation*, pages 674–680, 1989.
- [35] R. de Almeida and C. Melin. Exploration of unknown environments by a mobile robot. In T. Kanade, F. C. A. Goren, and L. O. Hertzberger, editors, *Conference on Intelligent Autonomous System, 2*, pages 715–725, 1989.
- [36] C. Dorai, J. Weng, and A. K. Jain. Optimal registration of multiple range views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 569–571, 1994.
- [37] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation*, 7(6):859–865, 1991.
- [38] G. Dudek and C. Zhang. Pose estimation from image data without explicit object models. In *Vision Interface*, Banff, Alberta, 1994.
- [39] H. F. Durrant-Whyte. Consistent integration and propagation of disparate sensor observations. *International Journal of Robotics Research*, 6(3):3–24, 1987.
- [40] H. F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal of Robotics and Automation*, 4(1):23–31, 1988.
- [41] H. F. Durrant-Whyte and J. J. Leonard. Navigation by correlating geometric sensor data. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Tsukuba, Japan, 1989.
- [42] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, 1987.
- [43] O. D. Faugeras. *Three Dimensional Computer Vision*. MIT Press, Cambridge, MA, 1993.

- [44] C. Fennema, A. Hanson, E. Riseman, J. R. Beveridge, and R. Kumar. Model-directed mobile robot navigation. *IEEE Transaction on Systems, Man, and Cybernetics*, 20(6):1352–1369, 1990.
- [45] A. M. Flynn. Combining sonar and infrared sensors for mobile robot navigation. *The International Journal of Robotics Research*, 7(6):5–14, 1988.
- [46] K. Fujimura and H. Samet. Accessibility: A new approach to path planning among moving obstacles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 803–807, 1988.
- [47] K. Fujimura and H. Samet. A hierarchical strategy for path planning among moving obstacles. *IEEE Transactions on Robotics and Automation*, 5(1):61–69, 1989.
- [48] H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau. Registration of multiple range views for automatic 3-D model building. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 581–586, 1994.
- [49] A. Gelb. *Applied Optimal Estimation*. MIT Press, 1974.
- [50] J. Gonzalez, A. Reina, and A. Ollero. Map building for a mobile robot equipped with a 2D laser rangefinder. In *IEEE International Conference on Robotics and Automation*, pages 1904–1909, 1994.
- [51] W. E. L. Grimson, T. Lozano-Pérez, and D. P. Huttenlocher. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, Cambridge, MA, 1990.
- [52] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley and Sons, New York, NY, 1986.
- [53] M. Hebert, I. Kweon, and T. Kanade. 3D vision techniques for autonomous vehicles. In C. E. Thorpe, editor, *Vision and Navigation: The Carnegie Mellon Navlab*, Boston, Mass., 1990. Kluwer Academic Publishers.
- [54] K. Higuchi, M. Hebert, and K. Ikeuchi. Building 3-D models from unregistered range images. In *IEEE International Conference on Robotics and Automation*, pages 2248–2253, 1994.
- [55] G. E. Hinton, C. K. I. Williams, and M. D. Revow. Adaptive elastic models for hand-printed character recognition. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, San Mateo, California, 1992. Morgan Kauffmann.
- [56] B. K. Horn. *Robot Vision*. MIT Press, Cambridge, Mass., 1986.
- [57] D. P. Huttenlocher, G. A. Klanderma, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

- [58] D. P. Huttenlocher and S. Ullman. Object recognition using alignment. In *First International Conference on Computer Vision*, pages 102–111, 1987.
- [59] S. Iyengar and A. Elfes. *Autonomous Mobile Robots, Vols. 1 and 2*. IEEE Computer Society Press, Los Alamitos, CA, 1991.
- [60] D. Kahaner, C. Moler, and S. Nash. *Numerical Methods and Software*. Prentice Hall, Englewood Cliffs, NJ, 1989.
- [61] A. Kalvin, E. Sconberg, J.T. Schwartz, , and M. Sharir. Two-dimensional, model-based, boundary matching using footprints. *International Journal of Robotics Research*, 5(4):38–55, 1986.
- [62] S. Kambhampati and L. S. Davis. Multiresolution path planning for mobile robots. *IEEE Journal of Robotics and Automation*, 2(3):135–145, 1986.
- [63] B. Kamgar-Parsi, J. L. Jones, and A. Rosenfeld. Registration of multiple overlapping range images: Scenes without distinctive features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):857–871, 1991.
- [64] T. Kanade, editor. *Three-Dimensional Machine Vision*. Kluwer Academic Publishers, Boston, Mass., 1987.
- [65] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, Oxford, 1993.
- [66] K. Kant and S. W. Zucker. Toward efficient trajectory planning: The path-velocity decomposition. *The International Journal of Robotics Research*, 5(3):72–89, 1986.
- [67] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1):90–98, 1986.
- [68] V. Koivunen, J. Vezien, and R. Bajcsy. Multiple representation approach to geometric model construction from range data. Technical Report MS-CIS-93-66, GRASP Lab., University of Pennsylvania, 1993.
- [69] A. Kosaka and A. C. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *CVGIP: Image Understanding*, 56(3):271–329, 1992.
- [70] D. J. Kriegman, E. Triendl, and T. O. Binford. Stereo vision and navigation in buildings for mobile robots. *IEEE Transactions on Robotics and Automation*, 5(6):792–803, 1989.
- [71] B. Kuipers and Y. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63, 1991.

- [72] B. J. Kuipers and T. S. Levitt. Navigation and mapping in large-scale space. *AI Magazine*, pages 25–43, Summer 1988.
- [73] Y. Lamdan and H. J. Wolfson. Geometric hashing : A general and efficient model-based recognition scheme. In *IEEE Second International Conference on Computer Vision*, pages 238–249, 1988.
- [74] S. Y. T. Lang and A. K. C. Wong. Building geometric world model with graph synthesis for sensor fusion in mobile robots. *Computational Intelligence*, 6:91–107, 1990.
- [75] J-C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, Mass., 1991.
- [76] S. Lavallee and P. Cinquin. IGOR: Image guided operating robot. In *Fifth International Conference on Advanced Robotics*, Pisa, Italy, 1991.
- [77] J. Leonard, H. Durrant-Whyte, and I. J. Cox. Dynamic map building for an autonomous mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1990.
- [78] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transaction on Robotics and Automation*, 7(3):376–382, 1991.
- [79] T.S. Levitt and D. T. Lawton. Qualitative navigation for mobile robots. *Artificial Intelligence*, 44:305–360, 1990.
- [80] T. Lozano-Perez. Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, 32(2):108–119, 1983.
- [81] T. Lozano-Perez and M. A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22:560–570, 1979.
- [82] F. Lu and E. Milios. An iterative algorithm for shape registration. In *Second International Workshop on Visual Form*, Naples, Italy, 1994.
- [83] F. Lu and E. Milios. Optimal spline fitting to planar shape. *Signal Processing*, 37(1), 1994.
- [84] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–938, Seattle, 1994.
- [85] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. Technical Report RBCV-TR-94-46, University of Toronto, 1994.
- [86] F. Lu and E. Milios. Optimal global pose estimation for consistent sensor data registration. In *IEEE International Conference on Robotics and Automation*, pages 93–100, Nagoya, Japan, 1995.

- [87] V. Lumelsky. A comparative study of the path length performance of maze-searching and robot motion algorithms. *IEEE Transactions on Robotics and Automation*, 7(1), 1991.
- [88] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun. Dynamic path planning in sensor-based terrain acquisition. *IEEE Transactions on Robotics and Automation*, 6(4):462–472, 1990.
- [89] T. Masuda and N. Yokoya. A robust method for registration and segmentation of multiple range images. In *Second IEEE CAD-based Vision Workshop*, pages 106–113, 1994.
- [90] P. S. Maybeck. The kalman filter: An introduction to concepts. *Stochastic Models, Estimation and Control*, 1:3–16, 1979.
- [91] Merriam-Webster. *Webster's Ninth's New Collegiate Dictionary*. Merriam-Webster, Springfield, Mass., 1985.
- [92] E. E. Milios. Shape matching using curvature processes. *Computer Vision, Graphics, and Image Processing*, 47:203–236, 1989.
- [93] F. Mokhtarian and A. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):34–43, 1986.
- [94] H. P. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *IEEE International Conference on Robotics and Automation*, pages 116–121, 1985.
- [95] V. Morellas, J. Minners, and M. Donath. Implementation of real time spatial mapping in robotic systems through self-organizing neural networks. In *IEEE/RSJ Conference on Intelligent Robots and Systems*, pages 277–284, 1995.
- [96] S. B. Nickerson, M. Jenkin, E. Milios, B. Down, P. Jasiobedzki, A. Jepson, D. Terzopoulos, J. Tsotsos, D. Wilkes, N. Bains, and K. Tran. Design of ARK, a sensor-based mobile robot for industrial environments. In *INTELLIGENT VEHICLES*, Tokyo, Japan, 1993.
- [97] B. J. Oommen, S. S. Iyengar, N. S. V. Rao, and R. L. Kashyap. Robot navigation in unknown terrains using learned visibility graph. part i: The disjoint convex obstacle case. *IEEE Journal of Robotics and Automation*, 3(6):672–681, 1987.
- [98] T. Pavlidis. Algorithms for shape analysis of contours and waveforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4):301–312, 1980.
- [99] M. A. Penna and J. Wu. Models for map building and navigation. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(5):1276–1301, 1993.
- [100] J. Reif and M. Sharir. Motion planning in the presence of moving obstacles. In *25th IEEE Symposium on Foundations of Computer Science*, pages 144–154, 1985.

- [101] M. Robinson and M. Jenkin. Reactive low level control for the ARK. In *Proceeding of Vision Interface '94*, pages 41–47, 1994.
- [102] Y. Roth-Tabak and R. Jain. Building an environment model using depth information. *IEEE Computer*, 6:85–90, 1989.
- [103] M. Rutishauser, M. Stricker, and M. Trobina. Merging range images of arbitrarily shaped objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 573–580, Seattle, 1994.
- [104] J. T. Schwartz and M. Sharir. On the piano movers' problem: II. general techniques for computing topological properties of real algebraic manifolds. In J. Schwartz, M. Sharir, and J. Hopcroft, editors, *Planning, Geometry, and Complexity of Robot Motion*, Norwood, N.J., 1987. Ablex Publishing Corp.
- [105] J.T. Schwartz and M. Sharir. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *International Journal of Robotics Research*, 6(2):29–44, 1987.
- [106] D. A. Simon, M. Hebert, and T. Kanade. Real-time 3-D pose estimation using a high-speed range sensor. In *IEEE International Conference on Robotics and Automation*, 1994.
- [107] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.
- [108] G. Soucy and F. P. Ferrie. Motion and surface recovery using curvature and motion consistency. Technical report, Center for Intelligent Machines, McGill University, 1993.
- [109] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):344–358, 1995.
- [110] B. Steer. Trajectory planning for a mobile robot. *The International Journal of Robotics Research*, 8(5):3–14, 1989.
- [111] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics, and Image Processing*, 40(3):361–387, 1987.
- [112] G. Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 1993.
- [113] O. Takahashi and R. J. Schilling. Motion planning in a plane using generalized voronoi diagrams. *IEEE Transactions on Robotics and Automation*, 5(2):143–150, 1989.
- [114] H. Takeda and J-C. Latombe. Sensory uncertainty field for mobile robot navigation. In *IEEE International Conference on Robotics and Automation*, pages 2465–2472, 1992.

- [115] Y. C. Tang and C. S. G. Lee. A geometric feature relation graph formulation for consistent sensor fusion. *IEEE Transactions on System, Man, and Cybernetics*, 22(1):115–129, 1992.
- [116] J. L. Turney, T. N. Mudge, and R. A. Volz. Recognizing partially occluded parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):410–421, 1985.
- [117] P. J. van Otterloo. *A Contour-Oriented Approach to Shape Analysis*. Prentice Hall, New York, 1992.
- [118] C. M. Wang. Location estimation and uncertainty analysis for mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 1230–1235, 1988.
- [119] P. Whaite and F. P. Ferrie. Autonomous exploration: Driven by uncertainty. Technical Report TR-CIM-93-17, Center for Intelligent Machines, McGill University, 1993.
- [120] S. Xie. Incremental construction of 3-d models of a scene from sequentially planned views. Technical Report CMPT TR 87-1, Department of Computer Science, Simon Fraser University, 1987.