Lehrstuhl für Realzeit-Computersysteme

Localization in Indoor Environments Using a Panoramic Laser Range Finder

Tobias Einsele

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. rer. nat. M. Lang
Prüfer der Dissertation:	1.	Univ.-Prof. Dr.-Ing. G. Färber
	2.	apl. Prof. Dr.-Ing., Dr.-Ing. habil. G. Ruske

Die Dissertation wurde am 19.09.2001 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 10.01.2002 angenommen.

Abstract

This work presents a self-localization system suited for robots roaming in everyday indoor environments such as offices or residential buildings. Thus, it has to cope with environments that are typically not a priori modelled or in any way specially prepared, but are places where people move around and objects change position from time to time. The proposed localization system utilizes a panoramic laser range finder with a viewing angle of nearly 360° that provides planar range scans of the immediate surroundings in a fixed height parallel to the floor. The sensory input is preprocessed by extracting line segment features, which indicate the walls or other vertical boundaries in the environment. The following step is a scan matching procedure: Two preprocessed scans, which are in general taken from different positions in the surroundings, are shifted and rotated against each other in such a way that they optimally coincide. This is accomplished by employing a modified version of the so-called DP-Algorithm. This algorithm is wellknown in the pattern recognition domain and provides the optimal assignment of line segments solely on basis of the two scans without relying on any additional source of information or a priori knowledge of the environment. Based on this assignment of line segments, localization is realized by setting up and solving an overdetermined equation system. A further result from this is an estimate of the respective position uncertainty, which is to be propagated through the successive localization steps. The position estimates together with the sensory observations can be combined to build a map of the environment. In this context the robust recognition of known places as performed by the DP-Algorithm based scan matching is utilized to reduce accumulated uncertainty. The constructed environmental map is an attributed graph, which allows for different degrees of refinement and the use of topological relations side by side with geometrical positions. The approach has been implemented on a mobile robot and tested in a real world office environment under varying conditions.

Contents

List of Figures

Glossary

N	Number of samples in a panoramic scan.
S	Set of scan points forming a panoramic scan.
s_u	Scan point produced by the panoramic laser range finder (PLRF).
$[r_u, \varrho_u]^T$	Specification of a scan point s_u in polar coordinates with r_u being the range value and ϱ_u the angle value.
n	Number of extracted line segments in a panoramic scan.
L	Set of extracted line segments in a panoramic scan.
ℓ_i	Extracted line segment.
$[a_i, \alpha_i]^T$	Hesse notation for the underlying straight line of line segment ℓ_i with a_i being the distance value and α_i being the angle value.
ϱ_{si}	Angle value that defines the start point of line segment ℓ_i on the underlying straight line.
ϱ_{ei}	Angle value that defines the end point of line segment ℓ_i on the underlying straight line.
$d_{\perp,u}$	Perpendicular distance of a scan point s_u to the current line.
$d_{\perp,\max}$	Maximum perpendicular distance of a scan point s_u to the current line.
$d_{\perp,\text{th}}$	Maximum allowed perpendicular distance of a scan point s_u to the current line.
$d_{u(u+1)}$	Euclidian distance between two scan points s_u and $s_{(u+1)}$.
$d_{\parallel,u(u+1)}$	Distance between two scan points s_u and $s_{(u+1)}$ parallel to the current line.
$[x_u, y_u]^T$	Cartesian specification of a scan point s_u.
M	Number of scan points s_u for which a regression line is to be found.

$e_{\mathrm{TLS},u}$	Perpendicular distance of a scan point s_u to its regression line.
E_{TLS}	Accumulated perpendicular distance.
$\boldsymbol{C}$	Variance-covariance matrix.
λ_1, λ_2	Eigenvalues in 2D with $\lambda_1 > \lambda_2$.
$\boldsymbol{v}_1, \boldsymbol{v}_2$	Orthogonal eigenvectors in 2D with $\boldsymbol{v}_1$ belonging to λ_1 and $\boldsymbol{v}_2$ belonging to λ_2.
$s_{xx} = {s_x}^2$	
$s_{yy} = {s_y}^2$	Variances in 2D.
$s_{xy} = s_{yx}$	Covariances in 2D.
$\boldsymbol{m} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}$	Mean value in 2D.
C	Set of extracted corners in a panoramic scan.
$c_{i(i+1)}$	Corner extracted on basis of the line segments ℓ_i and $\ell_{(i+1)}$.
$\Delta\alpha_{\mathrm{th}}$	Successive line segments ℓ_i and $\ell_{(i+1)}$ are intersected and a corner is established, if the absolute value of the angle difference, i. e. $\lvert\alpha_{(i+1)} - \alpha_i\rvert$ exceeds this minimum threshold.
$\begin{bmatrix} r_{i(i+1)} \\ \varrho_{i(i+1)} \end{bmatrix}$	Specification of the intersection point of the line segments ℓ_i and $\ell_{(i+1)}$ in polar coordinates.
L_{test}	Test scan consisting of m line segments with indices i, k and o.
L_{ref}	Reference scan consisting of n line segments with indices j, l and p.
(i, j)	Initially chosen pair of line segments ℓ_i, ℓ_j. Needed to bind the rotatory and one translatory degree of freedom.
P_{ij}	Set of tuples identifying the pairs of line segments that are admitted for matching, given that line segments ℓ_i, ℓ_j are aligned.
(k, l)	Pair of line segments chosen from P_{ij}. Needed to bind the remaining translatory degree of freedom.
$\Delta\alpha_{ij}$	Angle difference between line segments ℓ_i, ℓ_j *before* shift and turn.
$\Delta\alpha_{kl}, \Delta\alpha_{op}$	Angle differences between line segments ℓ_k, ℓ_l and ℓ_o, ℓ_p, respectively, *after* shift and turn.
Δa_{op}	Distance differences between line segments ℓ_o, ℓ_p *after* shift and turn.
$q_{ijkl}\,(o, p)$	Pairwise similarity measure between line segments ℓ_o, ℓ_p after having bound all three degrees of freedom by "locking" line segment pairs ℓ_i, ℓ_j and ℓ_k, ℓ_l.

$P_{\text{opt},ijkl}$	Given tuple (k, l), locally optimal subset of P_{ij}. Also denoted as locally optimal path.
$Q_{\text{max},ijkl}$	Locally optimal similarity measure accumulated along the path $P_{\text{opt},ijkl}$.
P_{opt}	Globally optimal path.
Q_{max}	Globally optimal similarity measure accumulated along the path P_{opt}.
$\boldsymbol{C}_o, \boldsymbol{C}_p$	Variance-covariance matrix belonging to the scan points of ℓ_o, ℓ_p, respectively.
ℓ_{op}	Overall distribution consisting of the union set of the scan points of ℓ_o and ℓ_p. Also denoted as virtual line segment.
$\boldsymbol{m}_{op}$	Mean vector of the overall distribution.
$\boldsymbol{C}^*_{op}$	Weighted mean variance-covariance matrix of $\boldsymbol{C}_o$ and $\boldsymbol{C}_p$.
$\boldsymbol{C}_{op}$	Variance-covariance matrix of the overall distribution.
g, g_{op}	Underlying straight line of the virtual line segmenet ℓ_{op}.
${s_{g\perp,o}}^2$, ${s_{g\perp,p}}^2$	Variances of ℓ_o, ℓ_p perpendicular to the straight line g.
${s_{g\parallel,o}}^2$, ${s_{g\parallel,p}}^2$	Variances of ℓ_o, ℓ_p parallel to the straight line g.
$\boldsymbol{V}^*_{op}$	Matrix of eigenvectors of $\boldsymbol{C}^*_{op}$.
b_{go}, b_{gp}	Distances between g and $\boldsymbol{m}_o$, $\boldsymbol{m}_p$, respectively.
ξ_{kl}	Optimal displacement of the second shift operation between L_{test} and L_{ref}.
$\mathcal{L}_{\text{ref}}$	Set of reference scans.
T	Number of reference scans in $\mathcal{L}_{\text{ref}}$.
$d(o, p)$	Local distance measure.
D_{min}	Minimum accumulated distance measure.
$D(o, p)$	Accumulated distance measure.
D_{end}	Accumulated distance measure at the end point of the DP forward search.
$Q_{ijkl}(o, p)$	Accumulated similarity measure.
$Q_{\text{end},ijkl}$	Accumulated similarity measure at the end of the DP forward search.
R_1	Lower left region of the search matrix.
R_2	Upper right region of the search matrix.

$L_{\text{ref,best}}$	Reference scan from $\mathcal{L}_{\text{ref}}$ that is the most similar to the current test scan L_{test}.
$Q_{\max,t}$	Similarity measure calculated between L_{test} and $L_{\text{ref,best}}$.
$\Delta\boldsymbol{X} = \begin{bmatrix}\Delta X \\ \Delta Y \\ \Delta\Phi\end{bmatrix}$	Displacement vector between two positions from where scans have been shot.
$\alpha_{g_{op}}$	Angle value of the straight line g_{op}, which is supposed to represent the actual face of the originating object.
b_{op}	Distance component between $\boldsymbol{m}_o$ and $\boldsymbol{m}_p$ perpendicular to g_{op}.
$s^2_{\Delta\Phi}$	Variance of $\Delta\Phi$.
$\boldsymbol{A}$	System matrix.
$\Delta\boldsymbol{x} = \begin{bmatrix}\Delta X \\ \Delta Y\end{bmatrix}$	Solution vector. In the present case the translatory components of the displacement vector $\Delta\boldsymbol{X}$.
$\boldsymbol{b}$	Right hand side vector. In the present case the coefficients b_{op}.
$\boldsymbol{r}$	Residual vector.
$\boldsymbol{I}$	Identity matrix.
$\boldsymbol{W}$	Diagonal weighting matrix.
w_κ	Weighting coefficients.
$\boldsymbol{A}_W$	Weighted system matrix.
$\boldsymbol{b}_W$	Weighted right hand side vector.
$\boldsymbol{C}_{\Delta\boldsymbol{x}}$	Variance-covariance matrix of the translatory components of the displacement vector.
ΔX_2	Exactly estimable translatory displacement component in case of partial localization.
$\boldsymbol{X} = \begin{bmatrix}X \\ Y \\ \Phi\end{bmatrix}$	Position in the global coordinate system.
$\boldsymbol{C}_{\boldsymbol{X}}$	Variance-covariance matrix of the position in the global coordinate system.
$\boldsymbol{J} = \begin{bmatrix}\boldsymbol{J}_1 \vert \boldsymbol{J}_2\end{bmatrix}$	Jacobian matrix with submatrices.
$\boldsymbol{C}_{\Delta\boldsymbol{X}}$	Variance-covariance matrix of the displacement vector.
V	Set of nodes in the graph based map.

Chapter 1

Introduction

During the last decades it could be observed that robots increasingly and successfully spread in almost all fields of manufacturing and production scenarios in order to assist or even completely replace the human operator in difficult, tedious or dangerous jobs. Popular examples range from small scale applications such as wire bonding in chip manufacturing or mounting the mechanical parts of a wristwatch movement up to large scale tasks like welding or lacquering a car body or transportation of parts in a plant by automated guided vehicles.

What all contemporary robots employed for applications as mentioned above have in common is that on the one hand they perform fast and accurately, but on the other hand heavily rely on strictly defined operating conditions. This latter characteristic is due to the fact that the robots' sensory skills are still that sparsely developed that an actual perception of the environment is limited. As a consequence, a typical nowadays robots' environment is often specifically designed and heavily prepared, e. g. with magnetic induction loops, beacons or markers on the flooring or walls, or at least exactly (CAD-) modelled. The robots' task is to be precisely specified and the robot has to be highly trained for the particular task. This requires a lot of human intervention and engineering involving considerable effort, which has to be spent over again if there is any change in the environment, the task or the robot. A further consequence is that unforeseen events, e. g. a car body that is offered in a different pose or people crossing the workspace, can only be coped with by simply stopping the robot's operation.

In manufacturing scenarios these shortcomings can be tolerated as preparing the environment is an actual option, CAD models are mostly available and

people can be banned from the respective areas. However, if a robot is to roam in an office or a home environment, e. g. in order to fulfill delivery or surveillance tasks, it is crucial not to rely on a priori models or environmental modifications as they are normally not at hand or cannot be installed, respectively. Furthermore, it is essential that the mobile robot is able to get along with people walking around in its workspace.

An immediate problem in this context is that on the one hand an environmental model is a prerequisite for the robot's task-oriented behaviour, on the other hand the model is not a priori given. So the sensory skills of a mobile robot employed in an office or home environment must be improved in such a way as to be capable of exploring, i. e. mapping the environment on its own. This means that during an exploration phase information gathered by the robot's sensory system has to be stored and arranged in a manner that the robot is able to find its way, i. e. to navigate in its environment thereafter.

A key feature in order to accomplish both, mapping and navigation, is the ability of self-localization: In the beginning of the exploration phase the mobile robot is standing at some place in the unknown environment. The sensory system wakes up and supplies an observation taken from the current position. This initial observation represents the nucleus of the later environmental model. In the following the robot begins to roam and the sensory system successively provides observations of the surroundings. Self-localization now renders the correct relation of the single sensory observations to each other and along that way an environmental map can gradually be constructed. When the map has been built, self-localization can be used for navigation, as it indicates the robot's current whereabouts within the given map.

Naturally, the main requirement on the self-localization system is to supply exact positional data. However, as in the present case self-localization is to assist during the map building phase, additional qualifications are needed: Firstly, the self-localization system has to be operable even if no a priori information about the environment is at hand. Furthermore, in order to obtain a consistent environmental model the localization system should be able to give a cue whether the current location has already been visited and mapped. While this feature is not that important as long as small environments are considered, it becomes essential when large cyclic environments are to be mapped where positional data may suffer from error accumulation.

In this work a localization system is presented, which uses a panoramic laser range finder providing planar scans of the surroundings as sensory input. The approach not only meets the above mentioned criteria, but can also cope with

the typical boundary conditions in indoor environments, i. e. no preparation of the environment allowed and people moving in the workspace. Beyond that, it can operate without any additional input from other sensors.

The presentation is organized as follows: Chapter 2 is a literature survey, which is aimed at giving an overview of current localization and map building techniques. Chapter 3 introduces the employed laser range finder and gives some details about its performance and measurement accuracy. Although laser range finders are state of the art sensors, this chapter may be interesting as not the commonly used 180° scanner, but a model featuring a viewing angle of nearly 360° has been chosen. Chapter 4 presents and discusses alternatives to preprocess the laser scanner's raw data such that the actual information is preserved and noise and clutter are reduced. In this context the two different methods filtering and feature extraction are addressed. After giving a detailed problem description chapter 5 reveals a method how laser scans taken from different positions in the surroundings can accurately and robustly be matched. The employed method utilizes a modified version of an algorithm that is wellknown in the pattern matching and recognition domain. Using the results from matching the actual localization problem is tackled in chapter 6. The principal localization procedure is developed and in the following expanded to the concept of position uncertainty. The second part of this chapter focuses on how localization results and sensory readings can be integrated into an effective environmental map. The resulting map structure combines geometrical and topological aspects. Chapter 7 presents several localization experiments conducted in a real-world office environment. During the experiments the laser scanner is the only sensory input. So the pure performance of the developed localization system is presented in different locations and under varying conditions. Chapter 8 gives a summary of what has been accomplished in this work and points out in which respects the present localization system can be improved.

Chapter 2

Related and Supplementary Work

In the present chapter a literature survey of approaches that deal with the robot localization problem in indoor environments will be given. Before detailing on the different techniques that have been published in this context, it is important to point out that the robot localization task cannot be viewed independently: Localization provides a positional fix of the robot in its environment. However, this information is only actually useful if a reference frame is given, i. e. if the positional fix is specified with respect to a geometrical or topological map of the surroundings. Usually, such an environmental map is not a priori given (cf. section 6.1). This is why the problem actually to be solved is referred to as concurrent localization and map building [Ren93] or simultaneous map building and localization [CTS97, HH01].

Whenever the robot localization task comes up there always exists one very simple and straightforward approach: Utilizing the robot's odometry as positional sensor, i. e. evaluating the angular encoders attached to the vehicle's wheels (cf. section 7.1). Excelling by its simplicity and availability, odometry suffers from two major sources of error: Firstly, the geometry of the vehicle's chassis and wheels is not exactly known. This is referred to as the systematic odometry error. Secondly, there is an inevitable wheel slippage, which results in a non-systematic error component. Both kinds of errors can be reduced: The former by calibrating the odometric system [BF96], the latter by e. g. using an encoder trailer as suggested in [Bor94]. However, as odometry is an internal sensor that lacks the possibility to correct potential inaccuracies by environment perception, positional error is accumulated and increases without bounds. This means that odometry performs well if short-

term or short-range localization is to be accomplished. Whenever long-term or long-range localization is the task, odometry has to be at least supported, if not even thoroughly replaced by a sensor system that actually perceives its environment and is thus capable of correcting for positional error.

In the following several successful localization and map building strategies found in literature will be outlined. Mostly, the presented techniques are primarily based on range sensors, i. e. laser range finders or ultrasonic transceivers. However, they sometimes also extend to image sensors such as monocular or stereo vision systems using CCD cameras.

2.1 Kalman Filter Approach

To tackle the localization and map building problem, an approach pursued by many researchers is to employ a Kalman filter. By its nature the Kalman filter is an optimal state estimator based on a Gaussian error model. The robot's position as well as the positions of objects gained from sensory readings are contained in a state vector. While the robot is roaming around gathering sensory input, the state vector is successively updated, i. e. a new estimate of the robot's and the objects' positions is calculated. The applied optimization criterion for the state vector update is the minimization of the (co-)variances. The aimed objective of the Kalman filter approach is to generate a geometrically exact, global environmental map, i. e. the robot's and the objects' positions are referenced with respect to a global coordinate system.

This procedure has several implications: Due to the Gaussian assumption, the approach can only correctly handle data distributions that are adequately represented by their first and second order moment, i. e. mean and variance. However, in a real-world environment there are two common cases where this requirement is not fulfilled: Firstly, outlying or spurious sensory data that do not agree with the Gaussian model are inevitable. Secondly, and even more important for practical application is that a typical indoor environment is characterized by its dynamics due to people trespassing the viewing angle of the sensor. Any moving object, however, severely violates the Gaussian assumption: For example, supposing the robot is standing still, successive range measurements of a moving object would certainly reveal a data distribution that is far off a Gaussian distribution. This is why it has to be ensured that a Kalman filter is only provided with Gaussian data originating from a (quasi-)static environment [LDWC90].

In order to realize the optimization criterion of the Kalman filter, it is necessary to establish correspondences between sensory readings taken from different positions. In the case of a range sensor this means that raw range readings [Cox91, WvP95, LM94, LM97, Gut99], clusters of range readings (ellipses) [KL97] or extracted geometric features such as line segments [GOR94, LFW94, KL97, AS97], corners [VTGA94, LDW91] or circles [LDW91] have to be related to each other. This is a highly critical task, as an incorrectly established correspondence will significantly distort the result of a state vector update operation.

In [LDWC90, LDW91] the geometric features line segments, corners and circles are tracked to provide the Kalman filter with the respective input using the control input to the robot vehicle as initial guess for the displacement between two successive sensory readings. In [Cox91, LM94, LM97, Gut99, GOR94, LFW94, KL97] odometric data give the clue how raw range readings, clusters of range readings or line segments may possibly mate, respectively. As detailed above, odometry and likewise the control input to the robot vehicle only provide an accurate position estimate if covered ranges are short. This is why, in order to avoid an incorrect correspondence between sensory readings, the above mentioned state vector update cycles will only cover short ranges, either. So, Kalman filtering can be characterized as an incremental approach, which has in turn, two consequences: If there is a serious failure, e. g. due to an incorrect correspondence, which results in a completely wrong position estimate, there is no chance of recovery, i. e. the wrong position will be maintained [GWN99]. Secondly, Kalman filtering is not suited for an initial localization of a robot in its environment, as in such a case an initial guess of the approximate position is not at hand.

Finally, a further characteristic of the Kalman filter approach is that if sensory data are not extremely sparse the state vector becomes very high-dimensional. As a consequence, in order to neither giving a too optimistic nor a too conservative state estimate, all coefficients of the resulting, very high-dimensional variance-covariance matrix have to be considered [CTS97]. The SPmap (symmetries and perturbations map) [CT96, CMNT99] as well as a modified filtering technique [HH01] present possible solutions to this problem.

2.2 Probabilistic Approach

In recent years several researchers have focused on a probabilistic approach to concurrent localization and map building. On basis of probabilistic models of the robot's motion and the robot's perception, environmental maps are built and updated from sensory readings and control input to the robot. The optimization criterion in this case is finding the most likely map given the sensory readings at different positions and the control input. According to the probabilistic nature, the robot's current position is given as a probability density function (PDF) defined on the currently optimal map [TBF98]. This procedure is also denoted as Markov localization and like the Kalman filter aims at building an exact, geometrically consistent map of the environment [SK95, SK97, FBTC98, BCF+98, TBF00]. As detailed in [BCF+98], Markov localization only applies if the robot's pose is the only (variable) state in the environment, i. e. again, a static environment is required. This is why in [FBTC98] an entropy and a novelty filter are suggested to provide Markov localization only with those pieces of the environment that are actually static.

In contrast to the state vector of the Kalman filter, the PDF does not make a definite statement about the robot's position in its environment. This becomes especially obvious if due to ambiguities the PDF is not a unimodal, but a multimodal distribution featuring several peaks, thereby supporting several position estimates, which might be far away from each other. In the work of [TBF98] these ambiguities are explicitly allowed, as the robot's capabilities of accurate measurements and of recognizing places or landmarks are very limited. Resolution of ambiguities is accomplished by a roaming robot gathering sensory information, which is successively incorporated in the map. Along that way, one peak of the PDF will gradually increase whereas the others will drop. This means that after several map and robot's position updates, one position estimate will be preferred to the others.

From the above mentioned it becomes clear that the probabilistic approach has a global view on the localization task. The advantages gained from this are feasibility of global localization and ability to recover from erroneous position estimates [BDFC98]. The major negative implication is the extreme computational effort: In order to find the most likely map, all possible maps have to be considered. According to [TBF98], this involves a search in a space with typically 10^6 dimensions. Furthermore, for each map the PDF has to be evaluated with reasonable resolution, involving an integration over 10^5 independent variables. Although there are techniques to efficiently search the likelihood space [DLR77] also in the present context of map building

[SK97], the effort for a map and position update remains enormous and is far away from being performed in real-time.

This is why a considerable amount of literature tries to alleviate this computational burden: Monte Carlo localization as a version of Markov localization is introduced in [FBDT99] and improved in [TFB00]; Condensation is suggested in [JAWA00]. Generally, the Monte Carlo method as well as Condensation belong to the class of so-called particle filters. The underlying idea is to represent the PDF by a discrete, sample-based approximation, i. e. weighted samples (particles) approximate the PDF. When performing a robot position update, a new sample is generated by randomly drawing a sample from the previously computed sample set with position and likelihood accordingly determined. The fact that a new sample is always derived from an existing sample has several implications: Whenever the robot's position is approximately known, i. e. when the existent sample set is rather dense, only a small fraction of the PDF has to be actually considered. Secondly, computational effort is only spent where it is really needed, namely around the position where the robot is supposed to be. Furthermore, computational effort can be adjusted by the number of samples and can be weighed against the desired accuracy. The capability of relocalization in case the robot loses its track can be maintained by adding a small number of uniformly distributed, random samples. However, the accuracy of relocalization is reduced according to the reduced density of samples. Summarizing the above it is to say that it is thanks to the particle filter methods that Markov localization with fine spatial resolution is feasible at all. Plain Markov localization without sample-based PDF approximation would be far too expensive due to its excessive computational and storage requirements.

2.3 Scan Matching Approach

Kalman filter as well as the probabilistic approaches can be used for any type of sensor: As outlined above, Kalman filtering often uses odometry or the vehicle control input besides some kind of environment perceiving sensor. As well, in [TBB+99] a monocular video system pointed at the ceiling is employed to support the probabilistic map building approach. In contrast, the techniques detailed in the following are limited to the range sensor domain.

The key idea of localization by scan matching is to determine a displacement vector between two 2D range scans, which are taken from different positions, by shifting and rotating the two scans against each other in such a way that

the two scans optimally coincide (cf. chapter 5). From this shift and rotation a displacement vector can be gained that provides (relative) localization.

The first technique in this context is a crosscorrelation approach presented in [WWvP94] and expanded to a map building method in [WvP95]: From the original range scans so-called angle and distance histograms are extracted and crosscorrelated. The maxima of the respective crosscorrelation functions represent the rotation and the shift in x- and y-direction between the two scans. Inherent problems are that the extracted histograms are extremely noisy and that the procedure only allows for minor changes of the environment. This means that even if odometry is used for an initial position estimate, evaluation of the crosscorrelation functions might be ambiguous or erroneous when the displacement between the two scans becomes larger.

A second, wellknown technique is the Cox algorithm presented in [Cox91]: Here, the single scan points of a range scan are assigned to a rectilinear line of the environmental map. To allow for a pairwise scan matching, a line extraction procedure has to be applied to one of the scans, so that this scan can then act as the environmental map. The choice of the line extraction procedure and which of the two scans it is applied to of course have an influence on the matching result. The Cox algorithm itself operates iteratively and tries to minimize an error sum by shifting and rotating the two respective scans against each other. Formally, it is a technique to calculate a position correction vector, i. e. a rather accurate position estimate, e. g. from odometry is necessary to avoid misassignments of scan points and to guarantee that the algorithm converges to the global optimum. The final output of the Cox algorithm is a position correction vector and a variance-covariance matrix that rates the match between the two scans.

Whereas the previous approaches only apply to polygonal environments, the IDC algorithm (Iterative Dual Correspondence) also performs well if surroundings are arbitrarily shaped [LM94]. This is accomplished by assigning the scan points of both scans directly to each other without an intermediate feature extraction step. By its nature, the IDC algorithm is like the Cox algorithm an iterative procedure that aims to minimize an error sum. The core of the algorithm are two heuristic scan point assignment rules: The closest-point rule and the matching-range rule. Simply put, the matching-range rule provides that the two scans are correctly rotated against each other, whereas the closest-point rule is responsible for the proper shift between the two scans. Again, an initial position estimate is mandatory to avoid erroneously assigned scan points. Just as the Cox algorithm, IDC gives a correction vector and a variance-covariance matrix.

However, the variance-covariance matrix calculated by the IDC algorithm proves to be too optimistic in polygonal environments. Furthermore, computational effort of the IDC algorithm exceeds that of the Cox algorithm. This is why [GS96] suggest a combined scan matching method, denoted as CSM, which combines Cox and IDC algorithm in that way that the computationally cheaper and more accurate Cox algorithm is applied in polygonal environments, whereas the IDC algorithm is only employed in non-polygonal surroundings.

As expressed in [GBFK98], localization using the IDC algorithm is more accurate than Markov localization. On the other hand, Markov localization proves to be more robust against noise induced by changes in the environment. In order to reduce noise sensitivity of IDC, [BB99] developed IDC-S, Iterative Dual Correspondence – Sector. In this approach the scans are divided into sectors, finally matching only those sectors between the two scans that contain the parts of the environment that remained unchanged.

If it is to establish a geometrically accurate and consistent map in a spacious environment, large cycles in the robot's path are a significant problem. Odometry is of course not suited to close these loops as the accumulated error will be prohibitive to correctly overlap the respective scans. This is why scan matching techniques are employed to provide a solution: In [LM97] the concept of consistent pose estimation is introduced, which, however, requires topologically correct maps. In [GK99] the Local Registration and Global Correlation (LRGC) method is presented that does not have such requirements: Within a certain search area not only one single scan, but several scans that have been integrated into a map patch are matched against the map built so far. Along that way, places that have already been visited and mapped are reliably recognized and cycles can topologically be closed. However, in order to provide a geometrically correct overlap of all scans being part of such a cycle, all poses of the cycle from which scans are taken have to be accordingly updated, i. e. actually closing a large cycle induces significant computational effort.

2.4 Topological Maps

In the approaches mentioned so far, the maps that were built and used for mobile robot localization aimed to be geometrically exact images of the environment. In the following, some alternative approaches will be outlined that employ topological maps for this purpose.

Following [KB91], a topological map consists of *distinctive places*, which are relevant for the mission of the mobile robot, and arcs, which connect two distinctive places in case they are adjacent. Thus, an appropriate representation for a topological map is a graph with nodes (distinctive places) and edges (arcs). As only mission relevant places together with their respective neighbourhood relations are maintained in a topological map, the amount of data to be stored and processed can be significantly reduced compared to a geometrically exact map where all parts of the environment have to be considered (cf. section 6.4).

The price for these computational savings is that in general a pinpoint position estimate has to be abandoned, i. e. the robot's whereabouts can only be roughly given to *somewhere in the vicinity of a distinctive place* or when an edge is followed to *somewhere between two distinctive places*. In [KB91] this lack of precision is alleviated by a so-called hill climbing technique that enables the robot to at least find the exact position from where the respective distinctive place has initially been seen.

Several researchers seized this basic, topological technique and extended or combined it with other methods: In [SK95] and [JK99] a topological map is a priori given to support probabilistic navigation and localization, respectively.

[KWN99] present a method to autonomously acquire a graph based map. Assuming a rectangular environment, the procedure suggested in this work relies on odometric data and sonar readings in order to distinguish between different places. Similarly, but employing a probabilistic approach [SK97] also learn a topological map from odometry and sonar.

Finally, in [TGF+98] it is allowed that the distinctive places degrade to *significant places*, which are still reliably detected, as this is done by a human operator, but are actually indistinguishable. This leads to topological ambiguities, which are resolved utilizing the capability of Markov localization to handle multimodal distributions, i. e. multiple position hypotheses (cf. section 2.2). Furthermore, a metric mapper is added in order to also provide a high-resolution geometric map.

2.5 Classification of the Present Work

The localization method presented in this work ranges in the category of scan matching approaches. However, it differs from the work carried out so far in several aspects: Firstly, the procedure employed for scan matching is an

algorithm borrowed from the pattern matching and recognition domain (cf. section 5.5) and is therefore completely self-contained. This means that for a basic matching operation only the two range scans are required; additional information e. g. in the form of an initial positional guess from the robot's odometry, is in no way necessary. On the other hand, if additional information is reliably available, it can smoothly be integrated if desired in order to reduce possible ambiguities.

As detailed above, the IDC algorithm is sensitive to changes in the environment, i. e. occluding and disclosing of parts of the surroundings due to a changing viewing angle while travelling. In contrast, the presented approach can very well cope with this effect, i. e. the ability of recognizing places that have been visited before [Zim95] is highly developed. This very crucial property is demonstrated by means of several experiments presented in chapter 7.

The employed map building approach follows the concept of the above mentioned topological maps. As the suggested implementation is rather straightforward, it is principally sensitive to ambiguities. However, due to the sophisticated matching procedure paired with the fact that the employed sensor system features a nearly 360° view in contrast to the commonly found 180° devices, ambiguities are not very frequent and can generally be resolved taking into account rough odometric estimates.

Chapter 3

Sensor System

3.1 General Description

The core of the sensor system used in this work is the AccuRange 4000-LV manufactured by Acuity Research [Acu96]. It is a laser diode based distance measurement sensor for ranges up to about 15 m. The sampling rate of the range sensor is adjustable and can be increased to a maximum of 50000 range measurements per second. The sensor emits visible red light at a wavelength of 670 nm and has an optical power of 5 mW, rating the instrument as a class IIIa laser product which can be granted eye safeness in scanning applications.

To obtain a panoramic 2D line scan of the surroundings, the range sensor is supplemented by a scanner consisting of an elliptical metal mirror and a DC motor with position encoder. As can be seen from figure 3.1 the mirror deflects the vertically outgoing beam by 90° into a flat planar scan. Further, it collects diffusely reflected return light and directs it into the sensor's collection lens. The mirror's rotary speed also is an adjustable parameter and can be set up to 3000 rpm. The scanning unit, designed to minimize shadowed areas, allows an overall viewing angle of about 350° excluding only two narrow sectors for the scanner's rails.

The uplink of the panoramic laser range finder (PLRF) to the data processing host computer is realized via a standard ISA interface board. Raw sensor data consisting of uncalibrated range samples together with correction coefficients and encoder readings are buffered on the interface and are segmented into single scans. The host reads these data and provides the necessary calibration, i. e. correction for target reflectivity, ambient light and sensor

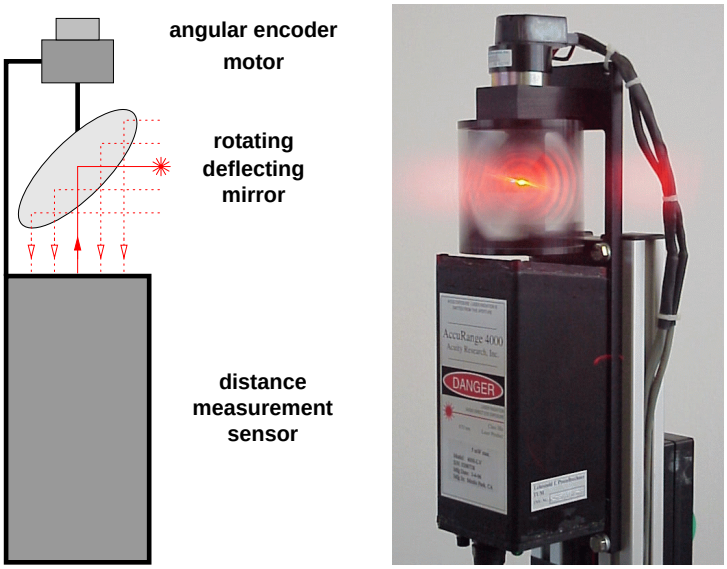


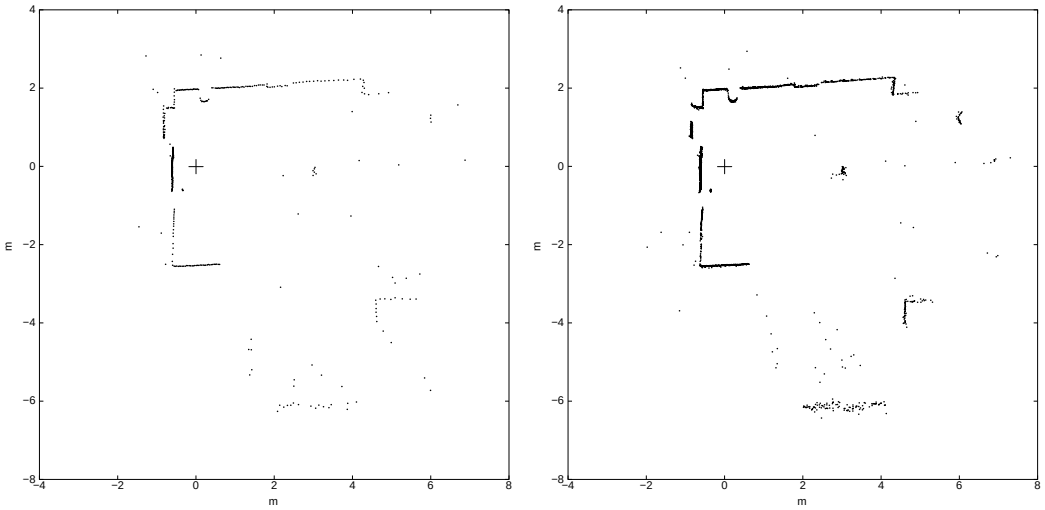
Figure 3.1: Panoramic Laser Range Finder (PLRF)

temperature. To issue commands from the host to the PLRF, there are two communication channels. Commands concerning the range sensor itself, e. g. switching the laser beam, are sent via a standard RS-232 serial port. Any other commands, i. e. interface and scanning unit control, are addressed to the ISA board.

The host computer in this setup is a standard PC running under the Linux operating system. Due to the PLRF's constant sampling rate and limited memory on the interface, reading data from the interface is a time-critical task. Therefore, this task is executed within a high priority interrupt service routine under control of a driver programme running in Linux kernel mode.

In contrast to other commercially available sensor systems offered by the manufacturers Sick, Leutze and Ibeo, the PLRF used in this work features a combination of large viewing angle, high sampling rate and, particularly, numerous accessible parameters for probing and testing: Sampling rate, mirror's rotary speed, maximum allowed range and laser power level can be adjusted and experimented with. The correction coefficients, i. e. return signal amplitude, ambient light amplitude and sensor temperature attached to every single range sample are not kept sensor internal, but are open to the developer.

Figure 3.2 shows two typical panoramic scans from our PLRF acquired with different parameter settings for sampling rate and mirror speed.



(a) (b)

Figure 3.2: Typical scans from our PLRF with (a) sampling rate 16667 samples/s and mirror speed 2190 rpm, leading to 456 range samples per scan and (b) sampling rate 50000 samples/s and mirror speed 1280 rpm, leading to 2344 range samples per scan.

3.2 Performance and Measurement Accuracy

This section is a general discussion of factors that affect the PLRF's performance and relies on the manufacturer's specification given in the user's manual [Acu96] as well as own experiments.

The PLRF's range sensor will detect diffuse reflections from objects of any colour with greatest sensitivity falling at about 2.5 m, although short distances right up to the front face of the sensor can be measured. It has no trouble picking up walls, floors, carpets, and even surfaces such as CRT screens from almost any angle. Shiny surfaces such as glossy plastic or paint can be more difficult to detect, depending on the angle at which the beam hits them. Completely invisible for the sensor are window panes and, naturally, any kind of specular material.

There are three types of noise that will affect the measurement accuracy in different ways. They are described below, but each has a range of sampling rates at which it is the predominant source of noise. Figure 3.3 shows the accuracy limit imposed by each type of noise for a given sampling rate. The first type is detector thermal noise, which originates in the signal detection photodiode, and is proportional to the square root of the sample rate. The

second type is laser diode noise, and the third type of noise is the resolution limitation imposed by the sampling method.

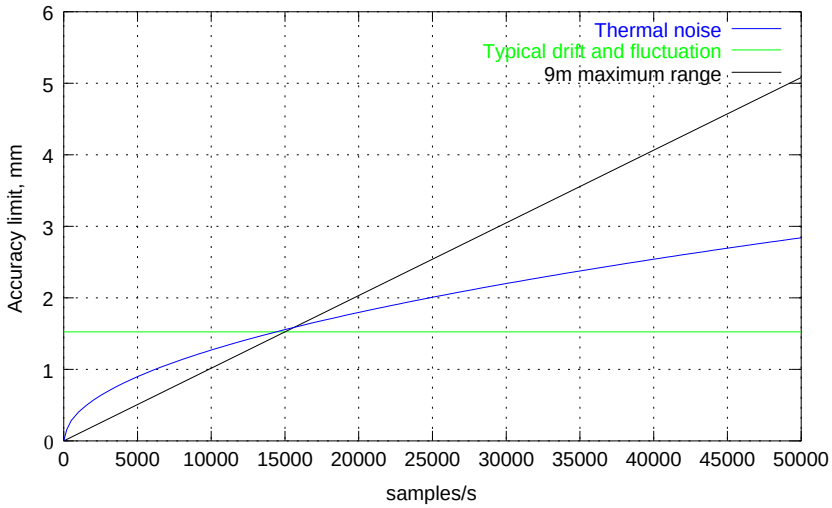


Figure 3.3: Limitations on accuracy versus sampling rate, 85% diffusely reflecting target

The vertical scale in figure 3.3 is the attainable accuracy, while the horizontal scale is the sampling rate. Each line represents a different constraint on accuracy due to noise or sampling resolution. For any sampling rate, the highest line at that rate represents the limiting factor and the attainable accuracy. At low sampling rates, i. e. below 10000 samples/s the limiting factor is the laser diode noise, shown as the horizontal line. At higher sampling rates the limiting factor becomes the detector's thermal noise, shown as the curved line proportional to the square root of the sampling rate. At the highest sampling rates, the sampling resolution becomes a factor, and the diagonal line shown in figure 3.3 represents the limitations of the sampling resolution with a maximum range of 9 m.

3.2.1 Detector Thermal Noise

Range measurement accuracy at high sampling rates is limited by thermal noise in the sensor's detector. Typically, a range measurement will be made by timing a number of cycles of the output. The larger the number of cycles timed, the better the averaging or filtering of this noise will be. According to the theory of noise power and noise bandwidth, the effect is that the standard deviation of the measurement error increases proportionally with the square

root of the signal bandwidth, or in this case the sampling rate. The noise in this sensor is $0.0127\,\text{mm}/\sqrt{\text{Hz}}$. Multiplying this value by the square root of the sampling rate will give the root-mean-square noise value (approximately the same as the standard deviation) for the measurement. Thus, a sampling rate of 10000 Hz gives readings with a standard deviation of about 1.27 mm.

3.2.2 Laser Diode Noise

There is another source of measurement error that needs to be considered when taking high accuracy measurements, caused by noise in the laser diode. This noise is characterized by random changes in the range reading that tend to increase with the time over which the readings are taken. This becomes noticeable over times of about 0.3 s or more, and increases up to times of several hours. The standard deviation of this drift is about 0.25 mm at 1 s, and up to 2.5 mm in 10 h.

3.2.3 Maximum Range Specification

One of the configuration parameters for the range sensor is the maximum expected range. This is to allow the sensor to obtain readings with the best possible resolution and accuracy. Internally, the time required to take a single sample depends on the distance being measured and the resolution used to take the measurement. If ranges are known to be short, better resolution and accuracy at high sampling rates may be obtained by reducing the maximum range.

3.2.4 Range Resolution

The diagonal line shown in figure 3.3 is an accuracy limit due to sampling resolution, assuming that the ranges to be measured are 9 m or less. This becomes the limiting constraint above 15000 samples/s. For ranges up to 18 m, the limitation would be a similar line with twice the slope. This is due to the fact that longer ranges take more time to resolve to the same precision.

3.2.5 Angular Resolution

The angular resolution of the PLRF is primarily determined by the scanner's position encoder which, in our case, features 2000 position counts/revolution, limiting angular resolution to 0.18°.

However, for high sampling rates together with lower rotary speeds this resolution may not be sufficient to guarantee that successively taken samples actually obtain differing encoder readings. To work around this shortcoming, the encoder additionally delivers an index pulse indicating a full revolution of the mirror. This can be utilized to determine the number of samples taken in the current scan N_ν. Assuming that the rotary motion of the mirror is uniform, which is sufficiently granted when the mirror has completely accelerated and the rotary speed is not too low, the samples along the scan are uniformly distributed as well. In this case, the angular resolution of the current scan $\Delta\varrho_\nu$ can be calculated independent of the encoder's angle readings according to:

$$\Delta\varrho_\nu = \frac{360^\circ}{N_\nu} \tag{3.1}$$

The main benefit of this procedure is that angular resolution only relies on the encoder's index pulse and can therefore be almost arbitrarily refined (cf. figure 3.2 b).

3.2.6 Other Factors Affecting Performance

In addition to noise, there are other factors that affect the indicated range output. One is the already mentioned amplitude of the return signal or reflectivity of the target. Indicated range can vary as much as 7.5 cm between very weak signals and very strong ones. Therefore, for each range sample the sensor produces an 8 bit signal strength output to take this effect into account. Amplitude output can also be used to create gray scale images of objects over which the beam is scanned, and to determine whether a sample is valid or too weak to be reliable.

Sensor temperature and ambient light level also affect the measurement slightly. Temperature and ambient light outputs allow these effects to be compensated for in software. In this work, they are not considered significant as these parameters do not vary widely in indoor environments.

An additional source of inaccuracy are the so-called outliers. These are sporadically occurring measurements that are wide off the right value bearing an

error far above noise level. For our PLRF, there are two prevalent reasons for outliers: The first are sudden range changes in the surroundings producing an output lying between the two distances. This is due to the fact that the range is averaged over the sampling duration, and also due to the fact that for a period of time, the laser beam will fall partially on each surface. The other reason are specular reflections that usually lead to measurements lying too far away. Both forms of outliers can be found in the scans of figure 3.2.

If the PLRF is installed on a moving platform, one has to consider that scan acquisition is in fact not a snapshot, but needs a certain period of time. As a consequence, acquiring a scan while the PLRF is in motion leads to a distortion of the scan affecting range as well as angular accuracy. This distortion is dependent on mirror speed, which determines acquisition time, and platform velocity. For example, assuming mirror speed to 3000 rpm which equals an acquisition time of 20 ms and platform velocities of 1 m/s and 180 °/s, respectively, the worst case distortion for a range sample can reach up to 2 cm and 3.6°.

Chapter 4

Sensor Data Preprocessing

From chapter 3 we learned that a panoramic scan from our PLRF has nearly a 360° view and consists of numerous measurements being subject to different kinds of noise, perturbation and distortion. This chapter focuses on different techniques of data reduction and analysis which are applied in order to extract the relevant information from a scan. Hereby, emphasis is laid on an algorithm being able to find 2D line segments.

4.1 Notation of a Scan

A panoramic scan is an ordered sequence of N measurements. Each measurement, denoted in the following as *scan point*, defines in polar coordinates, i. e. range r_u and angle ϱ_u, where the laser beam hits an object in the surroundings. Due to the scanning nature of the data acquisition process, it is ensured that the angles ϱ_u occur in an ascending order within this sequence. This is an important property that will be referred to later on. Thus, the formal notation for a scan is given by

$$S = \left\{ s_u = [r_u, \varrho_u]^T \;\middle|\; 0 \leq u < N \right\} \quad \text{with} \quad \varrho_u < \varrho_v \,;\, 0 \leq u < v < N \tag{4.1}$$

4.2 Classification

Current preprocessing methods for scan data can be separated into two subsections. Filtering techniques process a scan by modifying and/or removing

scan points. Thereby, dependent on the applied filter characteristic, noise can be smoothed, outliers are rejected and the number of scan points is reduced; but the scan in its characteristic is left untouched and remains a sequence of single measurements. Contrary to this, feature extraction tries to fit a number of scan points to a predefined geometric primitive that can be specified by parameters, e. g. line segments, corners or arcs. Again, noise and perturbation elimination is achieved as well as a very significant data reduction. However, in addition to that, the characteristic of the scan is changed as it is no longer a sequence of scan points, but a sequence of the respective geometric primitives. If a priori knowledge about the surroundings is available, e. g. a polygonal environment, feature extraction usually shows better results concerning clutter rejection than filtering. Furthermore, the extracted features can directly be used as input for a higher level processing. On the other hand, filtering performs better if there is no knowledge about the environment or if the polymorphic nature of the surroundings forbids any assumptions. Of course, filtering and feature extraction can be combined, e. g. a filter for outlier rejection with a subsequent extraction of line segments would be a possible approach. In the following, a few selected techniques of both preprocessing methods are outlined.

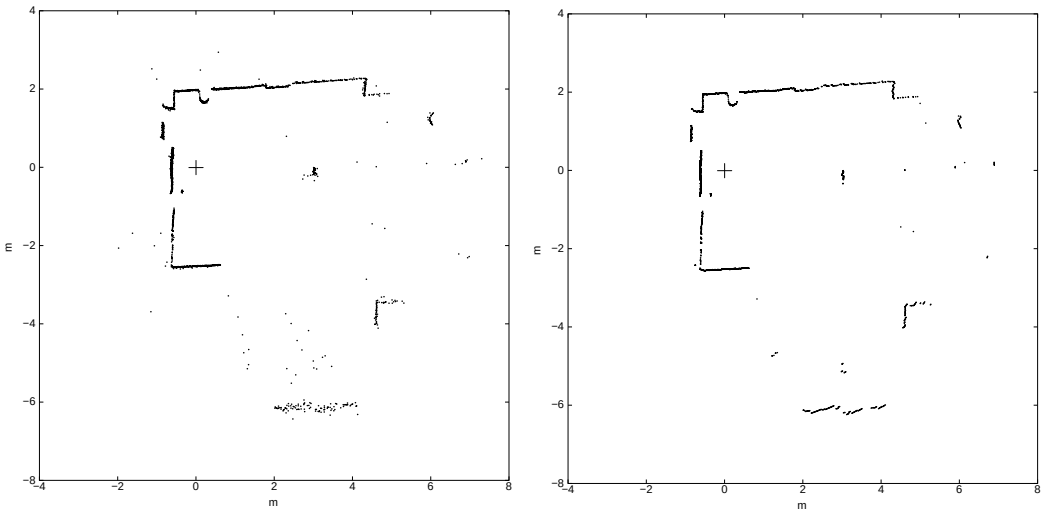
4.3 Filtering of Scan Data

4.3.1 Median Filter

A median filter is mainly used to reject outlying scan points. In contrast to low-pass or bandpass filters, this can be achieved with only marginally smoothing or blurring the structures in the scan, as the median filter is based on a nonlinear calculation: Considering a scan point u, a window containing W scan points is centered over that scan point. Then, the scan points within the window are sorted according to their range value, thereby permuting the original order. In the general case, the center position of the window will now bear another range value originating from somewhere in the window. This new range value is denoted as the median for scan point u. Before proceeding to the next scan point $(u + 1)$, i. e. shifting the filter window by one scan point, the effects of the sorting are discarded. To process a whole scan this procedure has to be repeated for all N scan points, leading to a complexity of $O(N\,W)$. As large window widths are able to distort the scan, i. e. rendering structures where there are none, typical values for W are rather small.

As can be seen from the algorithm, the estimation of the median is quite robust, because it does not break down until half of the scan points within the window are outliers.

Figure 4.1 shows the scan of figure 3.2 b in its original and its median filtered version. Well noticeable is the significantly reduced number of outliers.



original median filtered

Figure 4.1: Scan of figure 3.2 b in its original and median filtered version. The width of the filter window was set to $W = 5$ scan points.

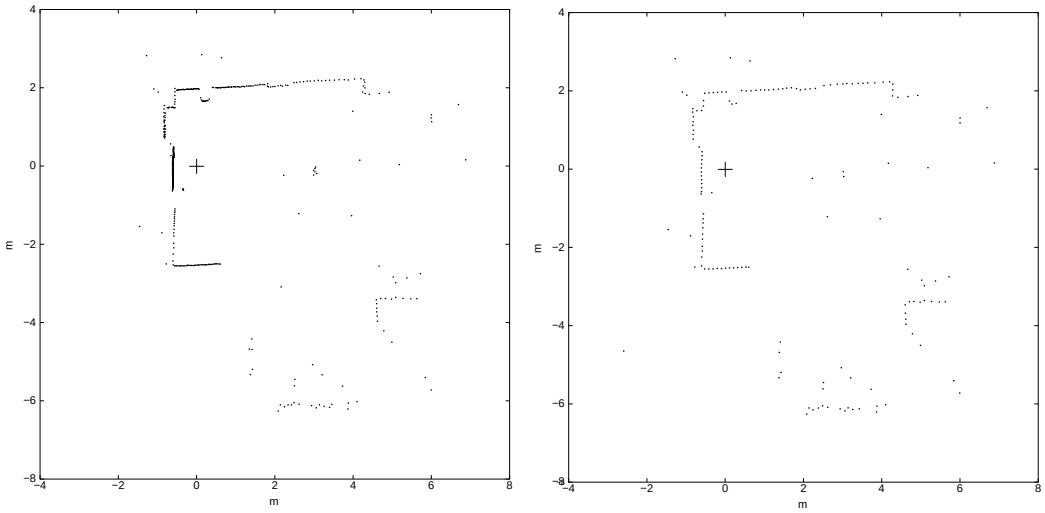
4.3.2 Reduction Filter

The approach of a reduction filter is to approximate a group of scan points by one single new scan point, thereby reducing the overall number of points. A group of scan points is specified by its radius defining a circle within which all scan points that belong to that group must lie. The new scan point substituting a group of scan points is simply calculated as the mean value of the group members. Consequently, a large radius leads to widespread groups and therefore to a strong reduction.

Another effect is that the filter equalizes spatial scan point density. In areas close to the sensor, where density is high, groups contain numerous scan points and therefore thinning is quite strong. Far away from the sensor, where density is low, groups consist of only few scan points leading to a quite moderate reduction. As a result from this, distances between successive scan points become more equal throughout a scan.

The main benefit of the filter is data reduction, saving subsequent modules computational effort. A problem arising from this reduction is the possible loss of important structural information. As well, the equalized scan point density is not necessarily a gain, as a finer spatial resolution when being close to an object is in fact helpful. That is why in practice the radius has to be chosen very carefully, not exceeding values of a few centimeters. An $O(N)$ algorithm implementing the reduction filter can be found in [Gut99].

Figure 4.2 shows the thinning and equalizing effect of a reduction filter applied on the scan of figure 3.2 a.



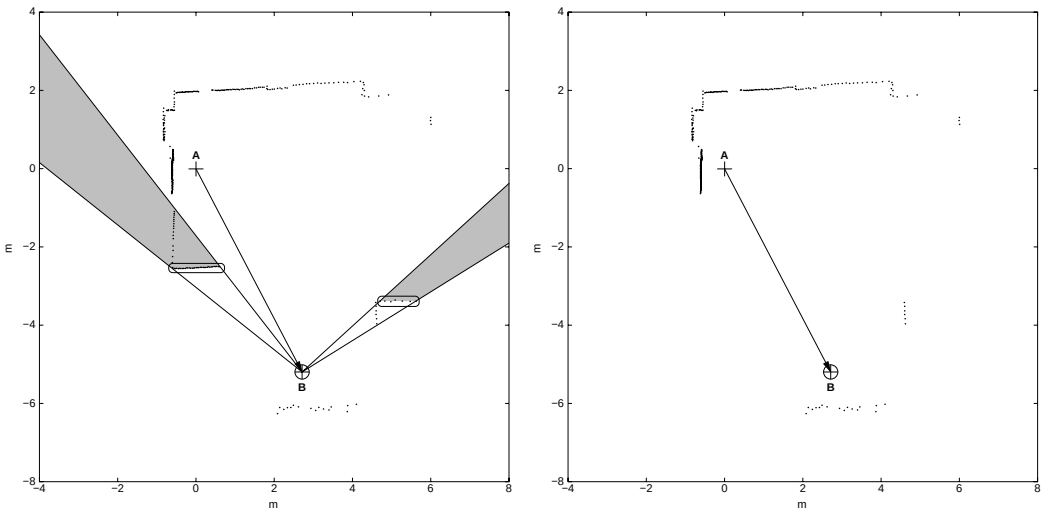
original reduction filtered

Figure 4.2: Scan of figure 3.2 a in its original and reduction filtered version with only 156 scan points. For illustrative reasons, the radius of the groups has been chosen to the very large value of $R = 5\,\text{cm}$.

4.3.3 Projection Filter

A projection filter is a sifting process, only preserving selected scan points and discarding the others. The sifting considers the following perspective criteria: Given a scan taken at position A and further given a new position B, e. g. as a displacement vector emanating from the position A, only those scan points are preserved that would be visible from the new position B. Any other scan points that are not visible or shadowed will be removed from the scan. So the actual procedure is a projection of the current scan onto the new position B.

To realize the filter, one utilizes the property that within a valid scan, successive scan points feature monotonously growing angle readings. If this rule is violated when projecting the scan points to the new position B, the respective structures are seen "from behind". This means that they are actually invisible from position B and that the scan points of the affected angle sectors must be removed to render the scan valid again. Additionally, the affected sectors screen other structures (grey areas in figure 4.3 a), which were previously visible. So the scan points of these screened structures have to be removed, too. The overall result of the projection filter is illustrated in figure 4.3 b. The appropriate algorithm has complexity $O(N^2)$ and can also be found in [Gut99].



(a) (b)

Figure 4.3: Projection Filter: (a) Scan taken at position A with scan points boxed that are invisible from position B. The resulting screened areas are shaded in grey. (b) Result of filtering.

The value of this filter is twofold. First, it is again data reduction. This time not by modifying scan points, but by focusing on the respectively visible parts of a scan. Second, ambiguities that possibly arise when dealing with multiple scans from different views can be avoided. A handicap of the projection filter is that, in contrast to the median and the reduction filter, it is not self-contained. As mentioned above, it relies on additional input in form of a new position or a displacement vector. This information cannot be drawn from the scan itself, but must be provided by another source. Thereby, the quality of the filter result fundamentally depends on the accuracy of the estimate of the new position on which the scan is projected.

4.4 Feature Extraction

4.4.1 Extraction of Line Segments

If the surroundings can be assumed to be of polygonal shape, the extraction of line segments will be an appropriate method. The basic concept is to substitute a scan by an ordered sequence of n line segments approximating the original run of the scan points. Each line segment ℓ_i is specified by four parameters, which are in the present work chosen to be the Hesse normal notation $[a_i, \alpha_i]^T$ with distance a_i and angle α_i for the underlying line, and two additional angles $[\varrho_{si}, \varrho_{ei}]^T$ to define start and end point on the line, respectively (cf. figure 4.4).

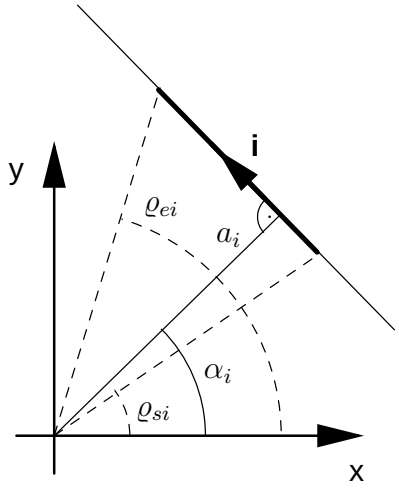


Figure 4.4: Parametric specification of a line segment ℓ_i.

Consequently, the formal notation of a scan expressed by equation (4.1) is changed to

$$L = \left\{ \ell_i = [a_i, \alpha_i, \varrho_{si}, \varrho_{ei}]^T \;\middle|\; 0 \leq i < n \right\}$$
$$\text{with} \quad \varrho_{si} < \varrho_{ei} < \varrho_{sj} \,;\, 0 \leq i < j < (n-1) \quad (4.2)$$

As it is the case with the scan points, the angles ϱ_{si}, ϱ_{ei} occur in increasing order. Again, this property will be utilized.

In the literature several approaches are to be found that solve the problem of extracting line segments from a sequence of points. The main differences between them can be observed in performance and complexity. Histograms

[WWvP94] or the Hough transform [FLWÅ93] are applied, as well as a novel solution called Adaptive Least Kth Order Squares (ALKS) that is highly adapted to the statistical characteristics of the data points [LMP98]. Those methods are able to reject outliers and join collinear segments, but all feature complexities of $O(N^3)$ or even more. Clustering techniques operating in (a, α)-feature space [VTGA94] require moderate effort, but are sensitive to noise when spatial density of scan points is high. The algorithm preferred in our case originates from image processing [PH74] and has a worst case complexity of $O(N^2)$. It gets along with noise, but is sensitive to outliers, i. e. line segments split when outliers occur. The algorithm is a recursive two-stage procedure, which has successfully been applied on scan data in [Gut99]. The actual realization proposed here preserves this two-stage nature, but introduces a better adaption to data points provided by a scanner with constant angular resolution.

Before going into the details, the algorithm will be roughly outlined. In the first step, the scan points of a scan are divided into an arbitrary number of groups consisting of consecutive scan points in their natural order. As a result, every group features a dedicated start and end scan point. Reasonable for the practice is any number of groups between 2 and 10. The reason for grouping is to avoid that large parts of the scan are considered over and over again having very little chance to belong to one line segment due to the panoramic nature of a scan.

The second step contains the recursive part (cf. figure 4.5). Start and end scan point of a group are connected by a straight line and for all scan points lying between start and end scan point the distances $d_{\perp,u}$ to this line are calculated. If the maximum of these distances $d_{\perp,\max}$ exceeds a given threshold $d_{\perp,\text{th}}$, the line is split at the scan point producing this maximum and two new straight lines are established: The first leading from the start scan point to the maximum distance scan point, the second emanating from the maximum distance scan point to the end scan point. In the following, the above procedure of maximum distance calculation, thresholding and possible splitting is recursively called until no more splitting takes place, i. e. the whole group of scan points is approximated by lines. In order to improve accuracy of the line parameters, an orthogonal linear regression is performed for each subgroup of scan points that is associated with a straight line. As start and end points of the extracted lines can easily be determined, a line segment description as introduced above, which also regards the direction of a segment, is provided. To avoid clutter, only line segments exceeding a minimum number of associated scan points are eventually established. So far the core of the algorithm. In the following, some aspects will be considered

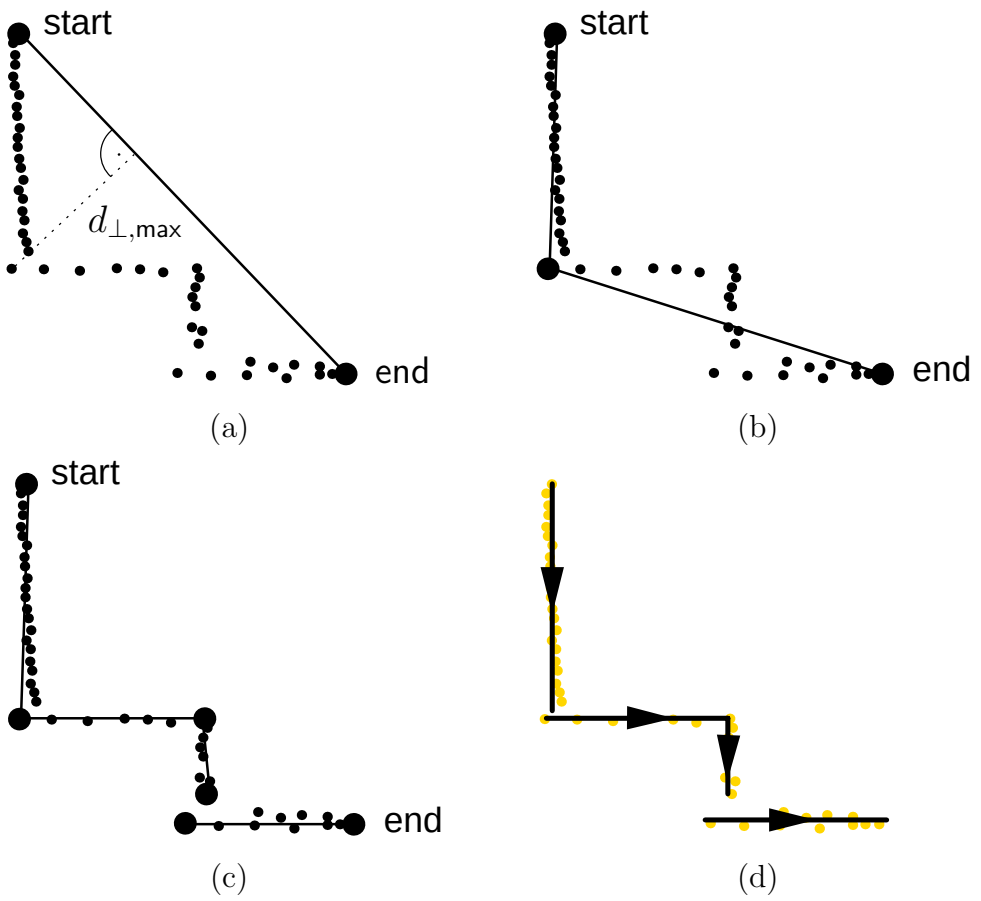


Figure 4.5: Line segment extraction within a group of scan points: (a), (b) First iteration of a line extraction cycle with maximum distance calculation and splitting. (c) Completed line segmentation. (d) After orthogonal linear regression.

in detail.

Due to the presence of multiple groups with boundaries arbitrarily set, it will happen that the end scan point of a group does not coincide with the end point of a line segment, as the line segment exceeds the group's end point. Since additional splitting of line segments is undesirable, the following proceeding is chosen(cf. figure 4.6). Before the last line segment within a group is established, the current end scan point of the group is replaced by the end scan point of the subsequent group. Afterwards, the iterative procedure as detailed above is started all over with unaltered start scan point and the new end scan point. In this way, all the groups of a scan are subsequently processed, using the originally set boundaries only as a first clue. Obviously,

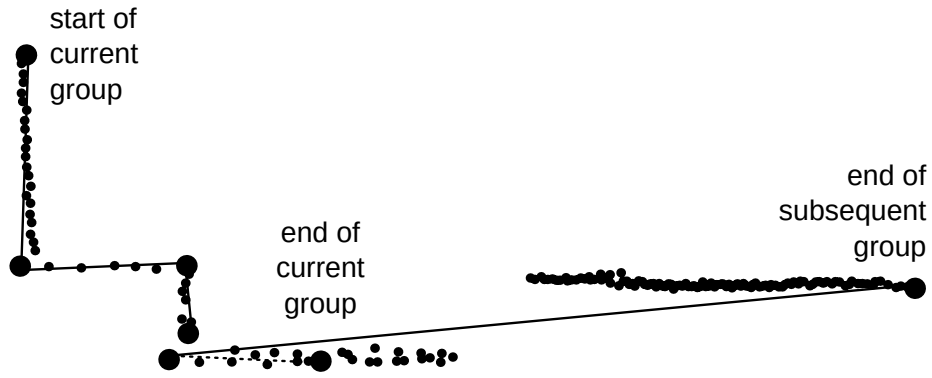


Figure 4.6: Segment extraction when crossing a group boundary.

this results in a re-processing of a certain portion of the group's scan points. However, this procedure implicates a benefit that will be explained below.

So far, only the (perpendicular) distance $d_{\perp,u}$ of a scan point to the current line was under consideration. Now the distance $d_{\parallel,u(u+1)}$ between two consecutive scan points u and $(u+1)$ that is parallel to the current line segment will also be used. If $d_{\parallel,u(u+1)}$ exceeds the threshold $d_{\parallel,\mathrm{th}}$, the current line segment will be split between the two scan points u and $(u+1)$. This means that in contrast to [Gut99] where only the absolute value of the distance

$$d_{u(u+1)} = \sqrt{{d_{\parallel,u(u+1)}}^2 + \left(d_{\perp,u} - d_{\perp,(u+1)}\right)^2} \tag{4.3}$$

is regarded (cf. figure 4.7), the proposed version of the algorithm treats the two distance components $d_{\parallel,u(u+1)}$ and $d_{\perp,u}$ separately. The advantage is evident: Due to the separate treatment, the two threshold values $d_{\perp,\mathrm{th}}$ and $d_{\parallel,\mathrm{th}}$ can be chosen differently. $d_{\perp,\mathrm{th}}$, which is crucial to resolve fine structures in the environment or to identify outlying scan points, is set to a rather small value of typically 5 cm. $d_{\parallel,\mathrm{th}}$, however, which is mainly used to avoid that a group of scattered scan points incidentally creates a line segment, can be set to a much larger value of about 25 cm. Along that way, the algorithm is on the one hand able to discern fine structures, and on the other hand it is capable of producing line segments even when distances between scan points grow larger, which is the case for long ranges.

In order to refine the parameter tuple $[a, \alpha]^T$ of the extracted line segments, an orthogonal linear regression is suggested. Contrary to ordinary linear regression, orthogonal linear regression assumes errors not only along the ordinate, but as well along the abscissa. This approach is also known as the

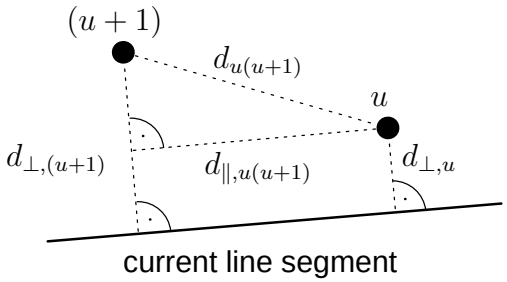


Figure 4.7: Distances considered when establishing a line segment.

errors-in-variables model or total least squares. As it is self-evident that errors in scan point position will affect both coordinates, orthogonal regression properly matches our case.

If $[x_u, y_u]^T$ with $0 \leq u < M$ is the cartesian description of the scan points for which a regression line is to be found, orthogonal linear regression realizes the optimization

$$E_{\text{TLS}} = \sum_{u=0}^{M-1} {e_{\text{TLS},u}}^2 = \sum_{u=0}^{M-1} (\cos\alpha \cdot x_u + \sin\alpha \cdot y_u - a)^2 = \text{Min} \qquad (4.4)$$

with error measure $e_{\text{TLS},u}$ denoting the perpendicular distance of a scan point to the regression line (cf. figure 4.8).

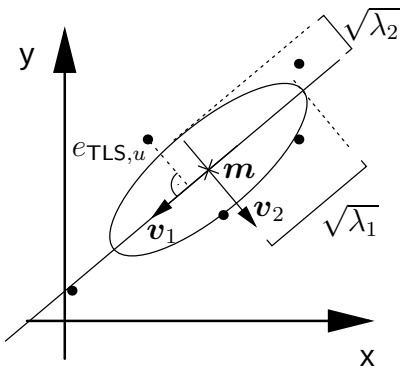


Figure 4.8: Orthogonal linear regression.

From the literature, e. g. [RL87] it is known that the above optimization problem can be equated with an eigenvalue problem. The matrix for which

eigenvalues and eigenvectors have to be determined is the variance-covariance matrix $\boldsymbol{C}$ set up by the distribution of scan points associated with the respective line segment. As can be seen from figure 4.8, the eigenvectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ are aligned with the principal axes of the distribution, and the eigenvalues λ_1, λ_2 are identical to the variances along these axes. In the present 2D case the variance along the second principal axis, i. e. λ_2 becomes minimal. So the accumulated error measure E_{TLS} that can be expressed as:

$$E_{\mathrm{TLS}} = (M-1) \cdot \lambda_2 \tag{4.5}$$

also reaches its minimum.

The actual formulae solving the optimization problem now read as follows:

Eigenvalues:

$$\begin{aligned} \lambda_1 &= 0.5 \cdot \left(s_{xx} + s_{yy} + \sqrt{(s_{xx} - s_{yy})^2 + 4\, {s_{xy}}^2} \right) \\ \lambda_2 &= 0.5 \cdot \left(s_{xx} + s_{yy} - \sqrt{(s_{xx} - s_{yy})^2 + 4\, {s_{xy}}^2} \right) \\ \lambda_1 &> \lambda_2 \end{aligned} \tag{4.6}$$

with (co-)variances:

$$\begin{aligned} s_{xx} = {s_x}^2 &= \frac{1}{M-1} \sum_{u=0}^{M-1} (x_u - \hat{x})^2 \\ s_{yy} = {s_y}^2 &= \frac{1}{M-1} \sum_{u=0}^{M-1} (y_u - \hat{y})^2 \\ s_{xy} = s_{yx} &= \frac{1}{M-1} \sum_{u=0}^{M-1} (x_u - \hat{x})\,(y_u - \hat{y}) \end{aligned} \tag{4.7}$$

defining the variance-covariance matrix:

$$\boldsymbol{C} = \begin{bmatrix} s_{xx} & s_{xy} \\ s_{yx} & s_{yy} \end{bmatrix} = \begin{bmatrix} {s_x}^2 & s_{xy} \\ s_{yx} & {s_y}^2 \end{bmatrix} \tag{4.8}$$

and mean values:

$$\begin{aligned} \hat{x} &= \frac{1}{M} \sum_{u=0}^{M-1} x_u \\ \hat{y} &= \frac{1}{M} \sum_{u=0}^{M-1} y_u \\ \boldsymbol{m} &= \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \end{aligned} \tag{4.9}$$

enable the calculation of eigenvectors:

$$\begin{aligned} \boldsymbol{v}_1 &= \begin{bmatrix} v_{1x} \\ v_{1y} \end{bmatrix} = \begin{bmatrix} \lambda_2 - s_{xx} \\ -s_{xy} \end{bmatrix} = \begin{bmatrix} s_{xy} \\ \lambda_1 - s_{xx} \end{bmatrix} \text{ belonging to } \lambda_1 \\ \boldsymbol{v}_2 &= \begin{bmatrix} v_{2x} \\ v_{2y} \end{bmatrix} = \begin{bmatrix} s_{xy} \\ \lambda_2 - s_{xx} \end{bmatrix} = \begin{bmatrix} s_{xx} - \lambda_1 \\ s_{xy} \end{bmatrix} \text{ belonging to } \lambda_2 \\ {\boldsymbol{v}_1}^T \cdot \boldsymbol{v}_2 &= \mathbf{0} \end{aligned} \tag{4.10}$$

which leads to the final formula for the refined tuple:

$$\begin{bmatrix} a \\ \alpha \end{bmatrix} = \begin{bmatrix} {\boldsymbol{v}_2}^T \cdot \boldsymbol{m} \\ \arctan \frac{v_{2y}}{v_{2x}} \end{bmatrix} \tag{4.11}$$

The eigenvalues λ_1, λ_2 calculated for each line segment can be used as an additional characterization: The larger λ_1, i. e. the variance in direction of the line segment, the longer the line segment is. The smaller λ_2, i. e. the variance perpendicular to the line segment, the more precisely the segment could be extracted from the associated scan points. So one can resume that the larger λ_1 and the smaller λ_2, the more significant and reliable the extracted line segment becomes. Later, we will refer to this qualitative statement and transform it into quantitative formulae.

4.4.2 Extraction of Corners

Extracting corners is based on line extraction. Successive line segments ℓ_i and $\ell_{(i+1)}$ are intersected and a corner is established, if the absolute value of the angle difference, i. e. $\left|\alpha_{(i+1)} - \alpha_i\right|$ exceeds the minimum threshold $\Delta\alpha_{\text{th}}$. The position of the corner with respect to the PLRF is identified by the intersection point, which is given as range $r_{i(i+1)}$ and angle $\varrho_{i(i+1)}$. So the parameter tuple specifying a corner reads as $\left[\alpha_i, \alpha_{(i+1)}, r_{i(i+1)}, \varrho_{i(i+1)}\right]^T$.

Again, an ordered sequence with an additional monotony criterion represents the corners within a scan:

$$\begin{aligned} C = \Big\{ c_{i(i+1)} = \left[\alpha_i, \alpha_{(i+1)}, r_{i(i+1)}, \varrho_{i(i+1)}\right]^T \mid \\ 0 \leq i < (n-1) \wedge \left|\alpha_{(i+1)} - \alpha_i\right| > \alpha_{\text{th}} \Big\} \\ \text{with} \quad \varrho_{i(i+1)} < \varrho_{j(j+1)} \,;\, 0 \leq i < j < (n-1) \end{aligned} \tag{4.12}$$

The algorithm implementing corner extraction is very similar to the line extraction procedure. As for corner detection only straight lines are needed,

indication of start and end angles is omitted. If the angle difference votes for the creation of a corner, the intersection point $\left[r_{i(i+1)}, \varrho_{i(i+1)}\right]^T$ is calculated according to:

$$\begin{bmatrix} r_{i(i+1)} \\ \varrho_{i(i+1)} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sin\left(\alpha_{(i+1)} - \alpha_i\right)} \cdot \sqrt{a_i^2 + a_{(i+1)}^2 - 2\, a_i\, a_{(i+1)}\, \cos\left(\alpha_{(i+1)} - \alpha_i\right)} \\ \arctan\left(\frac{a_{(i+1)} \cdot \cos(\alpha_i) - a_i \cdot \cos\left(\alpha_{(i+1)}\right)}{a_i \cdot \sin\left(\alpha_{(i+1)}\right) - a_{(i+1)} \cdot \sin(\alpha_i)}\right) \end{bmatrix} \tag{4.13}$$

As there is no complexity added compared to line extraction, the algorithm for corner extraction has the same order of $O\left(N^2\right)$.

4.5 Discussion

As the present work focuses on indoor environments, it is legitimate to assume that surroundings are of polygonal shape, i. e. walls and other vertical boundaries are straight and smooth, and if these structures meet or intersect a rather sharp (but not necessarily a 90°) angle will be formed. Small or large scale round features are quite unusual, receiving only a small share in the discernible structures. These assumptions can easily be validated by inspection when looking at the scans delivered by the PLRF. This implicates that geometric primitives like line segments or corners are well suited to match the surroundings and that these features are therefore qualified to approximate the run of the scan points.

Due to the proper fit of features with surrounding structure, the data reduction achieved by feature extraction can easily reach two orders of magnitude, e. g. 1000 scan points versus 10 extracted line segments. Comparing this to filtering techniques, it becomes evident that median filtering provides no data reduction at all, projection filtering being dependent on the view point does not guarantee any reduction, but only reduction filtering assures an actual decrease in the amount of data. However, feature extraction by far exceeds the performance of a reduction filter, which only reaches about one order of magnitude, e. g. 1000 scan points before versus 100 scan points after applying. In addition, feature extraction achieves data reduction without a significant loss of information: Whereas a reduction filter abolishes any kind of fine structure, which was originally available in the scan, line segment extraction is able to reproduce even the details.

Nevertheless, the question arises if the already mentioned combination of filtering and subsequent feature extraction will yield a benefit. The obvious advantages would be twofold: When a reduction or a projection filter are applied, the number of scan points will be reduced leading to a respective shorter execution time for feature extraction. Applying a median filter will eliminate outliers thereby avoiding additional splitting of line segments. This effects that extracted line segments become longer and fewer.

Considering the shorter execution time, one has to take into account that additional filters imply additional computational effort. A projection filter with its insecure reduction performance and complexity $O(N^2)$, which is in the same range as line extraction itself, is not worth the extra effort if only applied due to speed-up reasons. In case of a reduction filter, where complexity is only $O(N)$, an overall gain remains. But as experiments revealed also the reduction filter is arguable: If the radius R is chosen large, i. e. 5 cm or more, structural details will be irrecoverably lost. If R is small, details may reasonably be preserved, but computational gain becomes marginal.

Assuming a fixed window width W, the median filter only adds computational complexity of $O(N)$. However, it introduces nonlinear distortion, which affects the parameters of the extracted lines especially when W is chosen large, i. e. 7 or more scan points. This opposites to the fact that the wider the window is, the better outliers will be rejected. So, also median filtering is ambiguous and can in fact only be recommended for small W, or if one heavily relies on long line segments.

Finally, one can resume that median and reduction filtering prior to feature extraction are only of benefit, if applied carefully. In the other case, loss and error induced by the filters prevail and even worsen the result. The profit of projection filtering only becomes evident when the view point or the surroundings have essentially changed.

Chapter 5

Scan Matching

In the preceding chapter, filtering and feature extraction techniques were applied on the raw sensory data considering only one panoramic scan at a time. Now the focus is set on how to shift and turn *two* panoramic scans against each other so that they optimally coincide, thereby establishing correspondences between the data of the two respective scans. Furthermore, we want to gain a similarity measure in order to rate how well congruence between the two scans could be achieved.

5.1 Problem Description

As already discussed in section 4.5, line segment extraction is a very promising preprocessing technique, primarily due to its significant data reduction potential and its ability to reproduce even fine structures in a polygonally shaped indoor environment. This is why in contrast to several other approaches [BCF+98, TBB+99, LM97, Gut99], the input data to the scan matching problem are chosen to be preprocessed scans, each holding a sequence of line segments as expressed in equation (4.2).

In order to get a concept of the major problems and limitations of scan matching, a practical example is employed: A mobile platform is approaching a T-intersection of two corridors. After reaching the intersection, it executes a right turn. To simplify considerations, it is assumed that the origin of the platform coordinate system and the origin of the PLRF's coordinate system coincide.

Figure 5.1 shows the platform as a crosshair having one long leg indicating

the vehicle´s forward direction and the path of the platform sampled at positions p_0 through p_3 and plotted into a floor plan of the surroundings. In contrast, figure 5.2 shows the PLRF images from the respective positions after line segment extraction. The illustrated scans are *idealized* as at first only geometrical aspects of scan matching are addressed. The effects of the real sensor and the preprocessing procedure are left aside for the present and will be attended to later.

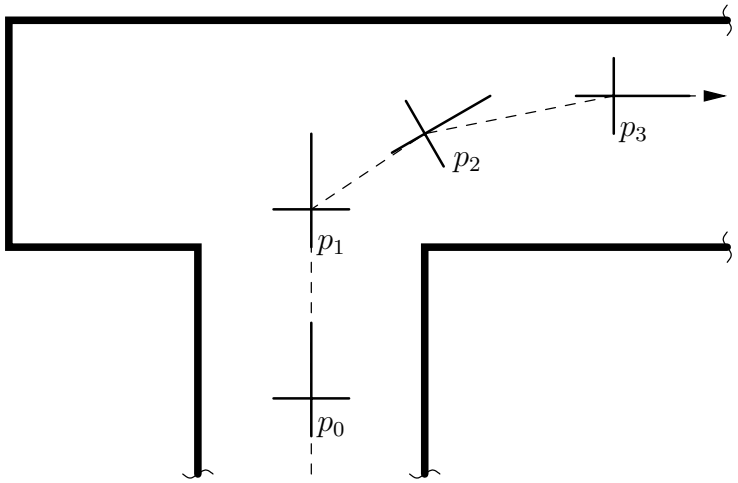


Figure 5.1: Floor plan of surroundings with plotted positions p_0 through p_3.

5.1.1 Geometrical Aspects

When looking at the floor plan with the plotted positions, the human observer may judge the task of scan matching an easy one. However, when viewing the PLRF images in figure 5.2, problems become more evident:

As it is assumed that there is no initial information about the distance travelled between the single positions, PLRF scans can only be represented with respect to the moving coordinate system of the sensor or the platform, respectively. Due to this lack of a global reference that would relate the scans to each other, any difference in position and orientation of corresponding segments is possible.

Although the environment chosen for this example is quite simply structured, there is only one position (p_1) from which all the walls can be seen and extracted as line segments. In every other position walls are at least partially occluded and the available information is reduced.

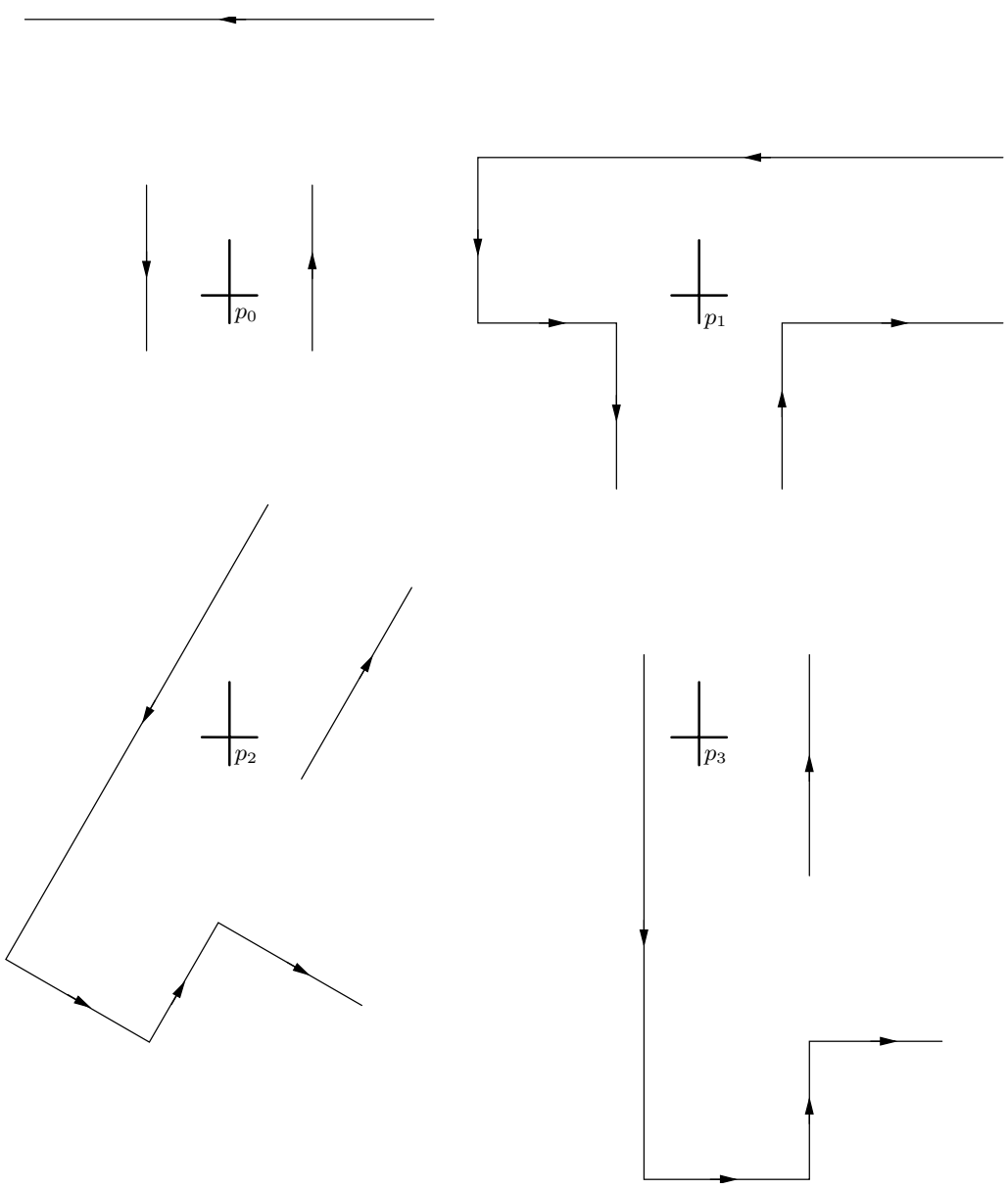


Figure 5.2: Idealized PLRF images after line segment extraction seen from positions p_0 through p_3.

Due to these occlusion effects a correspondence between the length of extracted line segments and the length of corresponding walls can only be established in the way that a line segment can never be longer than the underlying wall. Unfortunately, this already vague statement is rendered completely meaningless when considering that scans are compared to *each other* and that the length of the underlying wall is normally not known.

When comparing the scans taken from position p_0 and p_3, the above mentioned effects become especially adverse: These two scans only have two line segments in common, which differ in position, orientation and length.

The main problem manifesting itself here is that even if two line segments from two different scans originate from the same wall and therefore actually do correspond, *none* of the four parameters by which a line segment is specified proves to be invariant against position changes of the PLRF, i. e. travelling of the platform. It is important to mention that this statement holds true not only for the specification chosen in equation (4.2), but for any parametric specification of line segments.

5.1.2 Ambiguity Problem

Depending on the surroundings, there are cases where scan matching has more than one optimal solution, i. e. when there are several possible mappings between two scans that provide equally optimal matches. This always happens when the PLRF image of the surroundings is ambiguous due to symmetries. For example, if a rectangular room is assumed where only the four walls can be perceived, it is evident that there are always two possibilities with an orientation difference of 180° that provide an optimal match between two PLRF scans. If the room is perfectly square, the number of possibilities even increases to four with an orientation difference of 90°. The ambiguity problem is an inherent defect of scan matching and can therefore be considered an actual limitation, which can only be overcome by employing other sources of information. Usually this is not too serious as the different possibilities suggested by scan matching mostly feature large positional discrepancies so that a rather rough plausibility cross-check against e. g. odometry data will suffice to provide a unique solution.

5.1.3 Effects of Actual Scan Data

Up to this point only idealized PLRF scans were considered. Figure 5.3 gives a clue how an *actual* PLRF scan taken at position p_1 would presumably look like. In grey still visible the extracted line segments of the ideal case, the actual scan shows imperfections in position and orientation of the extracted line segments due to noisy sensory data (cf. section 3.2).

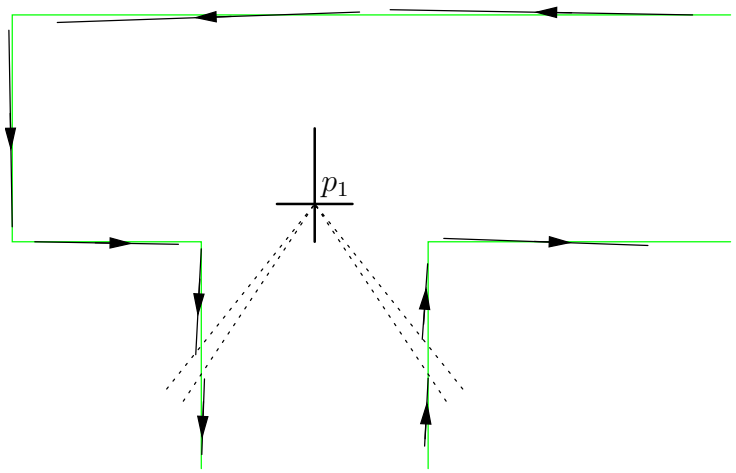


Figure 5.3: Actual PLRF scan from position p_1 after line segment extraction.

Beyond this, one observes that when surroundings form a concave or convex corner, there usually will be a gap between the respective consecutive segments. This results from the fact that especially when considering concrete walls, surfaces in corner regions often are uneven and jagged, which means that sample points taken from there are poorly reflected or even completely deflected (cf. figure 4.5).

The existence of seams or joints even in straight walls is another reason for gaps between segments. Again, the essential property of the surface to provide diffuse reflections may be lost in these areas, and invalid or outlying sample points may be produced. As a consequence, it is well possible that one long straight wall may disintegrate into several consecutive line segments divided by gaps. This already undesired disintegration effect is even worsened as it cannot be considered systematic: In one scan a seam in a straight wall may be harmless, as reflectance is only slightly disturbed. In the next scan, even if position and viewing angle are maintained, reflectance may be seriously affected and the wall will divide into two line segments.

When the range limit of the PLRF is exceeded, poor reflections increasingly

cause gaps and even whole dropout areas in preprocessed scans. Clearly this effect is heavily dependent on the surface the laser beam hits: A smoothly plastered, white painted wall will be recognized from much farther away than a jagged, dark grey concrete face.

Finally, there are two additional, about 5° wide gaps, which are caused by a systematic effect of the employed sensor: The rails supporting the scanning unit of the PLRF produce two shadowed sectors where no information about the surroundings can be gathered (cf. figure 5.3, dotted sectors).

5.1.4 Goals

The primary goal of scan matching is to identify the corresponding pairs of line segments in two different scans, taking into account the above mentioned geometrical aspects and the shortcomings of preprocessed PLRF scans. Secondly, a similarity measure is required that rates the quality of these correspondences and thereby also rates the overall quality of the match. How this is actually achieved and which methods are applied is the subject of the following sections.

5.2 Matching Method

Before presenting an overview over the matching method, a few notations concerning the participating scans are introduced. Without restriction of generality, one of the two preprocessed scans, usually the most recently recorded one, is denoted as *test scan* L_{test}, consisting of m line segments; the other preprocessed scan, usually a previously recorded one, is denoted as *reference scan* L_{ref}, consisting of n line segments. The line segment indices i, k and o denote line segments in the test scan, whereas indices j, l and p denote line segments in the reference scan. Following the notation of equation (4.2), this leads to:

$$\begin{aligned} L_{\text{test}} &= \left\{ \ell_i \;\middle|\; 0 \le i < m \right\} = \left\{ \ell_k \;\middle|\; 0 \le k < m \right\} = \left\{ \ell_o \;\middle|\; 0 \le o < m \right\} \\ L_{\text{ref}} &= \left\{ \ell_j \;\middle|\; 0 \le j < n \right\} = \left\{ \ell_l \;\middle|\; 0 \le l < n \right\} = \left\{ \ell_p \;\middle|\; 0 \le p < n \right\} \end{aligned} \tag{5.1}$$

The search for the corresponding pairs of line segments in two different scans naturally relies on a pairwise comparison of segments ℓ_i against ℓ_j. However, for such a comparison it is inevitable to have at hand at least one invariant

feature for each segment, on which the actual comparison and computation of a similarity measure between ℓ_i and ℓ_j can be based. Unfortunately, as detailed in section 5.1.1, this prerequisite is not fulfilled in the present case.

However, by considering not only ℓ_i, ℓ_j alone, but also the context of these segments, i. e. the remaining other line segments in L_{test} and L_{ref}, this prerequisite can be satisfied and a method for matching two scans can be deduced:

1. Two segments ℓ_i, ℓ_j are randomly chosen.
2. L_{test} is shifted and turned against L_{ref} in that way that line segments ℓ_i and ℓ_j become perfectly aligned. Without going into the details of the similarity measure between line segments, which will be discussed in section 5.3, it can already be understood here that this perfect alignment will maximize the similarity measure between ℓ_i and ℓ_j. Further, by this alignment two of the three degrees of freedom, which are possible in the plane, become bound.
3. To bind the remaining translatory degree of freedom, thereby establishing a common coordinate system between L_{test} and L_{ref}, another pair of line segments ℓ_k, ℓ_l is picked, so that
$$(k, l) \in P_{ij} \setminus \{(i, j)\} \tag{5.2}$$
holds true.

   P_{ij} is the set of tuples identifying the pairs of line segments that are admitted given that line segments ℓ_i, ℓ_j were aligned. The restriction applied here considers the fact that by choosing a tuple (i, j), the angle difference between L_{test} and L_{ref} *before* shift and turn is assumed to $\Delta\alpha_{ij} = (\alpha_i - \alpha_j)$. So it can be concluded that tuples (k, l) with an angle difference $\Delta\alpha_{kl} = (\alpha_k - \alpha_l)$ *after* shift and turn that significantly deviates from 0 cannot represent candidates for corresponding line segment pairs and can therefore be ruled out from the outset.
4. L_{test} is shifted against L_{ref} along the line established by the pair ℓ_i, ℓ_j so that the similarity measure between ℓ_k, ℓ_l also reaches its maximum (cf. section 5.3). Note that this shift does not affect the similarity measure between segments ℓ_i, ℓ_j.
5. After having bound all three degrees of freedom by "locking" line segment pairs ℓ_i, ℓ_j and ℓ_k, ℓ_l, a pairwise comparison of line segment parameters and thus the computation of a pairwise similarity measure between line segments ℓ_o, ℓ_p becomes possible:
$$q_{ijkl}\,(o, p) \geq 0 \quad \text{with} \quad (o, p) \in P_{ij} \tag{5.3}$$

6. By summing up these pairwise similarity measures in a way that
   - a maximum is reached and
   - restrictions implied by the natural order of line segments (cf. section 4.4.1) are respected,

   a local accumulated similarity measure is provided:

$$Q_{\max,ijkl} = \sum_{(o,p)\in P_{\text{opt},ijkl}} q_{ijkl}\,(o,p) \quad \text{with} \quad P_{\text{opt},ijkl} \subseteq P_{ij} \tag{5.4}$$

   How the optimal subset $P_{\text{opt},ijkl}$ is found will be discussed in section 5.5.

7. Repeating steps (3) to (6) for all admitted tuples (k,l) and repeating steps (1) to (6) for all possible tuples (i,j) allows for the global accumulated similarity measure between L_{test} and L_{ref}:

$$Q_{\max} = \max Q_{\max,ijkl} \tag{5.5}$$

   From this, a globally optimal subset P_{opt} can easily be derived by

$$P_{\text{opt}} = \operatorname{argmax} Q_{\max,ijkl} \tag{5.6}$$

   indicating all pairs of line segments ℓ_o, ℓ_p contributing to the best match between L_{test} and L_{ref}. Thus, the subset P_{opt} is likely to represent the actually corresponding pairs of line segments, which have been searched for.

A drawback of the above method is its inherent complexity of $O\,(m^2\,n^2)$, still leaving aside the complexity of step (6), which will be discussed later. Yet it can be notified that the computational effort that has actually to be spent is dramatically lower. Details on this issue are discussed in section 5.5.2.

Figures 5.4 through 5.6 illustrate the whole matching procedure employing the PLRF scans from the above example with the scan taken from position p_0 being the reference scan L_{ref}, and the scan from p_2 being the test scan L_{test}.

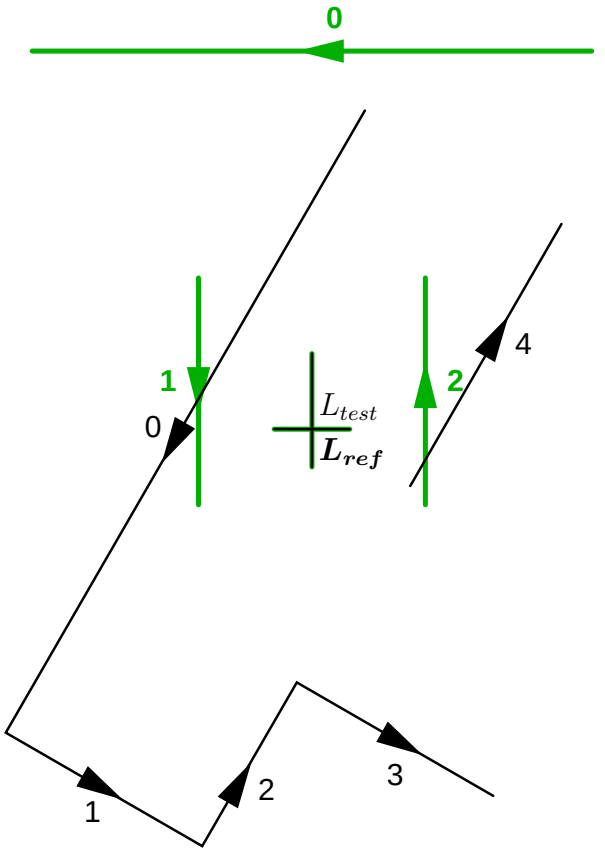


Figure 5.4: L_{ref} and L_{test} drawn on top of each other.

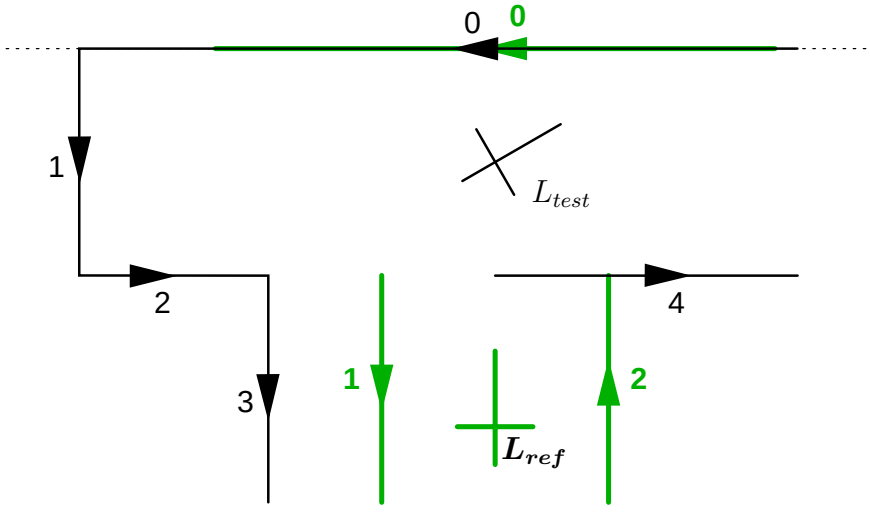


Figure 5.5: Situation after step (2).

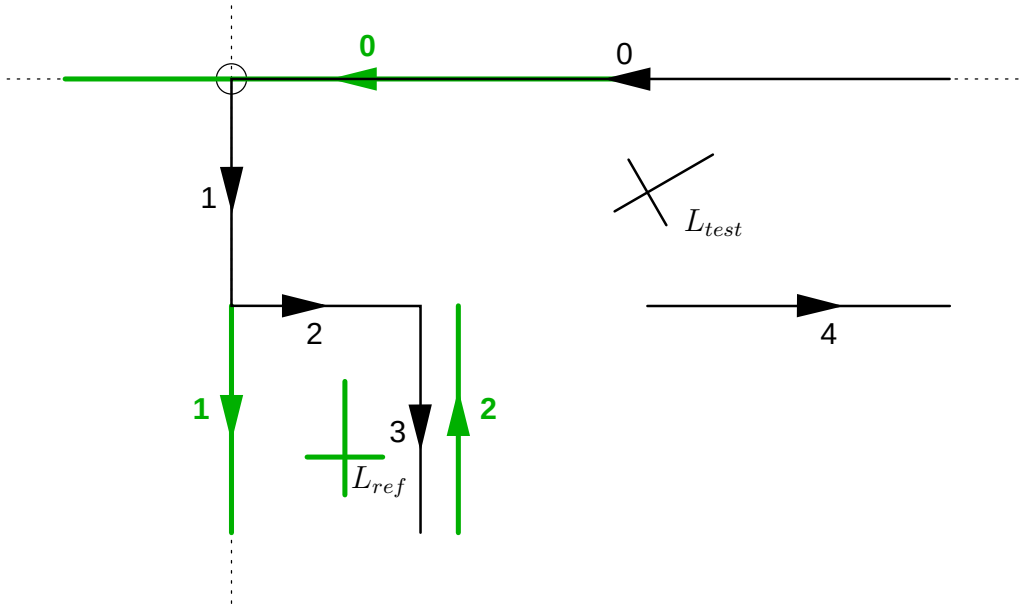


Figure 5.6: Situation after step (4).

In figure 5.4 the two scans are drawn on top of each other using the origin of the PLRF's coordinate system as initial reference.

Figure 5.5 shows the result after step (2) given tuple $(0, 0)$ has been chosen and the appropriate shift and turn operation to align segments and thus maximizing the similarity measure has been performed.

Consequently, in the present case the first set P_{ij} is P_{00} writing as:

$$P_{00} = \{(0, 0), (1, 1), (3, 1)\} \tag{5.7}$$

As can be seen, all other possible tuples show an angle difference of 90° or more after the shift and turn operation. The large angle deviations indicate that these tuples cannot represent candidates for corresponding line segment pairs and are therefore ruled out.

Figure 5.6 shows the situation after step (4) given that tuple $(1, 1)$ has been picked from P_{00} and the respective second shift operation has been finished.

As scans in the present example are idealized, the chosen pairs of line segments denoted by tuples $(0, 0)$ and $(1, 1)$ provide a perfect lock. All three degrees of freedom become bound and a common coordinate system between L_{test} and L_{ref} is established. Also note that the second shift operation did in no way affect the alignment established in step (2). Now computation of

the similarity measure $q_{0011}(o,p)$ with $(o,p) \in P_{00}$ and thus computation of a local accumulated similarity measure Q_{0011} according to step (6) becomes possible.

Obviously, the chosen lock with $(i,j) = (0,0)$, $(k,l) = (1,1)$ will not provide the best match between L_{ref} and L_{test}, i. e. $Q_{0011} \neq Q$. In order to find out the global accumulated similarity measure Q, iterations over all admitted tuples (k,l) and over all possible tuples (i,j) have to be performed according to step (7).

5.3 Similarity Measure Between Line Segments

After having picked two pairs of line segments denoted by tuples (i,j), (k,l) and thus binding all three degrees of freedom in the plane, computation of a pairwise similarity measure $q_{ijkl}(o,p)$ between the two segments identified by the tuple (o,p) is possible. The following paragraphs will illuminate how the similarity measure between line segments is actually computed.

First, there are a few basic requirements that make sense for an expression that is denoted as *similarity measure*:

- Strictly positive, i. e. $q_{ijkl}(o,p) \geq 0$.
- Monotonically increasing, i. e. the more similar the segments, the larger the similarity measure.
- Commutative, i. e. similarity of ℓ_o to ℓ_p is always the same as ℓ_p to ℓ_o.

The term *similarity between line segments* can be equated with the similarity of the descriptive parameters of the line segments. Based on this principle, two different similarity measures will be evolved and discussed.

5.3.1 Similarity Measure Based on Geometrical Parameters

Given one common coordinate system, the similarity between two line segments ℓ_o, ℓ_p, which are given in their geometrical, parametric description as:

$$\begin{aligned} \ell_o &= [a_o, \alpha_o, \varrho_{so}, \varrho_{eo}]^T \\ \ell_p &= [a_p, \alpha_p, \varrho_{sp}, \varrho_{ep}]^T \end{aligned} \tag{5.8}$$

can be rated by the following attributes:

- Differences in straight line parameters $[a, \alpha]^T$, i. e.
  - *distance difference* $\Delta a_{op} = (a_o - a_p)$ and
  - *angular difference* $\Delta\alpha_{op} = (\alpha_o - \alpha_p)$.
- Differences in angle parameters ϱ_s and ϱ_e defining start and end point of a line segment on the underlying straight line.

Due to occlusion (cf. section 5.1.1) and non-systematic disintegration and dropout effects (cf. section 5.1.3), the two angle parameters ϱ_s and ϱ_e are non-deterministic and are therefore not qualified to make any statement about the similarity of two line segments. The straight line parameters $[a, \alpha]^T$, however, are invariant against these factors. A line segment being shorter due to partial occlusion or a segment that has disintegrated e. g. into two parts still maintains its straight line properties. That is why Δa_{op}, $\Delta\alpha_{op}$ are the only attributes which are finally utilized for the geometrically based similarity measure.

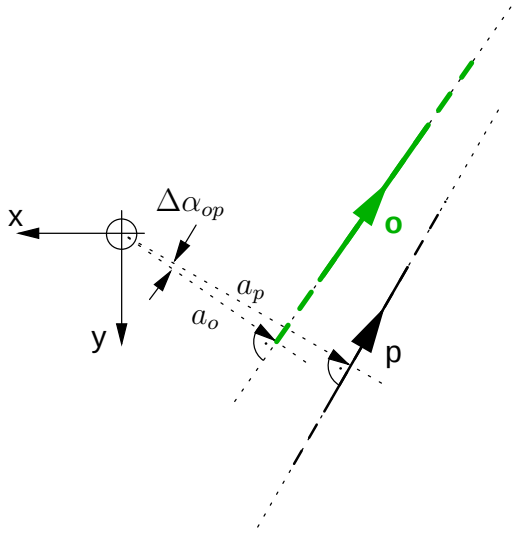


Figure 5.7: Similarity measure between two line segments based on geometrical parameters.

A straightforward notation of the similarity measure writes as:

$$\begin{aligned} q_{ijkl}\,(o,p) &= \frac{1}{1+A^2+B^2} \quad \text{with} \\ A &= c_a \cdot \Delta a_{op} \\ B &= c_\alpha \cdot \Delta \alpha_{op} \\ c_a, c_\alpha &\in \Re^+ \end{aligned} \tag{5.9}$$

If angular and distance differences are 0, the similarity measure reaches its maximum value 1. When differences increase, the similarity measure drops monotonically reaching 0 as asymptotic minimum. On the other hand, if segments become shorter or longer or if a segment is shifted along its underlying straight line, i. e. if there are changes in ϱ_s, ϱ_e, the similarity measure remains the same.

The above notation excels by its simplicity and also features invariance against the unreliable angle parameters ϱ_s, ϱ_e, but it also holds several drawbacks:

- Although there are four descriptive parameters for each line segment, only two parameters prove to be reliable and can actually be employed for the similarity measure.
- If a segment is far away from the origin of the coordinate system, a slight change in angle α implicates a huge change in distance a. This can be denoted as a leverage effect and means that the parameter tuple $[a, \alpha]^T$ becomes ill conditioned when segments are far away.
- The coefficients c_a, c_α providing the weighting between the angular and the distance difference can only be determined heuristically. This is due to the fact that angles and distances, which are independent terms with different units, have to be merged in one expression.

In order to overcome these problems an approach based on a different parameter set is considered next.

5.3.2 Similarity Measure Based on Statistical Parameters

Since an extracted line segment relies on a sequence of scan points, which can also be considered a 2D distribution of scan points, not only the geometrical, but also the statistical parameters, mean vector $\boldsymbol{m}$ and variance-covariance

matrix $\boldsymbol{C}$, are at hand. From $\boldsymbol{C}$, in turn, eigenvalues λ_1, λ_2 and eigenvectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ can be computed (cf. equations (4.6) through (4.10) in section 4.4.1).

From equations (4.10) and (4.11) it can be seen that the information contained in the straight line parameters $[a, \alpha]^T$ is likewise contained in $\boldsymbol{m}$ and $\boldsymbol{v}_1$, $\boldsymbol{v}_2$. However, the eigenvalues λ_1, λ_2, i. e. the variances along the principal axes, actually provide additional information: If there is a large difference between λ_1 and λ_2 with $\lambda_1 > \lambda_2$, the distribution is stretched and slim. This means that the distribution complies well with a line segment and therefore confidence of the segment can be rated high. Conversely, if λ_1, λ_2 show similar values, the distribution of scan points resembles more a blob than a line segment and accordingly confidence is low. This additional information may, of course, also contribute to the similarity measure.

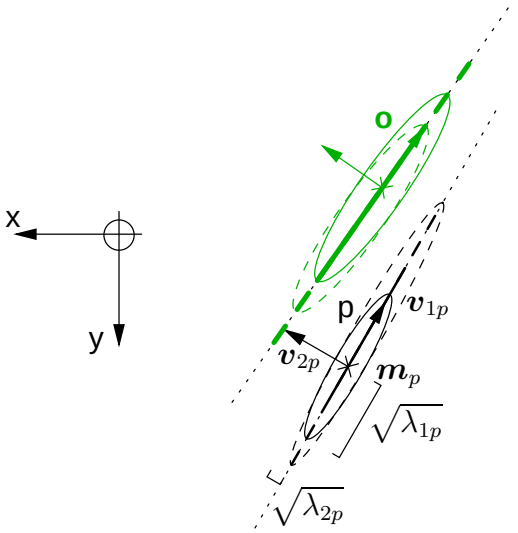


Figure 5.8: Similarity measure between two line segments based on statistical parameters.

When looking at figure 5.8, it seems at first glance that the problem of the similarity between line segments can be put down to the problem of the similarity between two 2D distributions, i. e. the similarity of two 2D probability density functions (PDFs). However, this approach would lead to unsatisfying results: A similarity measure between two PDFs is based on a comparison of the *complete* statistical parameter set of both participating distributions, e. g. Student's t-test considering mean values and variances. Consequently, unreliable statistical parameters $\boldsymbol{m}$ and λ would straightly affect the similarity measure computed on such a basis.

On the other hand, the unreliable angle parameters ϱ_s, ϱ_e inevitably lead to an unreliable mean vector $\boldsymbol{m}$ and eigenvalue λ_1 (illustrated as dashed ellipses in figure 5.8). Along that way unreliability would be propagated from the angle parameters ϱ_s, ϱ_e right to the similarity measure. As this is, of course, highly undesired, a novel similarity measure better adapted for the present problem has to be found.

Although the direct comparison of statistical parameters fails due to the inherent unreliability in $\boldsymbol{m}$ and λ_1, these parameters are, however, not completely useless for the similarity measure: A closer look at the mean vector $\boldsymbol{m}$ reveals that its unreliability solely manifests itself in a possible shift along the underlying straight line, i. e. along the first principal axis $\boldsymbol{v}_1$. Taking this characteristic into account, $\boldsymbol{m}$ could still be employed for the similarity measure. As well, the ratio λ_1/λ_2 definitely makes a statement about the compliance of the 2D distribution with a line segment.

The basic concept of the similarity measure $q_{ijkl}\,(o, p)$ to be developed is that the two distributions of scan points generating the line segments ℓ_o, ℓ_p are joined into one overall distribution, i. e. the union set of the scan points of ℓ_o, ℓ_p (cf. figure 5.9). This procedure is allowed because a common coordinate system is established and the two segments ℓ_o, ℓ_p may be supposed to originate from the same object, e. g. the same wall. Thereby, the properties of ℓ_o, ℓ_p are statistically combined in a virtual line segment ℓ_{op}. The relevant parameters of ℓ_{op} are determined from the overall distribution, i. e. a mean vector $\boldsymbol{m}_{op}$ and a variance-covariance matrix $\boldsymbol{C}^*_{op}$ are computed. The crucial point about these newly introduced parameters is that they can directly be calculated from the existing parameters $\boldsymbol{m}_o$, $\boldsymbol{m}_p$, $\boldsymbol{C}_o$, $\boldsymbol{C}_p$ without having to re-handle the single scan points of ℓ_o, ℓ_p.

The mean vector $\boldsymbol{m}_{op}$ simply is the mean vector of the overall distribution, which computes as:

$$\boldsymbol{m}_{op} = \frac{M_o}{M_o + M_p} \cdot \boldsymbol{m}_o + \frac{M_p}{M_o + M_p} \cdot \boldsymbol{m}_p \tag{5.10}$$

with M_o, M_p being the number of scan points belonging the line segments ℓ_o, ℓ_p.

The matrix $\boldsymbol{C}^*_{op}$ is the weighted mean variance-covariance matrix of $\boldsymbol{C}_o$ and $\boldsymbol{C}_p$, which computes as:

$$\boldsymbol{C}^*_{op} = \frac{M_o}{M_o + M_p} \cdot \boldsymbol{C}_o + \frac{M_p}{M_o + M_p} \cdot \boldsymbol{C}_p \tag{5.11}$$

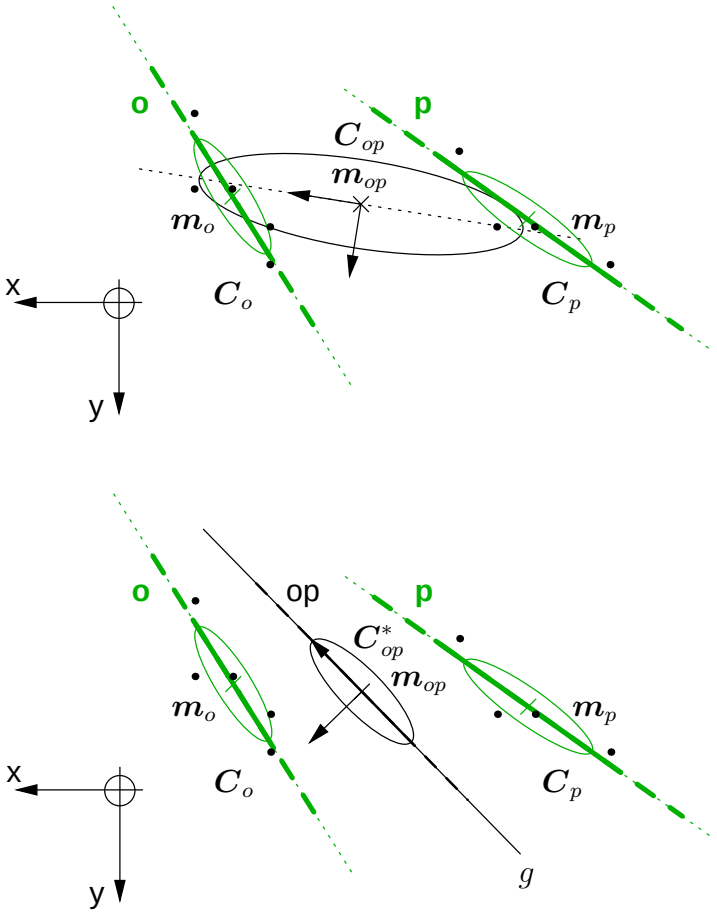


Figure 5.9: Statistical combination of line segments ℓ_o, ℓ_p based on $\boldsymbol{C}_{op}$ with unsatisfying result (a) and $\boldsymbol{C}^*_{op}$ featuring the desired characteristic (b).

As figure 5.9 shows, $\boldsymbol{C}^*_{op}$ is not identical with the variance-covariance matrix $\boldsymbol{C}_{op}$ of the overall distribution. The decisive difference can be seen from the drawings, when the angles of the principal axes, i. e. the respective eigenvectors are considered: As the virtual line segment ℓ_{op} should be a combination of properties of ℓ_o and ℓ_p, the angle of ℓ_{op} is supposed to be somewhere in between the angle of ℓ_o and ℓ_p. The eigenvectors of $\boldsymbol{C}_{op}$, $\boldsymbol{v}_{1,op}$, $\boldsymbol{v}_{2,op}$ do not fulfill this prerequisite, as their angles also depend on the positions of the mean vectors $\boldsymbol{m}_o$, $\boldsymbol{m}_p$ (cf. figure 5.9 a). The eigenvectors of $\boldsymbol{C}^*_{op}$, $\boldsymbol{v}^*_{1,op}$, $\boldsymbol{v}^*_{2,op}$ only rely on $\boldsymbol{C}_o$, $\boldsymbol{C}_p$ and therefore feature the desired characteristic. The underlying straight line of the virtual line segment ℓ_{op} is specified by $\boldsymbol{v}^*_{1,op}$ and $\boldsymbol{m}_{op}$ and will be denoted as straight line g (cf. figure 5.9 b).

As a shift of the mean vectors $\boldsymbol{m}_o$, $\boldsymbol{m}_p$ along the first principal axes $\boldsymbol{v}_{1,o}$, $\boldsymbol{v}_{1,p}$ should ideally have no effect on the similarity measure, the portion of the displacement between $\boldsymbol{m}_o$, $\boldsymbol{m}_p$ *parallel* to g is not considered. What actually determines the similarity measure $q_{ijkl}\,(o,p)$ are the *perpendicular* distances of $\boldsymbol{m}_o$, $\boldsymbol{m}_p$ to the straight line g, i. e. b_{go}, b_{gp}, together with the variance-covariance matrices $\boldsymbol{C}_o$, $\boldsymbol{C}_p$.

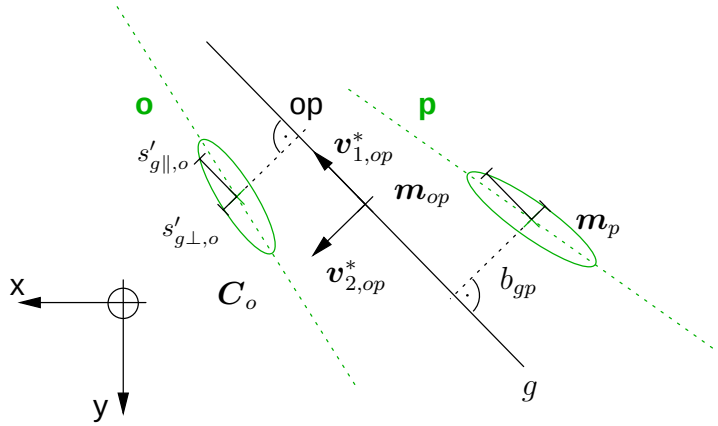


Figure 5.10: Calculation of the similarity measure $q_{ijkl}\,(o,p)$.

Thus, the terms of interest are the variances ${s_{g\perp,o}}^2$, ${s_{g\perp,p}}^2$ of ℓ_o, ℓ_p perpendicular to the straight line g, as well as the variances ${s_{g\parallel,o}}^2$, ${s_{g\parallel,p}}^2$ of ℓ_o, ℓ_p parallel to g. Calculation of these terms involves principal axis transformation of $\boldsymbol{C}_o$, $\boldsymbol{C}_p$ and application of the parallel axis theorem (cf. figure 5.10).

Based on the orientation of g, which is expressed by the eigenvectors of $\boldsymbol{C}^*_{op}$, the variances ${s'_{g\perp,o}}^2$, ${s'_{g\parallel,o}}^2$, ${s'_{g\perp,p}}^2$, ${s'_{g\parallel,p}}^2$ of ℓ_o, ℓ_p with respect to $\boldsymbol{m}_o$, $\boldsymbol{m}_p$ are calculated by a principal axis transformation:

$$\begin{bmatrix} {s'_{g\parallel,o}}^2 & * \\ * & {s'_{g\perp,o}}^2 \end{bmatrix} = \boldsymbol{V}^{*^T}_{op} \cdot \boldsymbol{C}_o \cdot \boldsymbol{V}^*_{op}$$
$$\begin{bmatrix} {s'_{g\parallel,p}}^2 & * \\ * & {s'_{g\perp,p}}^2 \end{bmatrix} = \boldsymbol{V}^{*^T}_{op} \cdot \boldsymbol{C}_p \cdot \boldsymbol{V}^*_{op} \qquad (5.12)$$

$$\text{with} \quad \boldsymbol{V}^*_{op} = \begin{bmatrix} \boldsymbol{v}^*_{1,op} & \boldsymbol{v}^*_{2,op} \end{bmatrix}$$

Applying the parallel axis theorem ("Satz von Steiner") the searched vari-

ances are:

$$\begin{aligned} {s_{g\parallel,o}}^2 &= {s'_{g\parallel,o}}^2 \\ {s_{g\perp,o}}^2 &= {s'_{g\perp,o}}^2 + \frac{M_o}{M_o - 1} \cdot {b_{go}}^2 \\ {s_{g\parallel,p}}^2 &= {s'_{g\parallel,p}}^2 \\ {s_{g\perp,p}}^2 &= {s'_{g\perp,p}}^2 + \frac{M_p}{M_p - 1} \cdot {b_{gp}}^2 \end{aligned} \tag{5.13}$$

with distances b_{go}, b_{gp} between g and $\boldsymbol{m}_o$, $\boldsymbol{m}_p$:

$$\begin{aligned} b_{go} &= \boldsymbol{v}_{2,op}^{*T} \cdot (\boldsymbol{m}_o - \boldsymbol{m}_{op}) \\ b_{gp} &= \boldsymbol{v}_{2,op}^{*T} \cdot (\boldsymbol{m}_p - \boldsymbol{m}_{op}) \end{aligned} \tag{5.14}$$

In the following the ratios $r_o = \frac{s_{g\perp,o}}{s_{g\parallel,o}}$ and $r_p = \frac{s_{g\perp,p}}{s_{g\parallel,p}}$ are considered. The smaller these ratios are, the better is the compliance of the line segments ℓ_o, ℓ_p with the virtual line segment ℓ_{op} lying on g and the larger the similarity measure should become.

This reasoning leads to the final notation of the similarity measure:

$$\begin{aligned} q_{ijkl}\,(o,p) &= \begin{cases} 1 - \dfrac{\max\,(r_o, r_p)}{r_{th}} & \text{if } \max\,(r_o, r_p) < r_{th}, \\ 0 & \text{otherwise.} \end{cases} \\ \text{with} \quad r_o &= \frac{s_{g\perp,o}}{s_{g\parallel,o}} \\ r_p &= \frac{s_{g\perp,p}}{s_{g\parallel,p}} \end{aligned} \tag{5.15}$$

Again, the theoretical maximum of the similarity measure is 1, although this extremum can only be reached if ℓ_o, ℓ_p are exactly aligned and are perfect line segments i. e. $\lambda_{2,o} = \lambda_{2,p} = 0$. The minimum value of 0 is reached whenever one of the two ratios exceeds the threshold value r_{th}.

The introduction of r_{th} together with the piecewise definition of the similarity measure serves an additional purpose: Not only that $q_{ijkl}\,(o,p) = 0$ means that the tuple (o,p) does not provide any contribution to the local accumulated similarity measure Q_{ijkl}. Beyond this, it means that the tuple (o,p) should be ruled out from the optimal path P_{opt} as line segments ℓ_o, ℓ_p are too different to get matched.

5.4 Optimal Displacement Between Two Scans

After having introduced a similarity measure between pairs of line segments, the problem inherent in step (4) of the matching procedure has to be addressed: Whereas the initial shift and turn in step (2) between L_{test} and L_{ref} to perfectly align line segments ℓ_i, ℓ_j is a straightforward operation, the second shift operation in step (4) to match line segments ℓ_k, ℓ_l and to provide a common coordinate system for the subsequent steps involves an optimization task. This is due to the fact that after the initial shift and turn ℓ_k and ℓ_l cannot be assumed parallel in the general case. As a consequence, the displacement ξ_{kl} of the second shift between L_{test} and L_{ref} has to be determined by maximizing the similarity measure $q_{ijkl}(o,p)$ with $(o,p) = (k,l)$ or:

$$\xi_{kl} = \underset{(o,p)=(k,l)}{\operatorname{argmax}} q_{ijkl}(o,p) \tag{5.16}$$

The solution for this optimization problem is found by the classical approach: First, the variable ξ_{kl} is introduced in equation (5.14) for b_{gk}, b_{gl} and along that way in the equation for the similarity measure (5.15). Then the first partial derivative $\frac{\partial q}{\partial \xi}$ is calculated and equated with 0.

Omitting intermediate steps, the final solution for ξ_{kl} reads as:

$$\xi_{kl} = (\hat{x}_k - \hat{x}_l) + (\hat{y}_k - \hat{y}_l) \cdot \left[\frac{A}{B} + sgn(B) \cdot \sqrt{1 + \left(\frac{A}{B}\right)^2} \right]$$

$$\begin{aligned} \text{with} \quad A &= (M_k - 1)(s_{xx,k} - s_{yy,k}) + (M_l - 1)(s_{xx,l} - s_{yy,l}) \\ B &= -2 \cdot [(M_k - 1)\, s_{xy,k} + (M_l - 1)\, s_{xy,l}] \\ \boldsymbol{m} &= \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \\ \boldsymbol{C} &= \begin{bmatrix} s_{xx} & s_{xy} \\ s_{yx} & s_{yy} \end{bmatrix} \end{aligned} \tag{5.17}$$

The plausibility of the above equation can be verified by assuming a special boundary condition: If $\boldsymbol{m}_k$, $\boldsymbol{m}_l$ only differ in the x-coordinate, i. e. $(\hat{y}_k - \hat{y}_l) = 0$, the displacement is always $\xi_{kl} = (\hat{x}_k - \hat{x}_l)$ independent of the (co-)variances of ℓ_k, ℓ_l. In this case $\boldsymbol{m}_k$, $\boldsymbol{m}_l$ and $\boldsymbol{m}_{kl}$ coincide after the shift. Consequently, b_{gk}, b_{gl} are 0, i. e. the portion of ${s_{g\perp,k}}^2$, ${s_{g\perp,l}}^2$ that is

attributed to the parallel axis theorem is eliminated. As ${s'_{g\perp,k}}^2$, ${s'_{g\perp,l}}^2$ as well as ${s_{g\parallel,k}}^2$, ${s_{g\parallel,l}}^2$ are invariant against a shift, the maximum of the similarity measure is reached by employing this displacement.

The general graphical interpretation of equation (5.17) is that after the shift $\boldsymbol{m}_k$, $\boldsymbol{m}_l$ (and then automatically $\boldsymbol{m}_{kl}$) have to lie on the straight line g. Only along that way the portions of the variances perpendicular to g that are attributed to the parallel axis theorem can be eliminated. This, in fact, realizes the whole optimization, as all other terms are invariant against a shift (cf. figure 5.11).

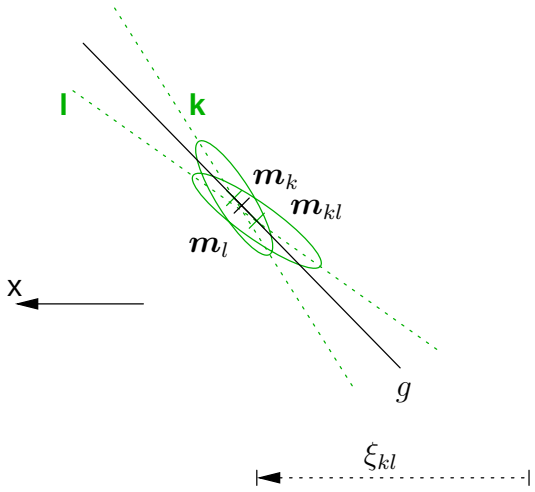


Figure 5.11: Optimal displacement ξ_{kl} between ℓ_k, ℓ_l.

The displacement ξ_{kl} as detailed in equation (5.17) also realizes another optimization: It can be shown that after the shift the second eigenvalue $\lambda_{2,kl}$ of the variance-covariance matrix of the overall distribution $\boldsymbol{C}_{kl}$ also reaches its minimum. This means that after this optimal shift the union set of the scan points of ℓ_k, ℓ_l features its best possible compliance with a line segment, i. e. it is optimally stretched and slim. Further, the eigenvectors of $\boldsymbol{C}_{kl}$, $\boldsymbol{v}_{1,kl}$, $\boldsymbol{v}_{2,kl}$, are identical to the eigenvectors of the weighted mean variance-covariance matrix $\boldsymbol{C}^*_{kl}$, $\boldsymbol{v}^*_{1,kl}$, $\boldsymbol{v}^*_{2,kl}$ (cf. figure 5.12).

These two properties put together supply a subsequent justification that the weighted mean variance-covariance matrix $\boldsymbol{C}^*_{op}$ (with $(o,p) = (k,l)$ in this case) and the derived straight line g do not only match a rather heuristically set up characteristic, but in fact represent the mathematical optimum to statistically combine two line segments ℓ_o, ℓ_p to the virtual line segment ℓ_{op}

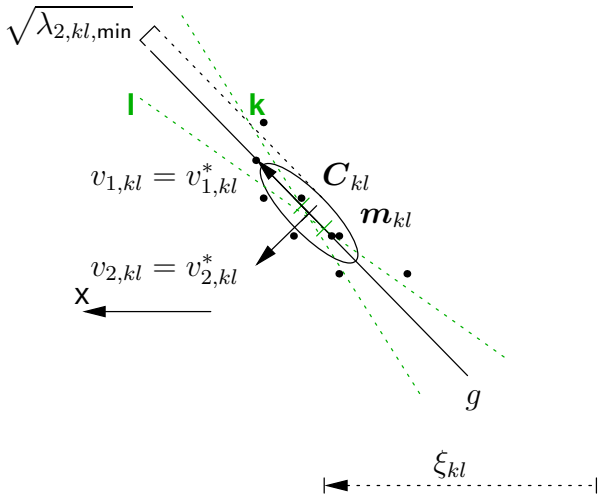


Figure 5.12: Optimal displacement ξ_{kl} also realizes the minimum of the second eigenvalue $\lambda_{2,kl}$ of the variance-covariance matrix of the overall distribution $\boldsymbol{C}_{kl}$. The distributions used here are the same as those in figures 5.9 through 5.11.

under the conditions of the present case.

By the determination of ξ_{kl} the remaining degree of freedom is bound and a common coordinate system for the subsequent steps of the matching procedure is at hand.

5.5 Assignment Of Line Segments

On the basis of a common coordinate system and a similarity measure that is able to rate a match between two line segments, it is now to illuminate how correspondences between line segments are actually established and how the best possible combination of correspondences between line segments of L_{test} and L_{ref} can be found. Related to this topic, but going one step further, is the question which reference scan is regarded the most similar to a given test scan under the condition that not only one reference scan, but a whole set of reference scans

$$\mathcal{L}_{\text{ref}} = \left\{ L_{\text{ref},t} \;\middle|\; 0 \leq t < T \right\} \tag{5.18}$$

is given.

In order to solve these tasks an algorithm originating from pattern recognition, known under the designation *DP-Algorithm* ("Dynamic Programming") is employed. The DP-Algorithm having its roots in the year 1962 [BD62] has since then, together with its derivatives proven its capacity and efficiency in a widespread field of pattern recognition applications, such as continuous speech recognition [PEBR94, Ein93], recognition of (sub-)strings in genetic code [GT93, GT96], object or silhouette or shape recognition in image processing [BHEF98, TY85, Mae90, Mae90] etc. .

To get a notion of the functioning of the DP-Algorithm the following section will give a general outline. The variables utilized in the formal notations intentionally correspond with variables already known in the context of scans and line segments. This analogy means no restriction of generality to the description of the DP-Algorithm and on the other hand helps to illustrate how the present problem of establishing correspondences between line segments can be tackled using the DP-Algorithm.

5.5.1 The DP-Algorithm

In the simplest case the input data for the DP-Algorithm are two pattern sequences L_{test} and L_{ref} of arbitrary length. The pattern sequences have to be discretized into single symbols ℓ_o, ℓ_p, so that each of the pattern sequences consists of a certain finite number of symbols.

$$\begin{aligned} L_{\text{test}} &= \left\{ \ell_o \;\middle|\; 0 \leq o < m \right\} \\ L_{\text{ref}} &= \left\{ \ell_p \;\middle|\; 0 \leq p < n \right\} \end{aligned} \tag{5.19}$$

The symbols ℓ_o, ℓ_p may be scalar or vector terms adopting analog or digital values. The only mandatory prerequisite for ℓ_o, ℓ_p is that a real-valued *distance function*

$$d\left(o,p\right) = d\left(\ell_o, \ell_p\right) \quad \text{with} \quad d\left(o,p\right) \in \Re_0^+ \tag{5.20}$$

is defined between them.

Given the two sequences and the distance function $d\left(o,p\right)$, the DP-Algorithm performs two tasks: Firstly, it gives the minimum accumulated distance D_{min} between the sequences L_{test} and L_{ref} according to the distance function and additional boundary conditions. Secondly, it supplies a set P_{opt} containing those tuples (o,p) that contribute to D_{min}.

These two results of the DP-Algorithm can be expressed by the equation

$$D_{\min} = \sum_{(o,p)\in P_{\text{opt}}} d\,(o,p) \tag{5.21}$$

and are illustrated in figure 5.13. Due to obvious reasons the set P_{opt} is often denoted as the *optimal path*.

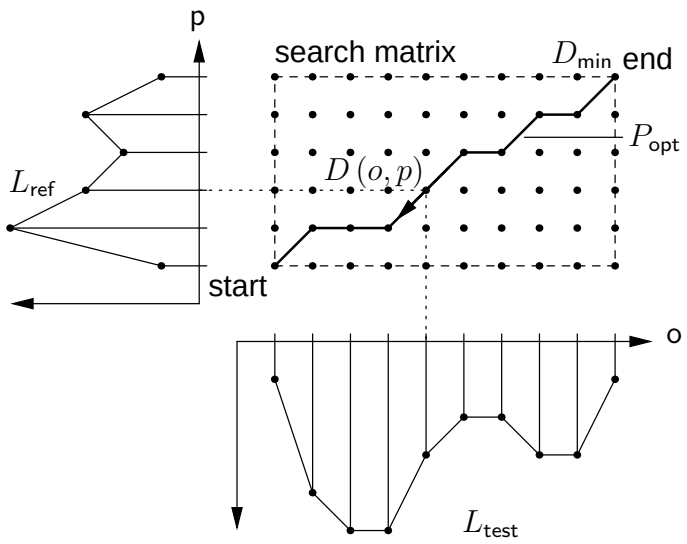


Figure 5.13: Comparing pattern sequences using the DP-Algorithm

$D_{\min}$ and P_{opt} are found by a two-stage procedure: The first stage is called the *forward search*. In the example of figure 5.13, the forward search starts in the lower left corner (*start*) and ends in the upper right corner (*end*) of the *search matrix*. This matrix is spanned by the $(m \cdot n)$ possible combinations of symbols ℓ_o, ℓ_p of the pattern sequences L_{test} and L_{ref}. The forward search consecutively processes all elements of the search matrix from bottom to top and from left to right.

The operation applied to each matrix element is denoted as *local recombination* and is based on a *local DP transition diagram* and an associated recursive *DP equation*. Typical appearances of these, valid for the example

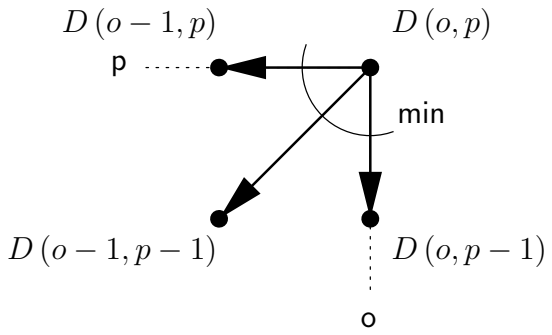


Figure 5.14: Typical local DP transition diagram

of figure 5.13 can be seen from figure 5.14 and equation (5.22).

$$D(o,p) = \min \begin{cases} D(o-1,p) & + & d(o,p) \\ D(o-1,p-1) & + & 2 \cdot d(o,p) \\ D(o,p-1) & + & d(o,p) \end{cases} \tag{5.22}$$

$$\text{with} \quad D(0,0) = d(0,0)$$

The purpose of the local recombination is to find for each matrix element, i. e. tuple (o,p), the minimum accumulated distance $D(o,p)$ from the start point to the current position (o,p) in the matrix. In addition, for each (o,p) the current transition path that was chosen to fulfill the minimum condition is memorized (in the present case this is the diagonal transition path).

After having finished the forward search, i. e. after having reached the end point, $D_{\min}$ is successfully determined as the following equation holds true:

$$D_{\min} = D_{\text{end}} = D(m-1, n-1) \tag{5.23}$$

The second stage called *backtracking* is necessary to obtain the set P_{opt}. Starting from the end point the memorized path transitions are traced back until the start point is reached. While doing that, the respective transition paths are one by one pieced together to finally make up the complete path P_{opt} (cf. figure 5.13).

The previously mentioned boundary conditions are on the one hand the choice of the local DP transition diagram together with the DP equation, on the other hand the selection of start and end point for the forward search. These boundary conditions are besides the distance function crucial for P_{opt}:

The first condition affects the trace of the optimal path. As it will be shown, this can be used to introduce topological restrictions. The second condition pinpoints two tuples in the search matrix that will definitely be part of P_{opt}.

5.5.2 Actual Application Of The DP-Algorithm

After the introduction to the DP-Algorithm, the close relation between the scan matching problem and the sort of problems the DP-Algorithm can solve becomes evident. Nevertheless, some modifications to the basic algorithm are necessary and boundary conditions have to be appropriately chosen to make it applicable for scan matching. These problem specific adaptions will be discussed in the present section.

Probably the most remarkable difference is that in the description of the DP-Algorithm the distance functions d and D were introduced, whereas in the specification of the scan matching problem the similarity measures q_{ijkl} and Q_{ijkl} were used. To adapt for this difference one has just to exchange variable names and to replace the min-decision by a max-decision in the DP-equation.

The reason for this modification is revealed when having a closer look at the properties of the optimal path P_{opt}: In case of a distance function, P_{opt} will be as short as possible, i. e. it will contain as few tuples as possible, since each additional tuple in the path adds an extra distance to the accumulated distance $D(o,p)$, which is to be minimized. Consequently, it easily happens that a tuple (o,p) is ruled out from P_{opt} even if the associated distance $d(o,p)$ is small.

In contrast, if a similarity measure is used, $P_{\text{opt},ijkl}$ will be as long as possible, i. e. it will contain as many tuples as possible, since each additional tuple adds an extra similarity to the local accumulated similarity measure $Q_{ijkl}(o,p)$, which is to be maximized. This consequences in the fact that a tuple (o,p) may remain in $P_{\text{opt},ijkl}$ even if $q_{ijkl}(o,p)$ is rather bad.

For the local DP transition diagram of figure 5.14 this would mean that if a distance function d were employed, the diagonal transition would be preferred in order to shorten the optimal path. In case of a similarity measure q_{ijkl}, the horizontal and vertical transitions would preferably be chosen in order to make the optimal path as long as possible.

Though one can try to compensate for this effect by adjusting the weighting of the different transitions (cf. DP-equation (5.22)), the better choice is to a-priori decide whether a long or a short optimal path is preferred for the

present task. As in the case of scan matching the goal is to map as many line segments as possible between L_{test} and L_{ref}, the use of a similarity measure q_{ijkl} is chosen.

As a method to avoid the above mentioned effect that tuples (o, p) with bad similarity measure $q_{ijkl}(o, p)$ are part of $P_{\text{opt},ijkl}$, it is possible to define a lower bound for the similarity measure. If this bound is reached, the associated tuple (o, p) is ruled out. The similarity measure proposed in equation (5.15) features this characteristic, establishing 0 as threshold value.

Ruling out tuples (o, p) from the outset, be it due to the above mentioned similarity measure restriction or due to the angular restriction detailed in section 5.2 step (3), is in fact an intervention in the procedure of the DP-Algorithm. Though this can be tolerated, it leads to dropouts in the search matrix, i. e. matrix elements are missing on certain positions. The local DP-transition diagram has to take these dropouts into account, in order to avoid that the forward search comes to a premature termination due to a lack of reachable matrix elements. Hence, the actually applied local DP transition diagram (cf. figure 5.15) and associated DP equation look like the following:

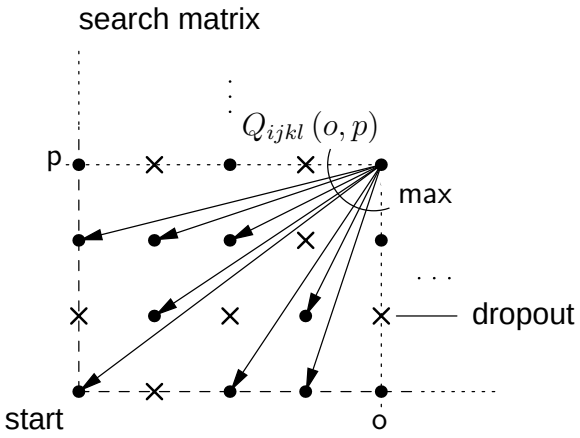


Figure 5.15: Actually applied local DP transition diagram

$$Q_{ijkl}(o, p) = \max\left\{Q_{ijkl}(o - r, p - s) + q_{ijkl}(o, p) \;\middle|\; 0 < r \leq o, 0 < s \leq p\right\}$$

$$\text{with} \quad Q_{ijkl}(0, 0) = q_{ijkl}(0, 0) \tag{5.24}$$

At first glance, the above procedure appears to be very costly, as all "pre-

decessors" of a tuple (o, p) have to be considered. In practice, however, the actual effort is much lower due to the numerous dropouts that occur and list-based implementation that avoids any computational overhead for the missing tuples.

Although the employed DP transition diagram has many degrees of freedom, it implies the topological restriction of preserving the natural order of the line segments ℓ_i, ℓ_j in L_{test} and L_{ref}. This can be seen from the strictly positive slope of the DP transition paths in figure 5.15.

The next issue to deal with are the start and end point for the forward search. Here it becomes effective that the sequences under consideration, L_{test} and L_{ref}, are not linear, but cyclic, i. e. they repeat with a modulus of m and n, respectively.

$$\ell_i = \ell_{(i+\nu \cdot m)} \quad \text{and} \quad \ell_j = \ell_{(j+\nu \cdot n)} \tag{5.25}$$

Since L_{test} and L_{ref} can be arbitrarily turned against each other, the consequence is that all $(m \cdot n)$ tuples (i, j) of the search matrix have to be tested as start point for the forward search (cf. figure 5.16) inducing a respective re-sorting of the search matrix.

The determination of the end point for the forward search also reveals a difference to the basic algorithm, as a definite end point cannot a-priori be identified. Instead of the end point, a second tuple $(k, l) \in P_{ij} \setminus \{(i, j)\}$ (corresponding to the tuple (k, l) in section 5.2 step (3)) is provided as intermediate target that is defined to be part of the optimal path. Due to the restrictions implied by the local DP transition diagram, which only allows for positively sloped transition paths, only two subspaces

$$\begin{aligned} R_1 &= \big\{(o, p) \;\big|\; 0 \leq o < k, 0 \leq p < l\big\} \\ R_2 &= \big\{(o, p) \;\big|\; k < o < m, l < p < n, (o, p) = (k, l)\big\} \end{aligned} \tag{5.26}$$

of the search matrix touching each other in the tuple (k, l) turn out to be relevant and have to be actually considered (cf. figure 5.16). The rest of the matrix can be ignored, as the optimal path will in no case enter the other regions (shaded areas in figure 5.16).

The actual end point is now determined online during forward search as that tuple (o, p) in subspace R_2 having the highest accumulated similarity measure $Q_{ijkl}(o, p)$, i. e.

$$Q_{\text{end},ijkl} = Q_{\max,ijkl} = \max\big\{Q_{ijkl}(o, p) \;\big|\; (o, p) \in R_2\big\} \tag{5.27}$$

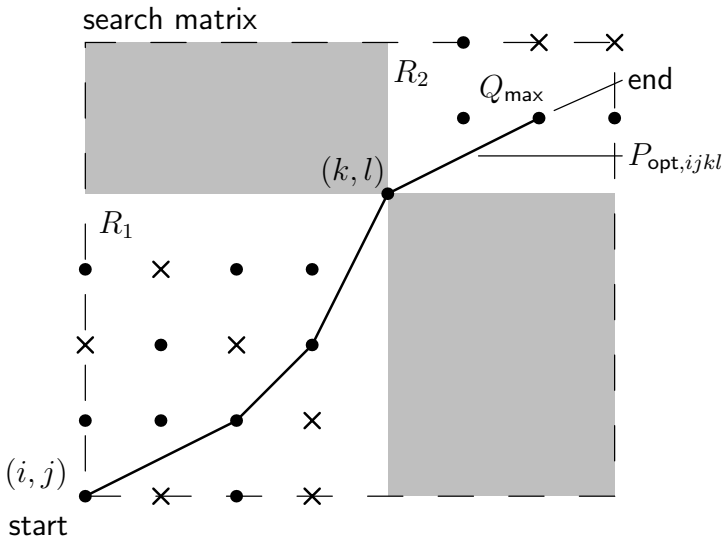


Figure 5.16: Applied forward search scheme

The final result, i. e. the pairs of line segments ℓ_o, ℓ_p contributing to the best match between L_{test} and L_{ref} can now be determined according to equations (5.5) and (5.6) in section 5.2 step (7).

As the DP-Algorithm is basically not intended for cyclic pattern sequences and intermediate targets, the above mentioned results in $(m \cdot n)$ iterations of the DP-Algorithm due to the unknown start point (i, j) and another $(m-1) \cdot (n-1)$ iterations due to the varying intermediate target (k, l). This leads to the $O\left(m^2\, n^2\right)$ iterations already mentioned in section 5.2 for one complete comparison between L_{test} and L_{ref}.

During the DP-Algorithm itself $(m \cdot n)$ max-decisions inducing a maximum of $(m \cdot n)$ tuple inspections each are performed. This means a complexity of $O\left(m^2\, n^2\right)$ for the DP-Algorithm alone, under the condition that a tuple inspection, is regarded as the basic operation. (Computational complexity of the backtracking procedure can be neglected, as it is $O\left(n\right)$.)

Consequently, the overall complexity of a scan matching procedure is given by $O\left(m^4\, n^4\right)$ tuple inspections with a tuple inspection being equal to one calculation of the similarity measure $q_{ijkl}\left(o, p\right)$ according to equation (5.15).

However, complexity considerations are worst case scenarios that do not necessarily make a correct statement about the effort that has to be actually spent. Fortunately, this is the case here, and the consequences of the tremendous complexity for long pattern sequences L_{test} and L_{ref} do not become

effective.

The main reason for this is that especially large search matrices are sparsely occupied, i. e. the number of dropouts increases faster than the number of active matrix elements that have to be considered. In an extreme example this could mean that there is only a narrow channel of active matrix elements, while the predominant majority of matrix positions contains dropouts.

In addition, two effort saving measures are introduced: As illustrated in figure 5.16 not the whole search matrix, but only the two subspaces R_1, R_2 have to be processed. It can easily be seen that the effectiveness of this measure depends on the position of the tuple (k, l) and varies between saving half of the effort and zero. Further, there is practically no difference between $P_{\text{opt},ijkl}$ and $P_{\text{opt},klij}$, i. e. (i, j) being the start point and (k, l) the intermediate target or (k, l) being the start point and (i, j) the intermediate target mostly leads to the same optimal path. Due to this near-symmetry, another half of the computational effort can be saved.

The final evidence that the statements made above hold true is supplied by practical application: A complete comparison between two scans L_{test} and L_{ref} containing a rather large number of 20 line segments each can be computed in about 100 ms on a commercial PC platform equipped with an Intel Celeron processor operating at a clock rate of 400 MHz.

The existence of a whole set of reference scans $\mathcal{L}_{\text{ref}}$ already mentioned in the early stages of section 5.5 and the inherent problem which reference scan is the most similar to a given test scan L_{test} can now treated straightforwardly.

According to the size of $\mathcal{L}_{\text{ref}}$, T comparisons between L_{test} and $L_{\text{ref},t}$ have to be carried out. The most similar reference scan $L_{\text{ref,best}}$ is the one that reached the highest global accumulated similarity measure, i. e.

$$L_{\text{ref,best}} = \operatorname{argmax} Q_{\text{max},t} \tag{5.28}$$

5.6 Summary

Starting with the extracted line segments as they are supplied from PLRF data preprocessing, the present chapter detailed on establishing pairwise correspondences between the line segments of two scans in such a way that the entire two scans optimally coincide.

As an approach to this task the characteristics of the extracted line segments with the inherent problems were investigated and discussed. In the following,

a matching method being able to cope with the lack of a global reference system was proposed. As a prerequisite for the matching procedure, a pairwise similarity measure between line segments was developed. In this context the statistical parameters of the extracted line segments were employed and the line segments are regarded as 2D probability distributions in order to realize the concept of a similarity measure that is based on the similarity between two 2D PDFs. Due to the characteristics of the distributions, standard techniques to compare the PDFs turned out to be not applicable. This is why a customized similarity measure was evolved, which is on the one hand intuitively derived from the statistical parameters and on the other hand respects the special boundary conditions of the participating distributions.

In order to actually assign line segments to each other, the use of the DP-Algorithm was suggested and several problem specific modifications to it were illustrated.

The proposed matching method leads to the final result of a set of line segment pairs that are supposed to coincide, i. e. to originate from the same object, e. g. the same wall. Considerations on the computational effort of the procedure and a brief description how it can easily be extended to deal with more than one reference scan conclude this chapter.

Chapter 6

Self-Localization and Map Building

6.1 Background

Generally spoken, the task of self-localization asks the question: *Where am I?* A typical answer to this question could be a geometric specification consisting of numbers and units like: *You are in position* 48°08.917′ *N,* 011°34.121′ *E.* Equivalent to this would be a topological description employing locations and spatial relations like: *You are in front of main entrance of the Technische Universität München, Germany.*

From this example it already becomes evident that a definite positional fix can only be gained from these answers, if additional information is at hand: Firstly, for a geometric description a reference system specifying an origin and units has to be defined. For a topological description a set of allowed and understood locations and spatial relations has to be identified, respectively. Secondly, the mere geometric or topological data strings cannot be interpreted and are worthless if there is not a however natured image of the real world being in accordance with the reference system. This image of the world together with the reference system is commonly denoted as *world model* or *environmental map* (cf. section 6.4).

In order to finally enable a system, be it human or technical, to perform self-localization, it must be equipped with at least one appropriate sensor, an at least rudimentary, i. e. in an extreme case empty map and the capability to *orientate*. Orientation means the matching between current sensory data

and map data with a subsequent interpretation that eventually provides a positional fix in the map.

The result of self-localization may be the binding of all existing degrees of freedom, i. e. six degrees of freedom in free space (three translations, three rotations) or three degrees of freedom in the plane (two translations, one rotation). This will be referred to as *complete localization*. If not all degrees of freedom could be bound, e. g. if only one translatory and one rotatory degree of freedom are bound in the plane, this is consequently denoted as *partial localization*.

Further, it is to discern between two different modes of localization. *Absolute localization* gives a positional fix in a global world model with respect to one global coordinate system having a well defined origin. The problem is that such a global coordinate system is normally a-priori not at hand and costly to establish, as in a spacious environment distances to the origin of the global coordinate system may become large and on the other hand should be accurately measured. In addition, local changes in the environment induce considerable effort to consistently incorporate into the global world model, which makes it hard to maintain. The main advantage is that very accurate position fixes even when a mobile system has travelled long distances in its environment will be possible.

Relative localization gives a relative displacement vector between two consecutively taken sensory snapshots. Thus, a global world model with a fixed coordinate system is not required. This characteristic is extremely well qualified for the *exploration* scenario. Per definition, in this case a-priori information about the environment is limited or even nonexisting and the goal is to build an environmental map. A drawback of this localization mode is that localization over long distances is accomplished by piecing together consecutive relative displacement vectors, which leads to an accumulation of positional errors. In practice this means that accuracy is good over short distances, but degrades without bounds, if distances become longer, as a reference to re-synchronize is missing.

A promising approach to merge these two localization modes in order to build an environmental map will be detailed in section 6.4.1.

The following section is about identifying a quantitative displacement vector between the two positions from where the two scans L_{test} and L_{ref} have been taken. According to the above mentioned, this can be considered a geometric relative self-localization.

6.2 Geometric Relative Self-Localization

The result of the matching procedure, which is a set of corresponding line segments P_{opt} between L_{test} and L_{ref}, has now to be transformed into a displacement vector $\Delta\boldsymbol{X} = [\Delta X, \Delta Y, \Delta\Phi]^T$ between the two positions from where these scans have been shot.

As an approach to this task, it is important to recall that an established correspondence between two line segments ℓ_o, ℓ_p indicates that these two line segments are supposed to originate from the same object. This statement allows for a quantitative geometric interpretation: The angular difference $\Delta\alpha_{op} = (\alpha_o - \alpha_p)$ can directly be mapped to the rotatory displacement $\Delta\Phi$. As well, from the distance difference $\Delta a_{op} = (a_p - a_p)$ a quantitative statement to determine the translatory displacements ΔX, ΔY can be gained. Taking into account the geometrical relations illustrated in figure 6.1, this

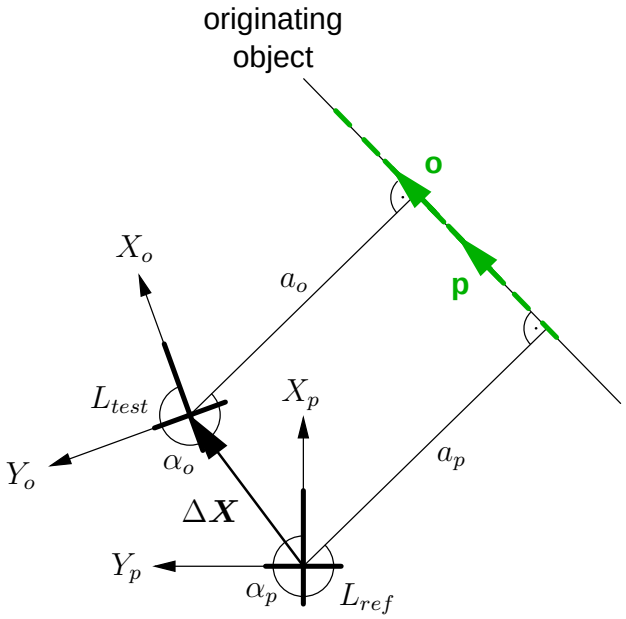


Figure 6.1: Determination of $\Delta\boldsymbol{X}$ taking into account geometrical relations.

leads to a straightforward notation for the displacement vector $\Delta\boldsymbol{X}$, which is given with respect to the coordinate system defined by L_{ref}:

$$\Delta\Phi = (\alpha_p - \alpha_o) = -\Delta\alpha_{op} \tag{6.1}$$

$$\cos\alpha_p \cdot \Delta X + \sin\alpha_p \cdot \Delta Y = (a_p - a_o) = -\Delta a_{op} \tag{6.2}$$

Having a closer look at the originating object reveals that due to the effects of actual scan data (cf. section 5.1.3) α_o, α_p will not exactly coincide although $\Delta\Phi$ is accordingly adjusted. This situation in conjunction with the fact that ℓ_o, ℓ_p are regarded as probability distributions (cf. section 5.3.2) is depicted in figure 6.2 .

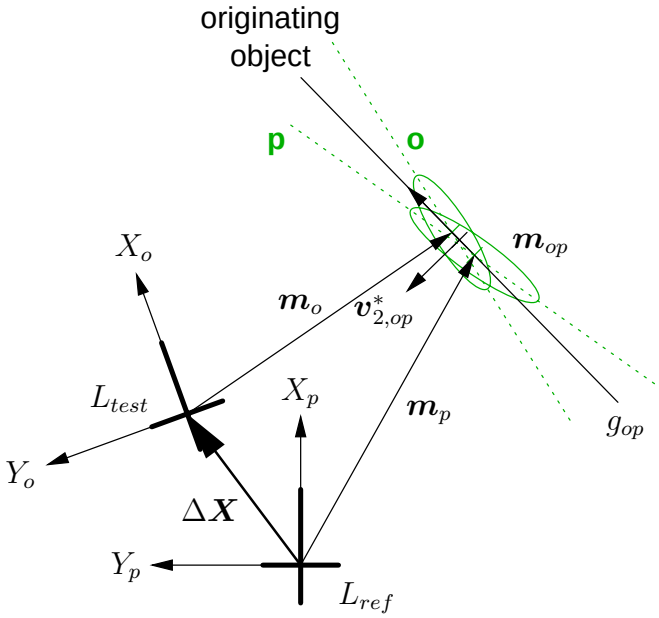


Figure 6.2: More precise determination of $\Delta\boldsymbol{X}$ considering probability distributions.

Due to the concept of a virtual line segment ℓ_{op}, which combines the statistical properties of ℓ_o and ℓ_p, the straight line g_{op} is supposed to represent the actual face of the originating object. Consequently, the parameters of g_{op} will appear in the notation. This results in a change of the second equation to:

$$\begin{aligned} \cos\alpha_{g_{op}} \cdot \Delta X + \sin\alpha_{g_{op}} \cdot \Delta Y &= \boldsymbol{v}^{*^T}_{2,op} \cdot (\boldsymbol{m}_o - \boldsymbol{m}_p) = b_{op} \\ \text{with} \quad \alpha_{g_{op}} &= \arctan\frac{v^*_{2,op,y}}{v^*_{2,op,x}} \\ \boldsymbol{v}^*_{2,op} &= \begin{bmatrix} v^*_{2,op,x} \\ v^*_{2,op,y} \end{bmatrix} \end{aligned} \tag{6.3}$$

As it is evident from equation (6.1), one pairing of line segments ℓ_o, ℓ_p is sufficient to explicitly determine $\Delta\Phi$. In contrast, two equations of type

(6.3), i. e. two pairings of line segments are needed to obtain a linear equation system that can explicitly be solved for the variables ΔX, ΔY.

In general, the set P_{opt} provides not only one or two, but multiple pairings (o, p) each contributing one equation of type (6.1) and one equation of type (6.3). This implicates an overdetermined linear equation system for $\Delta\Phi$ as well as for ΔX, ΔY.

An appropriate method of resolution to an overdetermined linear equation system is minimizing the sum of the squares of the residuals, also known as *least squares* method. However, this approach only provides satisfying results, if it is guaranteed that the input data is free from outliers. This means that spurious angular differences as well as incorrect distance difference equations must be thoroughly eliminated since one single invalid data point may already cause serious distortion. This is assured by the ruling out of unqualified tuples (o, p) as detailed in section 5.2, step (3) and in section 5.3.2 in context of the similarity measure. Hence the requirements for the least squares approach are fulfilled and it will be applied in the following.

In case of the explicit variable $\Delta\Phi$, the least squares method is simply realized by calculating the mean value, i. e.

$$\Delta\Phi = -\widehat{\Delta\alpha} = -\frac{1}{|P_{\text{opt}}|} \cdot \sum_{(o,p)\in P_{\text{opt}}} \Delta\alpha_{op} \tag{6.4}$$

thereby minimizing the variance

$$s_{\Delta\Phi}{}^2 = s_{\Delta\alpha}{}^2 = \frac{1}{|P_{\text{opt}}| - 1} \cdot \sum_{(o,p)\in P_{\text{opt}}} \left(\Delta\alpha_{op} - \widehat{\Delta\alpha}\right)^2 \tag{6.5}$$

with $|P_{\text{opt}}|$ being the number of tuples (o, p) in P_{opt}.

In case of the implicit variables ΔX, ΔY the following generally overdeter-

mined linear equation system has to be solved:

$$\begin{bmatrix} \cos\alpha_{g_0} & \sin\alpha_{g_0} \\ \vdots & \vdots \\ \cos\alpha_{g_\kappa} & \sin\alpha_{g_\kappa} \\ \vdots & \vdots \\ \cos\alpha_{g_{(|P_{\text{opt}}|-1)}} & \sin\alpha_{g_{(|P_{\text{opt}}|-1)}} \end{bmatrix} \cdot \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} = \begin{bmatrix} b_0 \\ \vdots \\ b_\kappa \\ \vdots \\ b_{(|P_{\text{opt}}|-1)} \end{bmatrix}$$

using the abbreviation κ for the tuple $(o,p)_\kappa \in P_{\text{opt}}$

and having the format: $\boldsymbol{A}_{\langle|P_{\text{opt}}|\times 2\rangle} \cdot \Delta\boldsymbol{x}_{\langle 2\times 1\rangle} = \boldsymbol{b}_{\langle|P_{\text{opt}}|\times 1\rangle}$ (6.6)

Formally, the solution to (6.6) is given by:

$$\Delta\boldsymbol{x} = \boldsymbol{A}^{-1} \cdot \boldsymbol{b} \tag{6.7}$$

However, due to its generally non-square format, $\boldsymbol{A}$ is not invertible, i. e. $\boldsymbol{A}^{-1}$ does not exist. A straightforward solution to this problem is offered by the so-called *pseudo-inverse* $\boldsymbol{A}^+$ of matrix $\boldsymbol{A}$.

Without detailing on the proof that the pseudo-inverse actually meets the condition of minimizing the sum of the squares of the residuals [LH74], the following gives an illustration how $\boldsymbol{A}^+$ is found:

$$\begin{aligned} \boldsymbol{A}_{\langle|P_{\text{opt}}|\times 2\rangle} \cdot \Delta\boldsymbol{x} &= \boldsymbol{b} \\ \boldsymbol{A}^T_{\langle 2\times|P_{\text{opt}}|\rangle} \cdot \boldsymbol{A}_{\langle|P_{\text{opt}}|\times 2\rangle} \cdot \Delta\boldsymbol{x} &= \boldsymbol{A}^T_{\langle 2\times|P_{\text{opt}}|\rangle} \cdot \boldsymbol{b} \\ \left(\boldsymbol{A}^T \cdot \boldsymbol{A}\right)_{\langle 2\times 2\rangle} \cdot \Delta\boldsymbol{x} &= \boldsymbol{A}^T_{\langle 2\times|P_{\text{opt}}|\rangle} \cdot \boldsymbol{b} \\ \Delta\boldsymbol{x} &= \left(\boldsymbol{A}^T \cdot \boldsymbol{A}\right)^{-1}_{\langle 2\times 2\rangle} \cdot \boldsymbol{A}^T_{\langle 2\times|P_{\text{opt}}|\rangle} \cdot \boldsymbol{b} \\ \text{with (6.7):}\quad \boldsymbol{A}^+_{\langle 2\times|P_{\text{opt}}|\rangle} &= \left(\boldsymbol{A}^T \cdot \boldsymbol{A}\right)^{-1}_{\langle 2\times 2\rangle} \cdot \boldsymbol{A}^T_{\langle 2\times|P_{\text{opt}}|\rangle} \end{aligned} \tag{6.8}$$

A prerequisite for this approach is that the square matrix $\left(\boldsymbol{A}^T \cdot \boldsymbol{A}\right)$ must not be singular, i. e. the inverse matrix must exist. Although in practical application exact singularity will be an unlikely event, the equally critical case of near-singularity will definitely occur. This results from a numerical ill condition of the matrix $\left(\boldsymbol{A}^T \cdot \boldsymbol{A}\right)$, which consequences in an inaccurate or even useless result for $\Delta\boldsymbol{x}$. Although such an ill condition can be detected, e. g. by checking if the determinant of $\left(\boldsymbol{A}^T \cdot \boldsymbol{A}\right)$ ranges near zero, it cannot be prevented or corrected for.

This is why another procedure for solving (6.6) is preferred. It is less sensitive against numerical ill condition by employing an augmented equation system: This approach considers that the equation system of (6.6) cannot exactly be solved and should therefore rather read as $\boldsymbol{A}\cdot\Delta\boldsymbol{x} \approx \boldsymbol{b}$. In order to re-establish the equal sign the *residual vector* $\boldsymbol{r}$ is added to the original equation, i. e.

$$\boldsymbol{r} + \boldsymbol{A} \cdot \Delta\boldsymbol{x} = \boldsymbol{b} \tag{6.9}$$

The boundary condition that embodies the least squares method is that the residual vector $\boldsymbol{r}$ must be orthogonal to the vector $(\boldsymbol{A} \cdot \Delta\boldsymbol{x})$, i. e.

$$\begin{aligned} (\boldsymbol{A} \cdot \Delta\boldsymbol{x})^T \cdot \boldsymbol{r} &= \boldsymbol{0} \\ \Delta\boldsymbol{x}^T \cdot \boldsymbol{A}^T \cdot \boldsymbol{r} &= \boldsymbol{0} \end{aligned} \tag{6.10}$$

In order to satisfy the above least squares condition, there are several possibilities:

- $(\boldsymbol{A} \cdot \Delta\boldsymbol{x}) = \boldsymbol{0}$: This is generally not true, as $\boldsymbol{A} \cdot \Delta\boldsymbol{x} \approx \boldsymbol{b}$ and $\boldsymbol{b} \neq \boldsymbol{0}$.
- $\boldsymbol{r} = \boldsymbol{0}$: This is only possible if the original equation system (6.6) has an exact solution, which is likewise generally not the case.
- $\Delta\boldsymbol{x} = \boldsymbol{0}$ or $\boldsymbol{A} = \boldsymbol{0}$: These are trivial solutions.
- Or
$$\boldsymbol{A}^T \cdot \boldsymbol{r} = \boldsymbol{0} \tag{6.11}$$
This is the only general solution meeting the least squares condition.

From (6.9) and (6.11) the augmented equation system:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}_{\langle |P_{\text{opt}}| \times 2 \rangle} \\ \boldsymbol{A}^T_{\langle 2 \times |P_{\text{opt}}| \rangle} & \boldsymbol{0} \end{bmatrix}_{\langle (|P_{\text{opt}}|+2) \times (|P_{\text{opt}}|+2) \rangle} \cdot \begin{bmatrix} \boldsymbol{r} \\ \Delta\boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{bmatrix} \tag{6.12}$$

is obtained, which can be solved with standard techniques for the position vector $\Delta\boldsymbol{x} = [\Delta X, \Delta Y]^T$ and the residual vector $\boldsymbol{r}$. Owing to its structure the above matrix is much less sensitive against numerical ill condition. Furthermore, the residual vector $\boldsymbol{r}$ is calculated as a by-product.

6.2.1 Scaling

In an exactly determined linear equation system, i. e. $\boldsymbol{A}$ is square and not rank deficient, multiplying the rows with constant values does not change

the solution vector $\Delta\boldsymbol{x}$. However, in the overdetermined case the solution is affected by this operation. The effect can be used to realize a different weighting of the single equations, in order to express that some equations in the system are more significant than others. This means that the original matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$ are transformed by the diagonal matrix $\boldsymbol{W}$ to:

$$\begin{aligned} \boldsymbol{A}_W &= \boldsymbol{W} \cdot \boldsymbol{A} \\ \boldsymbol{b}_W &= \boldsymbol{W} \cdot \boldsymbol{b} \\ \text{with} \quad \boldsymbol{W} &= \operatorname{diag}\left(w_0, \ldots, w_\kappa, \ldots, w_{(|P_{\text{opt}}|-1)}\right) \end{aligned} \tag{6.13}$$

and w_κ being the weighting coefficients.

The actual result of weighting is that those rows receiving a large weighting coefficient w_κ will show a small residual coefficient r_κ after solution of (6.12) with $\boldsymbol{A}$, $\boldsymbol{b}$ replaced by $\boldsymbol{A}_W$, $\boldsymbol{b}_W$.

Concerning the determination of w_κ, [LH74] and [GL93] suggest to use a measure of the uncertainty inherent in b_κ. Applying this suggestion to the present case means to employ the variances ${s_{g\perp,o}}^2$, ${s_{g\perp,p}}^2$ as calculated in equation (5.13), since they represent the measure for exactly this uncertainty. Thus, the weighting coefficients w_κ are determined by:

$$w_\kappa = \frac{1}{\max\left(s_{g\perp,o}, s_{g\perp,p}\right)_\kappa} \tag{6.14}$$

6.2.2 Position Uncertainty

Calculating the weighting coefficients as described above also offers the possibility to directly gain an estimate for the position uncertainty in $\Delta\boldsymbol{x} = [\Delta X, \Delta Y]^T$. This is given in form of the variance-covariance matrix $\boldsymbol{C}_{\Delta\boldsymbol{x}}$, which will be derived in the following.

Generally it is true that if

$$\boldsymbol{x} = \boldsymbol{B} \cdot \boldsymbol{b} \tag{6.15}$$

and $\boldsymbol{C}_{\boldsymbol{b}}$ being the variance-covariance matrix of vector $\boldsymbol{b}$, the variance-covariance matrix $\boldsymbol{C}_{\boldsymbol{x}}$ of vector $\boldsymbol{x}$ computes as

$$\boldsymbol{C}_{\boldsymbol{x}} = \boldsymbol{B} \cdot \boldsymbol{C}_b \cdot \boldsymbol{B}^T \tag{6.16}$$

Applying this to the present case, it becomes evident from equation (6.7) that $\boldsymbol{B} = \boldsymbol{A}^{-1}$. Further, assuming that there is no correlation among the

coefficients of $\boldsymbol{b}$, it is true that $\boldsymbol{C_b} = \boldsymbol{W}^{-2}$. With $\boldsymbol{x} = \Delta\boldsymbol{x}$ this leads to the notation:

$$\boldsymbol{C}_{\Delta\boldsymbol{x}} = \boldsymbol{A}^{-1} \cdot \boldsymbol{W}^{-2} \cdot \left(\boldsymbol{A}^{-1}\right)^T \tag{6.17}$$

As $\boldsymbol{A}^{-1}$ does generally not exist, a conversion of the above equation employing elementary matrix operations determines the final result for the variance-covariance matrix $\boldsymbol{C}_{\Delta\boldsymbol{x}}$ to:

$$\boldsymbol{C}_{\Delta\boldsymbol{x}} = \left(\boldsymbol{A}^T \cdot \boldsymbol{W}^2 \cdot \boldsymbol{A}\right)^{-1} \tag{6.18}$$

The verification that in the above equations the matrix dimensions agree and that $\boldsymbol{C}_{\Delta\boldsymbol{x}}$ is a symmetric $\langle 2 \times 2 \rangle$-matrix will be left to the reader.

It is evident that large coefficients in $\boldsymbol{C}_{\Delta\boldsymbol{x}}$ identify a large positional uncertainty, which is of course highly undesired under the aspect of a geometrically accurate self-localization.

The reason for substantial uncertainty is twofold: Firstly, the weighting coefficients of $\boldsymbol{W}$ may be extremely small, i. e. the variances ${s_{g\perp,o}}^2$, ${s_{g\perp,p}}^2$ are large, which will lead to large coefficients in $\boldsymbol{C}_{\Delta\boldsymbol{x}}$. This can easily be avoided by ruling out the according tuples (o, p).

Secondly and much more frequent in practical application, the matrix $\boldsymbol{A}$ may suffer from rank deficiency in that way that the apparently overdetermined or exactly determined equation system is in fact underdetermined or extremely ill-conditioned, i. e. the matrix $\boldsymbol{A}$ becomes singular or near-singular, which results in infinitely or enormously large coefficients in $\boldsymbol{C}_{\Delta\boldsymbol{x}}$, respectively. This situation cannot be avoided and at first glance it seems that under these circumstances the result of a self-localization will be rendered worthless. However, this is not true as some information can still be gained.

Of course, complete localization will no more be possible. Nonetheless, a partial localization binding one translatory and the rotatory degree of freedom in the plane can be performed. This induces further examination of $\boldsymbol{C}_{\Delta\boldsymbol{x}}$, which will be illustrated by means of a quantitative example. Unless otherwise noticed, the units used are metres for distances and square metres for (co-)variances and eigenvalues.

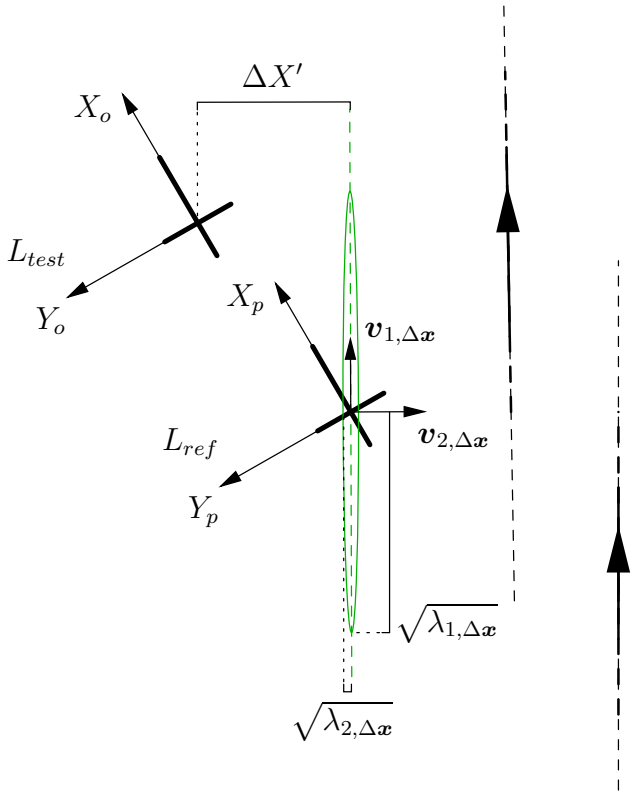


Figure 6.3: Example for a self-localization introducing a method for partial localization.

Given that the following values have been acquired:

$$\begin{aligned} \alpha_{g_0} &= 240^\circ \\ \alpha_{g_1} &= 241^\circ \\ \\ b_{g_0} &= (3.50 - 5.50) \\ b_{g_1} &= (2.10 - 4.05) \end{aligned} \tag{6.19}$$

which are illustrated in figure 6.3, and furthermore given the variances:

$$\begin{aligned} {s_{g\perp,0}}^2 &= (0.01)^2 \\ {s_{g\perp,1}}^2 &= (0.05)^2 \end{aligned} \tag{6.20}$$

defining the weighting matrix:

$$\boldsymbol{W} = \begin{bmatrix} \frac{1}{0.01} & 0 \\ 0 & \frac{1}{0.05} \end{bmatrix} \tag{6.21}$$

the resulting linear equation system reads as:

$$\begin{bmatrix} 100 \cdot \cos(240°) & 100 \cdot \sin(240°) \\ 20 \cdot \cos(241°) & 20 \cdot \sin(241°) \end{bmatrix} \cdot \Delta\boldsymbol{x} = \begin{bmatrix} 100 \cdot (-2.00) \\ 20 \cdot (-1.95) \end{bmatrix} \tag{6.22}$$

having the solution:

$$\Delta\boldsymbol{x} = \begin{bmatrix} \Delta\boldsymbol{X} \\ \Delta\boldsymbol{Y} \end{bmatrix} = \begin{bmatrix} 3.47 \\ 0.31 \end{bmatrix} \tag{6.23}$$

According to equation (6.18), the variance-covariance matrix computes as:

$$\boldsymbol{C}_{\Delta\boldsymbol{x}} = \begin{bmatrix} 6.41 & -3.69 \\ -3.69 & 2.13 \end{bmatrix} \tag{6.24}$$

As the exact solution in the present case would be $\Delta\boldsymbol{x}_e = [3.17, 0.48]^T$, the above calculated solution $\Delta\boldsymbol{x}$ proves to be not very accurate. This becomes evident by the large coefficients of $\boldsymbol{C}_{\Delta\boldsymbol{x}}$. (Note that the square root of the coefficients expresses the respective standard deviation given in metres.) Consequently, complete localization must be abandoned.

In the following, the eigenvalues $\lambda_{1,\Delta\boldsymbol{x}}$, $\lambda_{2,\Delta\boldsymbol{x}}$ of $\boldsymbol{C}_{\Delta\boldsymbol{x}}$ are determined. This can be achieved in a numerically stable way by taking the reciprocal values of the eigenvalues of $\boldsymbol{C}_{\Delta\boldsymbol{x}}^{-1} = \left(\boldsymbol{A}^T \cdot \boldsymbol{W}^2 \cdot \boldsymbol{A}\right)$. Similarly, the eigenvectors $\boldsymbol{v}_{1,\Delta\boldsymbol{x}}$, $\boldsymbol{v}_{2,\Delta\boldsymbol{x}}$ of $\boldsymbol{C}_{\Delta\boldsymbol{x}}$ are found by just adopting those of $\boldsymbol{C}_{\Delta\boldsymbol{x}}^{-1}$. Here the two universally valid properties are utilized that the eigenvalues of a matrix and its inverse matrix behave reciprocal to each other and that eigenvectors are maintained if the matrix gets inverted.

In the present example, the above procedure leads to:

$$\begin{aligned} \lambda_{1,\Delta\boldsymbol{x}} &= 8.54 \\ \lambda_{2,\Delta\boldsymbol{x}} &= 9.62 \cdot 10^{-5} \\ \\ \boldsymbol{v}_{1,\Delta\boldsymbol{x}} &= \begin{bmatrix} -0.866 \\ 0.499 \end{bmatrix} \\ \boldsymbol{v}_{2,\Delta\boldsymbol{x}} &= \begin{bmatrix} -0.499 \\ -0.866 \end{bmatrix} \end{aligned} \tag{6.25}$$

$\boldsymbol{C}_{\Delta\boldsymbol{x}}$ is depicted in figure 6.3 by its uncertainty ellipse. This visualizes that the uncertainty along $\boldsymbol{v}_{1,\Delta\boldsymbol{x}}$ is that large that a quantitative positional information would be rather useless (standard deviation is $\sqrt{\lambda_{1,\Delta\boldsymbol{x}}} = 2.92\,\mathrm{m}$). In

contrast, along $\boldsymbol{v}_{2,\Delta\boldsymbol{x}}$ standard deviation is only $\sqrt{\lambda_{2,\Delta\boldsymbol{x}}} = 9.81 \cdot 10^{-3}\,\mathrm{m}$ (not pictured according to actual scale in figure 6.3, as it would be too small). This indicates that a displacement ΔX_2 along $\boldsymbol{v}_{2,\Delta\boldsymbol{x}}$ can be accurately calculated.

In order to obtain ΔX_2, the following transformation is applied:

$$\Delta X_2 = \boldsymbol{v}_{2,\Delta\boldsymbol{x}}^T \cdot \Delta\boldsymbol{x} \tag{6.26}$$

In the case of the present example this results in:

$$\Delta X_2 = \begin{bmatrix} -0.499 & -0.866 \end{bmatrix} \cdot \begin{bmatrix} 3.47 \\ 0.31 \end{bmatrix} = -1.9981 \tag{6.27}$$

So one can resume that although complete localization proved to induce too much error and had to be abandoned, partial localization determining a displacement ΔX_2 along $\boldsymbol{v}_{2,\Delta\boldsymbol{x}}$ is still possible.

Concerning accuracy, it is to say that in the present case the exact value for the displacement along $\boldsymbol{v}_{2,\Delta\boldsymbol{x}}$ would be $\Delta X_{2,e} = -1.9983\,\mathrm{m}$. This shows that the above given standard deviation along $\boldsymbol{v}_{2,\Delta\boldsymbol{x}}$ appropriately reflects the actual facts and that in general an accurate estimate for ΔX_2 can be given.

Consequently, this kind of partial localization is an effective alternative whenever information from the surroundings becomes so sparse that the exact position is lost, e. g. while navigating through a long uniform corridor. Within the given limits the gained positional information is accurate and due to the small matrix dimensions of $\langle 2 \times 2 \rangle$ the additional computational effort is very small.

6.3 Concatenating Displacement Vectors

If complete relative self-localization is feasible, i. e. all three components of the displacement vector $\Delta\boldsymbol{X} = [\Delta X, \Delta Y, \Delta\Phi]^T$ can be determined with satisfying accuracy, an up-to-date geometric position with respect to a global coordinate system is maintained by concatenating the single displacement vectors.

Taking into account the geometric relations from figure 6.4, the position in the global coordinate system is iteratively calculated according to:

$$\boldsymbol{X} = \begin{bmatrix} X \\ Y \\ \Phi \end{bmatrix} := \begin{bmatrix} X + \Delta X \cdot \cos\Phi - \Delta Y \cdot \sin\Phi \\ Y + \Delta X \cdot \sin\Phi + \Delta Y \cdot \cos\Phi \\ \Phi + \Delta\Phi \end{bmatrix} \tag{6.28}$$

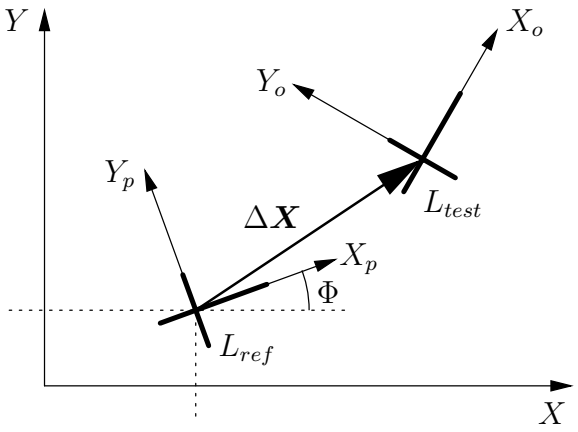


Figure 6.4: Updating the position with respect to a global coordinate system by concatenating displacement vectors.

The respective position uncertainty will again be given in form of a variance-covariance matrix denoted as $\boldsymbol{C_X}$. As the equation (6.28) is non-linear, calculation of $\boldsymbol{C_X}$ requires the Jacobian matrix $\boldsymbol{J}$ of the right hand side of the above equation, i. e. the matrix of partial derivatives of the position vector with respect to the vector of variables. This means:

$$\boldsymbol{J} = \frac{\partial \boldsymbol{X}}{\partial \left[X, Y, \Phi, \Delta X, \Delta Y, \Delta\Phi\right]^T}$$

$$\boldsymbol{J} = \left[\begin{array}{ccc|ccc} 1 & 0 & (-\Delta X \cdot \sin\Phi - \Delta Y \cdot \cos\Phi) & \cos\Phi & -\sin\Phi & 0 \\ 0 & 1 & (\Delta X \cdot \cos\Phi - \Delta Y \cdot \sin\Phi) & \sin\Phi & \cos\Phi & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \tag{6.29}$$

$$\boldsymbol{J} = \left[\boldsymbol{J}_1 \;\middle|\; \boldsymbol{J}_2\right]$$

The position vector $\boldsymbol{X}$ and the displacement vector $\Delta\boldsymbol{X}$ are uncorrelated. Furthermore, the translatory displacement $\Delta\boldsymbol{x} = [\Delta X, \Delta Y]^T$ and the rotatory displacement $\Delta\Phi$ are uncorrelated, either. So the likewise iterative notation of the variance-covariance matrix $\boldsymbol{C_X}$ can be simplified to:

$$\begin{aligned} \boldsymbol{C_X} = \begin{bmatrix} s_{XX} & s_{XY} & s_{X\Phi} \\ s_{YX} & s_{YY} & s_{Y\Phi} \\ s_{\Phi X} & s_{\Phi Y} & s_{\Phi\Phi} \end{bmatrix} &:= \boldsymbol{J}_1 \cdot \boldsymbol{C_X} \cdot {\boldsymbol{J}_1}^T + \boldsymbol{J}_2 \cdot \boldsymbol{C_{\Delta X}} \cdot {\boldsymbol{J}_2}^T \\ \text{with} \quad \boldsymbol{C_{\Delta X}} &= \begin{bmatrix} \boldsymbol{C_{\Delta x}} & \boldsymbol{0} \\ \boldsymbol{0} & {s_{\Delta\Phi}}^2 \end{bmatrix}_{\langle 3\times 3\rangle} \end{aligned} \tag{6.30}$$

In the following a simple test case is applied to the above equation: Firstly, it is assumed that the uncertainty inherent in the displacement vectors is different from zero, i. e. $\boldsymbol{C}_{\Delta \boldsymbol{X}} \neq \boldsymbol{0}$. Secondly, the initial position is supposed to be exact, i. e. $\boldsymbol{C}_{\boldsymbol{X}} = \boldsymbol{0}$ in the beginning. Furthermore, there shall be no motion at all, i. e. $\Delta \boldsymbol{X} \equiv \boldsymbol{0}$ will be measured over all times in the ideal case. This consequences in $\boldsymbol{J}_1 = \boldsymbol{J}_2 = \boldsymbol{I}$. Looking at the result that is yielded by equation (6.30), it becomes evident that the position uncertainty, i. e. the coefficients of $\boldsymbol{C}_{\boldsymbol{X}}$ will increase by accumulating the coefficients of $\boldsymbol{C}_{\Delta \boldsymbol{X}}$ with each additional concatenation of displacement vectors. This is plausible due to the fact that the error inherent in the displacement vectors propagates and accumulates with each concatenation.

Summarizing it is to say that on the basis of complete relative self-localization an up-to-date geometric position with respect to a global coordinate system can be maintained. This is accomplished without establishing a complex world model, but just by concatenating consecutive displacement vectors, which on their part depend on the respective current scan and its immediate predecessor scan as L_{test} and L_{ref}. The method excels by its simplicity, however, the gained position estimate suffers from unbound error accumulation, i. e. over time the given position will become more and more inaccurate and finally useless. In addition, position maintenance will unrecoverably break down, when complete localization is lost.

After introducing a few major issues of map building, the following section will detail on an approach that utilizes a self-creating environmental map in order to maintain an accurate, robust long term position estimate. In consideration of the above mentioned it will be respected that concatenation chains of displacement vectors are to be kept short and that the possibility to re-synchronize in order to reduce accumulated uncertainty is given. Furthermore, the approach will be able to deal with temporarily lost complete localization.

6.4 Map Building

In order to get a notion of the principal structure and problems of an environmental map in the robotics field, a few major issues will be briefly presented. The first question regards the format in which environmental data are to be stored. As mentioned in section 6.1, the map contains reference data with which current sensory data are matched. To render this procedure as effective as possible, environmental data are recorded in exactly that format that

is supported by the employed sensor, e. g. line segments in case of a PLRF. If one environmental map is to support multiple sensors, the non-trivial goal is to store data in a form that can be converted to any sensor specific format with moderate effort [Ruß94, HS96].

Another topic is the degree of refinement of the world model. Advantages and drawbacks of the different realizations are evident: A coarse degree of refinement means a small amount of data to be stored and to be processed. As drawback, the result of e. g. a self-localization may be not very precise, ambiguous or simply not possible, as detail information is missing. In contrast, a fine degree of refinement means substantial effort for data storing and processing, but may on the other hand generate localization results that are more precise. A desirable solution for this tradeoff would be a variable degree of refinement, i. e. coarse where it is possible, fine where it is necessary.

A further issue is the structuring of the data in the world model, i. e. the internal representation of the environmental data base. This is crucial insofar as access latencies as well as easy maintenance and extensibility of the world model are important matters that heavily depend on this internal representation.

The last important point to mention here is accuracy and consistency of the environmental map. These topics particularly become important if the map is not an a priori given ground truth with absolute accuracy, but an image of the environment that is successively acquired, updated and extended by sensory input, i. e. if an unknown environment has to be explored and the corresponding map has to be built by the mobile system on its own. As there is an extensive literature addressing optimization of accuracy and guaranteeing consistency of an explored map in a widespread environment [BCF+98, TBB+99, LM97, Gut99], the present work will not deal with these methods. Instead, it directs the focus on a particular map structure that, even though it still depends on these methods if maximum accuracy has to be reached, is capable of significantly reducing the thereby induced effort by dividing up a spacious environment into several small areas, which can independently considered.

In the following section the concept of a graph based map and its functioning in order to support localization is presented. For these explanations it is assumed that the graph based map is already existent, so localization is performed in an a priori known world model. Subsequently, it will be shown that the graph structure is also suited for the exploration scenario, i. e. a graph based environmental map can easily be maintained, extended and, with some restrictions, even built from the scratch by the mobile system on

its own.

6.4.1 Graph Based Map

The basic idea of the present approach is to establish the world model as an attributed, directed graph, i. e. the environmental map consists of attributed nodes connected by directed edges. The nodes represent certain locations in the environment, whereas the edges describe the paths between these locations. This concept has already been suggested by [KB91] and is similarly to [TGF+98] utilized here to maintain a robust localization of a mobile system in its environment.

The attributes of a node are firstly a position estimate $\boldsymbol{X}$ together with the variance-covariance matrix $\boldsymbol{C_X}$ expressing the respective position uncertainty. Secondly, a node is attributed with the preprocessed PLRF scan L_t taken from exactly the given position. So, the set of nodes V of the graph comprises the set of reference scans $\mathcal{L}_{\text{ref}}$ introduced in equation (5.18).

In case of complete localization and if there is a direct path between two nodes, the connecting edge needs no attributes at all. However, if the direct path between two nodes is obstructed, the edge is to be attributed with a list of consecutive intermediate positions that have to be accessed prior to heading for the target node. Furthermore, in the event that only partial localization is possible, e. g. while navigating through a long corridor, an edge can be attributed with its approximate bearing in the coordinate system and a topological directive like: *Follow the wall until complete localization can be re-established.* Along that way, a target node can be accurately reached although complete localization is temporarily lost.

An immediate consequence of involving the above mentioned topological directives is that the concept of a world model using one global coordinate system has to be abandoned, as geometric information may sometimes be incomplete. Instead, a novel concept is introduced that employs several local coordinate systems interconnected by topological relations. This means that one adheres to the present coordinate system as long as localization accuracy is rated precise enough. If this is no more ensured, a simple topological directive like wall following has to take over until complete localization can again be performed with satisfying accuracy. However, the catchment area of the current coordinate system has been left, i. e. a geometrically exact displacement vector with respect to the current coordinate system cannot be given. This is why the re-established complete geometric localization will be

done with respect to a new coordinate system, which is related to the current coordinate system by the respective topological directive.

Figure 6.5 on page 98 shows an exemplary structure of the environmental map.

The presented approach is operable as long as the employed topological directives are precise enough to guide the mobile vehicle into the catchment area of the new coordinate system. There it must be possible to perform a geometrically exact re-localization, so that a target node can be accurately accessed. Furthermore, it must be accepted that while travelling between two coordinate systems an exact geometric localization is not possible, as the vehicle is usually outside the catchment area of any existing coordinate system.

As the topological directives can only provide coarse directions, the above mentioned re-localization is a crucial issue. This means, it is necessary to robustly recognize a previously captured scene even if the view has changed and, in a worst case scenario, no initial guess of the whereabouts is available. Subsequently, it is to perform a relative localization with respect to the position where the captured scene has been taken. In the present case of PLRF scans robust recognition is performed by the scan matching procedure (cf. chapter 5) using the DP-Algorithm (cf. section 5.5). Here it becomes crucial that the developed scan matching is able to find the best matching reference scan $L_{\text{ref,best}}$ to a given test scan L_{test} from a set of reference scans $\mathcal{L}_{\text{ref}}$ without relying on positional a priori information. However, in case an initial guess of the position should be available, it is easy to integrate in the scan matching procedure: Reference scans that are unplausible due to their position can be excluded from the outset, rendering the matching process faster and more reliable because of reduced ambiguity. After having found $L_{\text{ref,best}}$ along that way, relative self-localization as described in section 6.2 provides the desired position estimate.

Figure 6.5 illustrates that the graph structure is able to support the above mentioned concept of a variable degree of refinement of the environmental map, i. e. if required, numerous nodes can be established in immediate geometric vicinity, whereas other areas are modeled by one single node. Along that way the different demands on accuracy and the differences in complexity of the environmental structure can be considered. For example, when passing a doorway, required accuracy is high as free space is limited. In addition, the perceived scene rapidly changes over travelled distance, as one room is left and a new room is entered. Consequently, this situation will require several nodes within a small area. On the other hand, when passing a large room

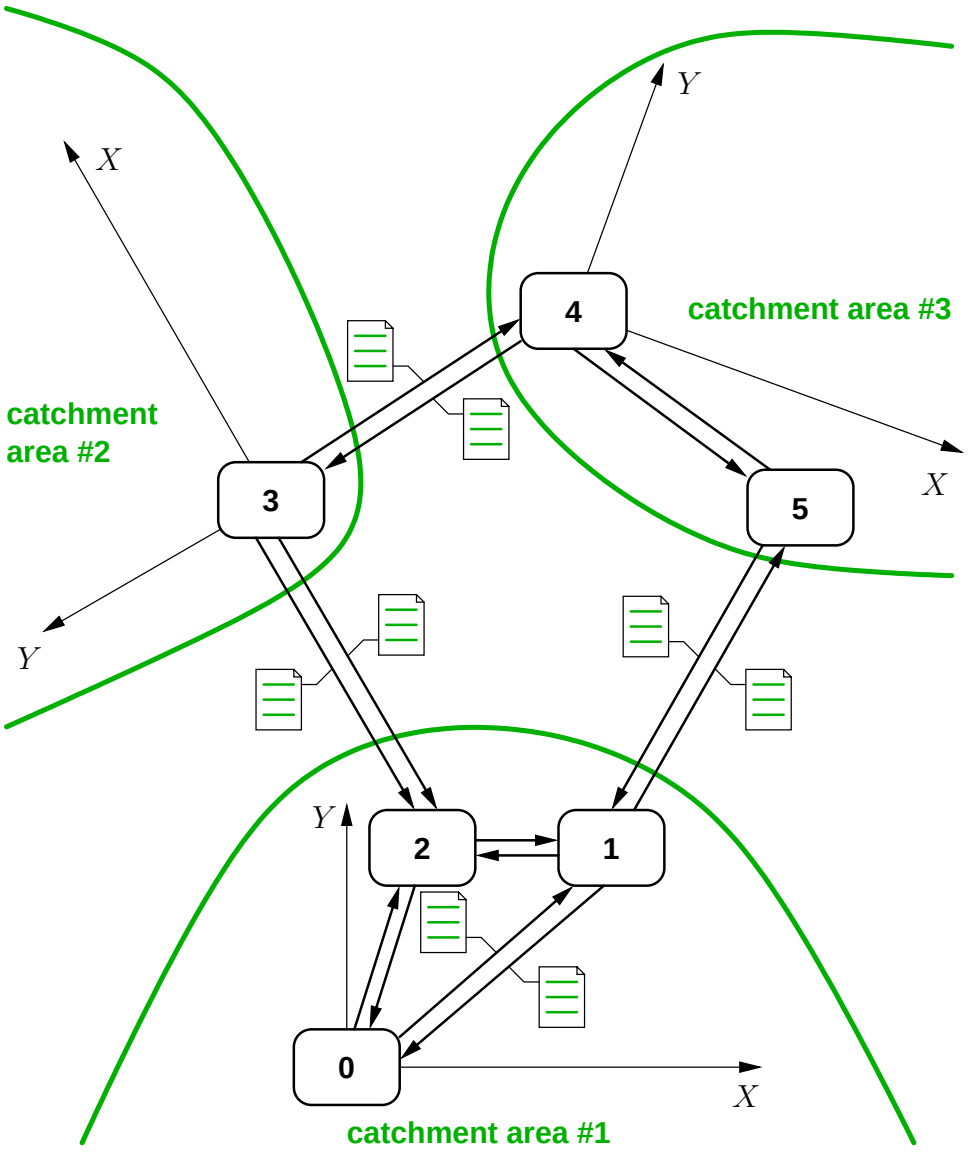


Figure 6.5: Exemplary structure of the graph based environmental map.

with a lot of free space, one single node may be an adequate modelling. This example illustrates that most of an environment can be modelled coarsely, while only small areas require fine modelling. However, fine modelling is inevitable and so the variable degree of refinement is the only way to keep the amount of data to be stored and processed in moderate bounds, especially if widespread environments are to be covered by the map.

In order to provide an exact position estimate, be it as attribute to a node or as a relative self-localization with respect to a node in the current coordinate system, it is important to keep concatenation chains of displacement vectors short. This is necessary since error accumulation as detailed in equation (6.30) will deteriorate the localization result with each additional concatenation. The graph based map with its several instead of one coordinate systems complies well with this requirement: The uncertainty of the first node of each coordinate system, i. e. the node that identifies the origin (in the example of figure 6.5 these are the nodes 0, 3 and 4) is assumed to be 0. This is justified as such an *initial node* represents the origin of a self-contained and geometrically independent coordinate system and therefore it can arbitrarily be set. Consequently, when travelling from the initial node of the current coordinate system to the initial node of a new coordinate system, uncertainty increases the larger the distance from the starting node becomes. If finally the catchment area of the current coordinate system is left, uncertainty increases towards infinity. At the moment when the catchment area of the new coordinate system is entered, uncertainty is decreased again and will be further reduced to a final value near zero the closer the vehicle approaches the initial node of the new coordinate system. From this it can be seen that the topological relations interconnecting the coordinate systems are able to break up concatenation chains of displacement vectors and along that way contribute to avoid that uncertainty is accumulated over a too long distance.

Of course, if one coordinate system comprises a larger area, concatenation chains get longer, accuracy and consistency issues may become a matter and the respective methods are to be applied. Nevertheless, if the suggested graph based map structure is properly pursued, the area that has to be processed will only be a small fraction of the overall area that is covered by the map. Thus, a great deal of the computational effort can be saved.

Due to the involved topological relations, the graph based approach does not support a geometrically exact CAD-drawing-like image of the overall map area. However, this is not a drawback as even for localization purposes like in the present case, it is not required either. Again, the corridor-example will illustrate this: If a corridor is only to pass, it is rather irrelevant if its

length is 4 m or 6 m, as long as it is possible to e. g. reliably detect its end or to recognize the crucial T-intersection where to turn right.

A definite advantage of the presented approach is that the environmental map actually is a graph. This means that all the algorithms available in this domain, e. g. searching the nodes or edges of a graph, adding or removing nodes or edges, finding the shortest or an alternative path etc. , can without any change be applied to the map. So the map does not only support localization by storing positions, reference scans and paths, but is also capable of assisting with navigation, i. e. finding an appropriate route and guiding the vehicle from a start position to a target position.

Up to this point the environmental map was supposed to be already existent. In the following section it is shown that a graph based map structure is also suited for the exploration scenario. In order to illustrate this, it will be roughly outlined how a graph based map could autonomously be explored by the mobile system on its own.

6.4.2 Exploring the Map

If it is to autonomously explore a map, the mobile system has to be equipped with an exploration strategy to provide a purposeful behaviour during the exploration phase. As exploration strategies are out of the scope of this work, only a very basic behaviour of the vehicle is considered in order to demonstrate the suitability of the graph based map approach for the exploration scenario: When only partial localization is feasible, the longest line segment in the current scan is determined and the vehicle moves along it. When complete localization is possible, the vehicle stops, turns around and leaves the area using the same line segment along which it has come. This behaviour does certainly not result in a thorough exploration of the environment, but it provides a simple travelling back and forth in a corridor.

In order to built a model of this scene, the following procedure is pursued: Whenever complete localization is possible, it is tried to recognize the area by matching the current scan with the previously recorded scans stored in the nodes of the already established graph. Now there are two cases:

1. If matching succeeds with a satisfying similarity measure, the respective area has obviously already been visited and the associated scan has been stored in a node of the map. Consequently, a new node will not be needed and a new edge from the previous node to the current node will only be established, if it does not already exist. Furthermore,

complete localization can be performed with respect to the previously recorded scan of that area.

2. If scan matching fails, the respective area is not yet registered in the map. Thus, a new initial node making up a new coordinate system is required. As well, a new edge connecting the new node with the previous node is introduced.

Along that way, a floor plan of a corridor as given in figure 6.6 a results in a graph based environmental map as illustrated in figure 6.6 b.

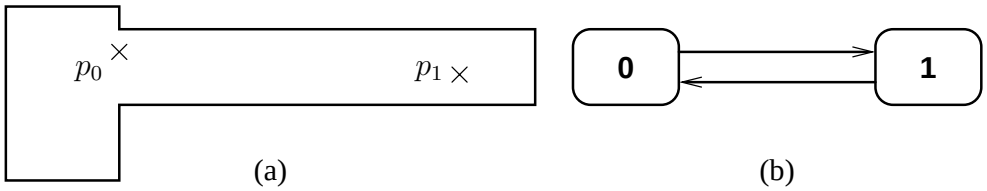


Figure 6.6: Floor plan (a) and corresponding graph based environmental map (b).

Node 0 and node 1 are both initial nodes of the respective coordinate systems. The scans the nodes are attributed with are in case of node 0 the scan taken from position p_0 and for node 1 the scan taken from position p_1. A similar experiment conducted in a real world environment will be presented in the following chapter.

6.4.3 Summary

Based on a particular assignment of line segments between two scans as it is provided by the scan matching procedure, the present chapter introduced a method to determine a displacement vector between the two respective scans. As this method generally involves an overdetermined linear equation system, numerical, scaling and accuracy issues were discussed in the following. In this context special focus was laid on the aspect of partial localization, which offers the possibility to still guide a vehicle even if an exact position can no more be given because the associated linear equation system becomes ill-conditioned or underdetermined.

In order to obtain an up-to-date position estimate it was illustrated how to concatenate displacement vectors and it was also indicated that position uncertainty accumulates without bounds with each additional concatenation.

After discussing a few major topics of environmental maps in the field of robotics, the concept and the working of a graph based map utilizing several coordinate systems and topological directives were introduced. This was followed by a consideration of the specific properties of the graph structure. In this connection, it was pointed out that the structure heavily depends on the robust recognition capabilities of scan matching in order to re-synchronize, but, on the other hand, effectively supports the concept of short concatenation chains of displacement vectors, thus successfully holding down position uncertainty.

Finally, there was a brief excursion to illustrate that the graph based map is well suited for the exploration scenario.

Chapter 7

Experimental Evaluation

The most critical touchstone for any novel development in the field of robotics naturally is its performance in a real world environment. Consequently, the localization system presented in this work has been implemented on a real robot and tested in different, typical indoor scenarios. The following sections describe these scenarios and document the experiments and their respective results. Preceding to this, the general experimental setup, i. e. the robot used in the experiments and several boundary conditions are introduced.

7.1 General Setup

The robot on which the localization system has been implemented and tested is our experimental mobile platform *MARVIN* (cf. figure 7.1). In the present case this quite popular name for a robotic system [Ada79] is an acronym for *M*obile *A*utonomous *R*obot with *VI*sion based *N*avigation [BBE+98, BEH+99].

MARVIN's chassis is the commercially available LABMATE base [TRC91], which has a differential drive with two driven wheels and four supporting castors (cf. figure 7.2). Though non-holonomic LABMATE is capable of linear and rotational motions.

The sensory equipment of LABMATE are a front and a rear bumper that issue an emergency stop to the driving motors if there is a mechanical contact. Secondly, an odometry is available, i. e. attached to the driving wheels are angular encoders that provide positional data by integrating encoder pulses. However, due to inevitable slippage between the driving wheels and the floor,

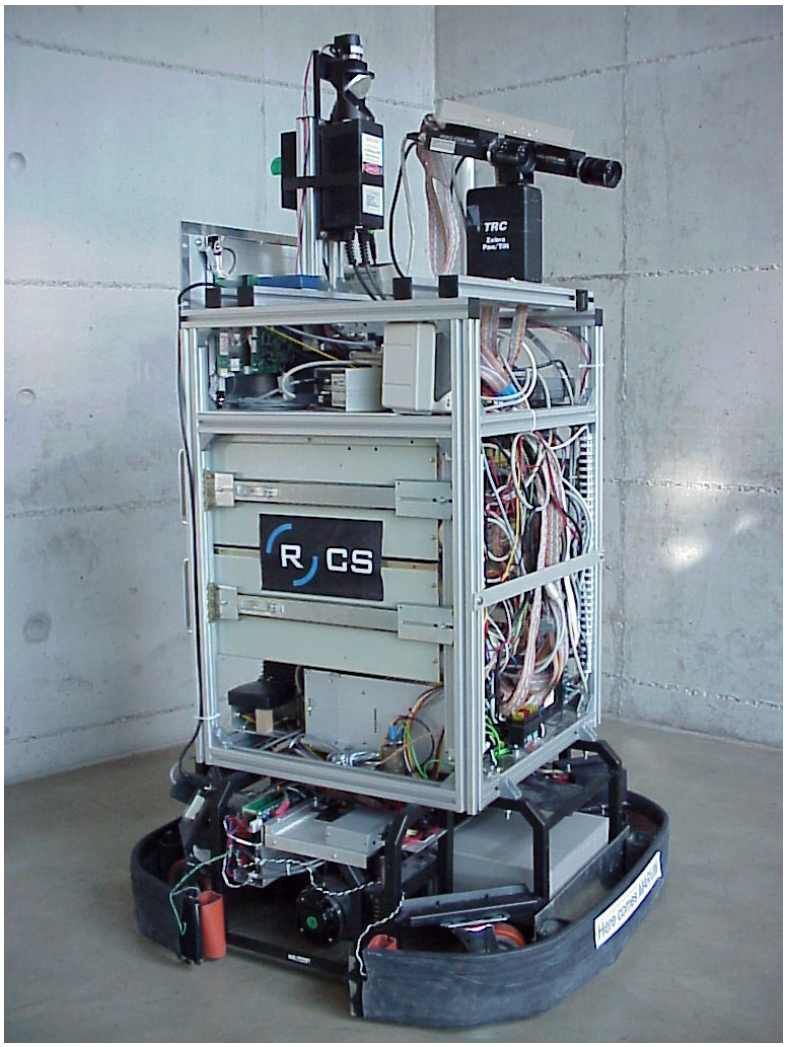


Figure 7.1: Experimental mobile platform *MARVIN*.

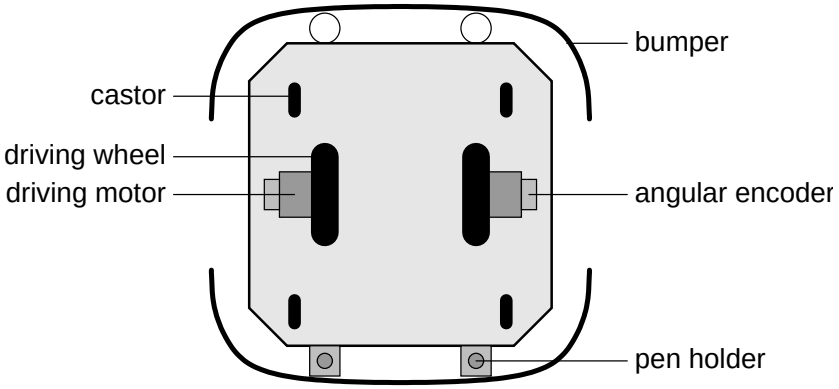


Figure 7.2: Bottom view of LABMATE.

especially if the vehicle's path is curved and the floor is not perfectly even, the integrating measuring principle of the odometry will quickly accumulate considerable error. So the odometric position estimate is quite unreliable after only a few metres of travelling. Consequently, odometric data is in no way qualified to provide a reference for the vehicle's position.

On the other hand, a position reference is essential if an actual test of the present localization system is to be carried out. As expensive geodetic measuring equipment was not at hand, a much simpler approach, still guaranteeing adequate accuracy was pursued: Two pen holders are firmly screwed to the LABMATE base. The spring mechanism of the pen holders allows for pen marks on the floor that can be gauged via a tape measure. The achievable accuracy of this method can be enhanced to about 1 to 2 mm and 1° if after a drive the vehicle is sent back to its exact starting position, i. e. the initial pen marks on the floor. Along that way the accumulated positional error can directly be revealed by comparing the actual (zero) position with the position given by the localization system.

Mounted on top of LABMATE is a rack holding four IPCs running under Linux, a TFT display, a stereo vision system, which is not in use here, and the PLRF sensor as already pictured in figure 3.1.

The algorithms related to PLRF localization are running in one Linux process. Visualization of the PLRF scans and the vehicle's position is handled in an extra process in order to provide decoupling between algorithms and graphics. A possible inquiry of odometric data is done within a third process.

Furthermore, it is to mention that during the experiments the PLRF localization system is running completely self-contained, i. e. neither a priori

knowledge of the environment nor of the vehicle's kinematics is employed. In addition, no initial positional guess, e. g. from the odometry, is utilized. So, the PLRF is the only source of information and the presented results reflect the pure performance of the PLRF localization system, in no way altered or covered by the output of another sensor.

7.2 Scenario 1: Laboratory

In the first experimental scenario *MARVIN* is roaming in a typical laboratory environment with white painted walls, a rather smooth linoleum flooring and a large window front (cf. figure 7.3). The surroundings were in no way

back view side view

Figure 7.3: *MARVIN* standing in his zero position in the laboratory.

prepared for the experiments, i. e. all interior equipment remained in its place; even some actually disturbing superstructures were not removed (cf. figure 7.3 back view: An antenna for wireless video signal transmission on the left border. Side view: A miniature model of a high bay racking in the upper right corner).

In order to get a feeling of what the PLRF is able to perceive in such an environment, figure 7.4 shows a typical preprocessed scan, shot from a position that is defined to be the zero position for the following experiments. In the figure *MARVIN* is visualized as a crosshair having one long leg which indicates the vehicle's forward direction. As can be seen there are several gaps in the scan due to poor reflections. The widest gap is found on the right hand side of *MARVIN*. This clearly results from the window front which is actually invisible to the PLRF.

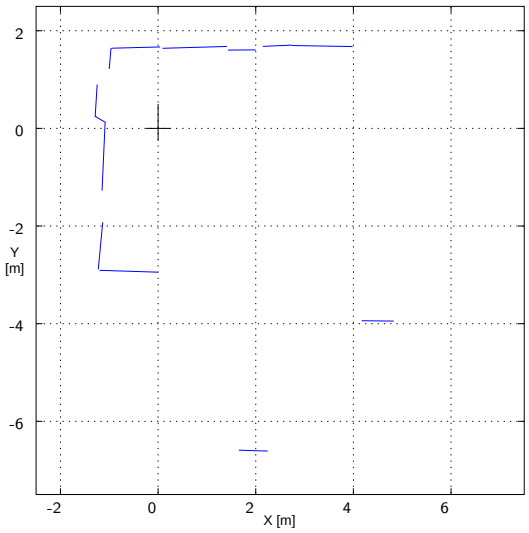
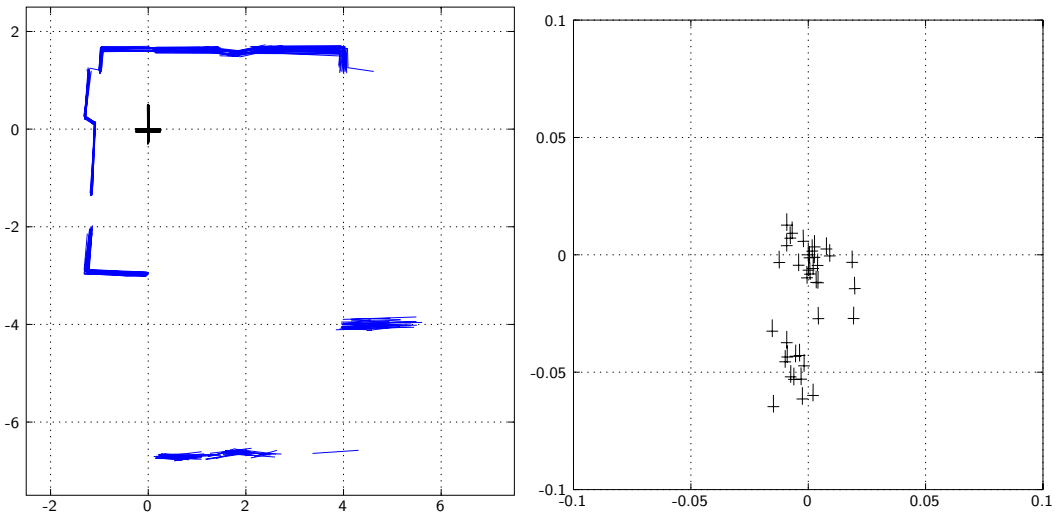


Figure 7.4: Preprocessed PLRF scan taken from the zero position in the laboratory.

7.2.1 Experiment 1: *MARVIN* Standing

During the first experiment *MARVIN* keeps standing totally still. The localization system is running, i. e. scans are successively taken and displacement vectors are determined. In the ideal case *MARVIN*'s crosshair would exactly rest in the zero position and the single scans would properly superimpose drawing one line segment right on top of the other. As can be seen from figures 7.5 a and 7.6 a this is not the case. The self-contained PLRF localization system knowing nothing about the halted vehicle perceives some motion due to noisy sensory data. This results in a jitter of *MARVIN*'s calculated position around the actual zero position. Figures 7.5 b and 7.6 b show magnified areas around the respective zero positions visualizing *MARVIN*'s apparent motion during about 40 successively taken scans.

In the example of figure 7.5 the displacement vectors are determined between the successively taken scans and are then concatenated to a simple chain as detailed in section 6.3. As a consequence one outlying displacement vector lastingly affects the vehicle's position. This can be seen from figure 7.5 b where two clustering areas are found: One around the actual zero position, another around an outlying position about 5 cm away. Furthermore, chaining displacement vectors means an accumulation of positional error, which can be expressed by the positional variance-covariance matrix $\boldsymbol{C_X}$. In the example



(a) (b)

Figure 7.5: Jitter of *MARVIN*'s calculated position around the actual zero position in case of chained displacement vectors.

of figure 7.5 the final value for $\boldsymbol{C_X}$ calculates to:

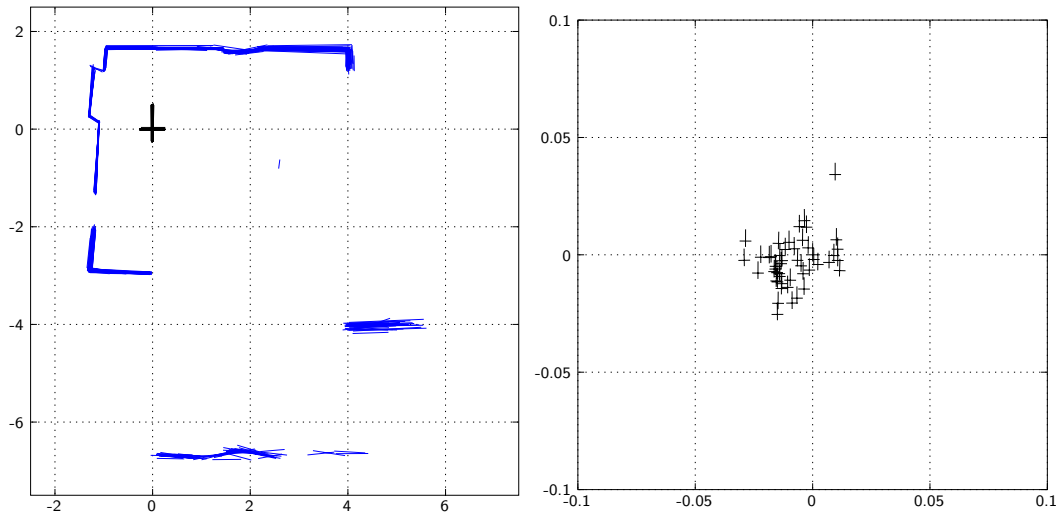
$$\boldsymbol{C_X} = \begin{bmatrix} s_{XX} & s_{XY} & s_{X\Phi} \\ s_{YX} & s_{YY} & s_{Y\Phi} \\ s_{\Phi X} & s_{\Phi Y} & s_{\Phi\Phi} \end{bmatrix} = \begin{bmatrix} 5.5 \cdot 10^{-4} & 6.8 \cdot 10^{-5} & 1.6 \cdot 10^{-4} \\ 6.8 \cdot 10^{-5} & 1.1 \cdot 10^{-3} & 7.6 \cdot 10^{-4} \\ 1.6 \cdot 10^{-4} & 7.6 \cdot 10^{-4} & 4.0 \cdot 10^{-2} \end{bmatrix} \tag{7.1}$$

with distances given in metres and angles in radians. So the according standard deviations derived from the above given variance-covariance matrix are:

$$\begin{aligned} s_X &= \sqrt{s_{XX}} = 2.4 \cdot 10^{-2}\,\mathrm{m} \\ s_Y &= \sqrt{s_{YY}} = 3.4 \cdot 10^{-2}\,\mathrm{m} \\ s_\Phi &= \sqrt{s_{\Phi\Phi}} = 0.20 = 11.4^\circ \end{aligned} \tag{7.2}$$

revealing moderate standard deviations of X and Y, whereas the deviation of Φ is not acceptable.

This is why in the example of figure 7.6 displacement vectors are not stringently determined between two successive scans. Instead, the scans are recorded and the displacement vector for the current scan is established with respect to the best matching scan recorded so far. As the number of shot scans linearly increases over time, this procedure implicates a linearly increasing number of scan matches, which would make the computational effort explode. To prevent this, the number of scans employed for scan matching



(a) (b)

Figure 7.6: Jitter of *MARVIN*'s calculated position around the actual zero position in case of displacement vectors determined with respect to the best matching recorded scan.

is bounded to a maximum of T, which typically ranges from 10 to 20. What is gained by this approach can be seen from figure 7.6 b: Although there is a clearly discernible outlying displacement vector about 3.5 cm away, the clustering around the actual zero position is not affected. In addition, as the displacement vectors do not form one long chain, the positional variance-covariance matrix $\boldsymbol{C_X}$ shows coefficients being about one order of magnitude smaller than in (7.1).

A typical appearance of $\boldsymbol{C_X}$ would be:

$$\boldsymbol{C_X} = \begin{bmatrix} 3.1 \cdot 10^{-5} & 2.7 \cdot 10^{-6} & 9.0 \cdot 10^{-6} \\ 2.7 \cdot 10^{-6} & 3.8 \cdot 10^{-5} & -1.5 \cdot 10^{-5} \\ 9.0 \cdot 10^{-6} & -1.5 \cdot 10^{-5} & 3.3 \cdot 10^{-3} \end{bmatrix} \tag{7.3}$$

with respective standard deviations:

$$\begin{aligned} s_X &= 5.6 \cdot 10^{-3}\,\mathrm{m} \\ s_Y &= 6.2 \cdot 10^{-3}\,\mathrm{m} \\ s_\Phi &= 0.06 = 3.2^\circ \end{aligned} \tag{7.4}$$

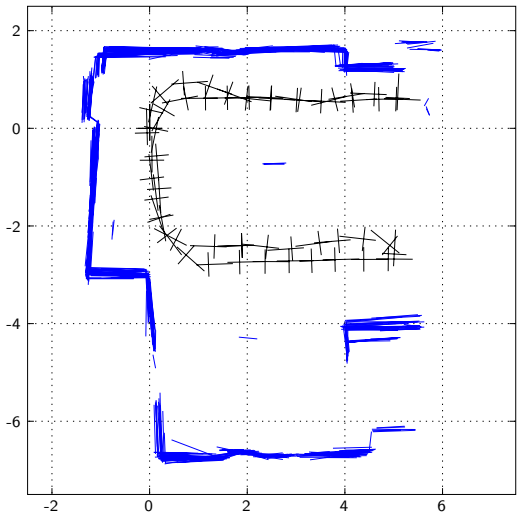
which is quite a satisfying result.

7.2.2 Experiment 2: *MARVIN* Moving

In the second experiment *MARVIN* is actually travelling the laboratory. Start and end point of *MARVIN*'s C-shaped course is the already known zero position. At first the vehicle is heading for the upper right turning point, then is coming back, passing the zero position and continuing its journey to the lower right turning point. From there it returns to the zero position, which is guaranteed to be exactly reached. The total distance covered amounts to about 26 m.

The following figures oppose the results of two experimental drives, which involve about 50 PLRF scans each: Figure 7.7 shows a drive with chained displacement vectors, whereas figure 7.8 depicts a drive with displacement vectors determined with respect to the best matching recorded scan as detailed in the previous section. Indicated are the superimposed scans together with the vehicle's respective position in its environment. This is achieved by applying the scan matching and localization techniques as detailed in chapters 5 and 6. Furthermore given are the positional variance-covariance matrices and the respective standard deviations as well as the output of the PLRF localization system and the output of *MARVIN*'s odometry when having returned to the zero position. Thus, any deviation from the actual zero position directly reveals the error of the respective localization system.

The drive of figure 7.8 excels by its extreme accuracy when returning to the starting point. This clearly is a consequence of the localization mode, which in the present case generates a minimally short concatenation chain of length 1 (identifiable from the zeros in $\boldsymbol{C_X}$). However, even in the worst case scenario of figure 7.7, where all 50 displacement vectors get simply chained, accuracy is still satisfying and significantly better than *MARVIN*'s odometry.



Variance-covariance matrix when returned to the zero position:

$$\boldsymbol{C_X} = \begin{bmatrix} 0.20 & 0.15 & -0.06 \\ 0.15 & 0.66 & -0.18 \\ -0.06 & -0.18 & 0.07 \end{bmatrix} \tag{7.5}$$

Respective standard deviations:

$$\begin{aligned} s_X &= 0.45\,\mathrm{m} \\ s_Y &= 0.81\,\mathrm{m} \\ s_\Phi &= 0.27 = 15.4^\circ \end{aligned} \tag{7.6}$$

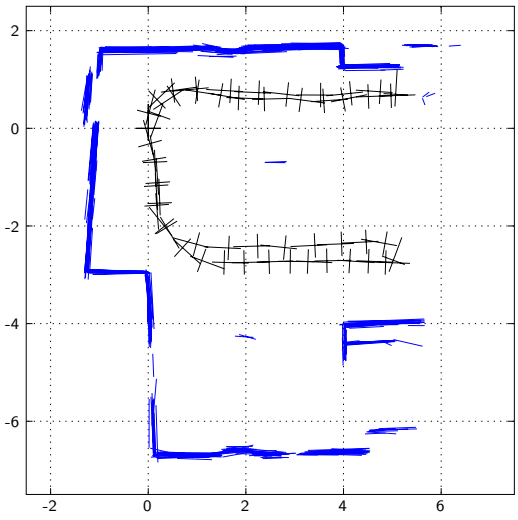
PLRF based position estimate when returned to the zero position:

$$\begin{aligned} X &= -0.05\,\mathrm{m} \\ Y &= -0.09\,\mathrm{m} \\ \Phi &= 1.4^\circ \end{aligned} \tag{7.7}$$

Odometry based position estimate when returned to the zero position:

$$\begin{aligned} X_{\mathrm{Odo}} &= -0.05\,\mathrm{m} \\ Y_{\mathrm{Odo}} &= -0.33\,\mathrm{m} \\ \Phi_{\mathrm{Odo}} &= 8.6^\circ \end{aligned} \tag{7.8}$$

Figure 7.7: Experimental drive with chained displacement vectors.



Variance-covariance matrix when returned to the zero position:

$$\boldsymbol{C_X} = \begin{bmatrix} 1.3 \cdot 10^{-5} & 1.6 \cdot 10^{-6} & 0 \\ 1.6 \cdot 10^{-6} & 2.8 \cdot 10^{-5} & 0 \\ 0 & 0 & 1.0 \cdot 10^{-3} \end{bmatrix} \tag{7.9}$$

Respective standard deviations:

$$\begin{aligned} s_X &= 3.6 \cdot 10^{-3}\,\mathrm{m} \\ s_Y &= 5.3 \cdot 10^{-3}\,\mathrm{m} \\ s_\Phi &= 0.03 = 1.8^\circ \end{aligned} \tag{7.10}$$

PLRF based position estimate when returned to the zero position:

$$\begin{aligned} X &= -0.0004\,\mathrm{m} \\ Y &= -0.0007\,\mathrm{m} \\ \Phi &= -0.1^\circ \end{aligned} \tag{7.11}$$

Odometry based position estimate when returned to the zero position:

$$\begin{aligned} X_{\mathrm{Odo}} &= -0.09\,\mathrm{m} \\ Y_{\mathrm{Odo}} &= -0.42\,\mathrm{m} \\ \Phi_{\mathrm{Odo}} &= 10.7^\circ \end{aligned} \tag{7.12}$$

Figure 7.8: Experimental drive with displacement vectors determined with respect to the best matching recorded scan.

7.2.3 Experiment 3: Low Scan Density

In the third experiment it is to illustrate that an additional feature of the PLRF localization system is its ability to allow for large distances between two successive localizations, i. e. a fairly accurate position estimate is still possible even if spatial density of PLRF scans is extremely low.

An example for this is given in figure 7.9 where only four scans taken from quite different positions could properly be superimposed.

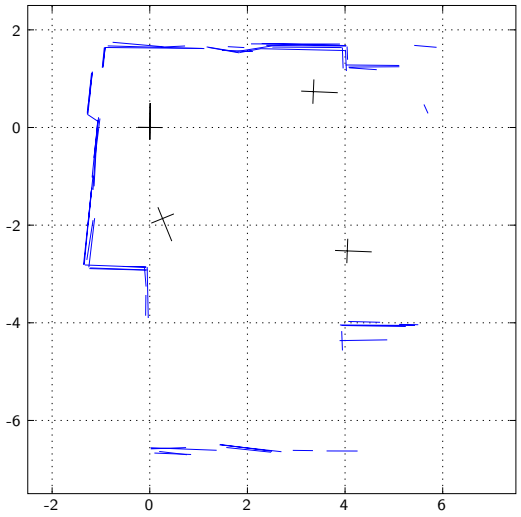


Figure 7.9: Experimental drive with only four scans taken from extremely different positions.

In the present example the variance-covariance matrix $\boldsymbol{C_X}$ with the largest coefficients is related to the lower right position and reads as:

$$\boldsymbol{C_X} = \begin{bmatrix} 3.0 \cdot 10^{-2} & 1.4 \cdot 10^{-2} & 1.0 \cdot 10^{-2} \\ 1.4 \cdot 10^{-2} & 2.4 \cdot 10^{-2} & 7.6 \cdot 10^{-3} \\ 1.0 \cdot 10^{-2} & 7.6 \cdot 10^{-3} & 4.6 \cdot 10^{-3} \end{bmatrix} \tag{7.13}$$

with standard deviations:

$$\begin{aligned} s_X &= 0.17\,\mathrm{m} \\ s_Y &= 0.16\,\mathrm{m} \\ s_\Phi &= 0.07 = 3.9^\circ \end{aligned} \tag{7.14}$$

Although the concatenation chain of displacement vectors has only length 2, the above given standard deviations are rather large, which indicates an

only moderate accuracy. An explanation for this is the heavily structured environment in the laboratory that quickly changes its appearance during roaming around.

7.3 Scenario 2: Elevator Hall

In the second experimental scenario *MARVIN* is wandering in an elevator hall of a modern office building. Compared to the laboratory, conditions are significantly worse: The walls are rough, dark grey concrete and the flooring is a rather uneven screed. Besides, the surroundings are characterized by steel and glass constructions and, notably, a number of heavily reflecting vertical pipes (cf. figure 7.10).

back view front view

Figure 7.10: *MARVIN* standing in his zero position in the elevator hall.

Inevitably, these construction properties result in rather fragmentary PLRF scans, which can be seen from figure 7.11 showing a single scan taken from *MARVIN*'s zero position in the elevator hall.

In the following, the experiments 1 to 3 as conducted in the laboratory scenario are now likewise repeated in the elevator hall. Boundary conditions of the experiments are the same as in the previous section, so the results can be given in a compact form. Furthermore, an additional fourth experiment is presented in this section showing that PLRF localization alone is able to cope with a typical doorway situation.

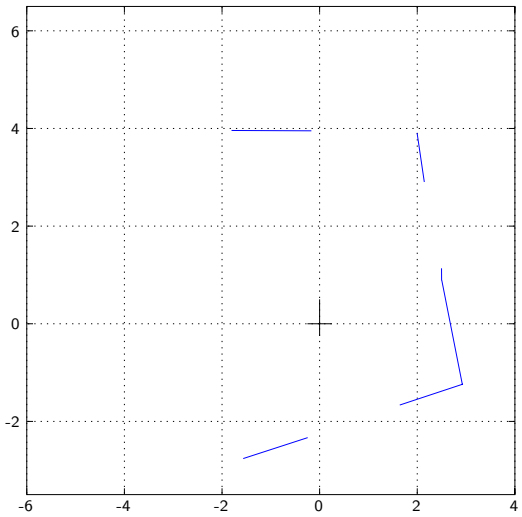
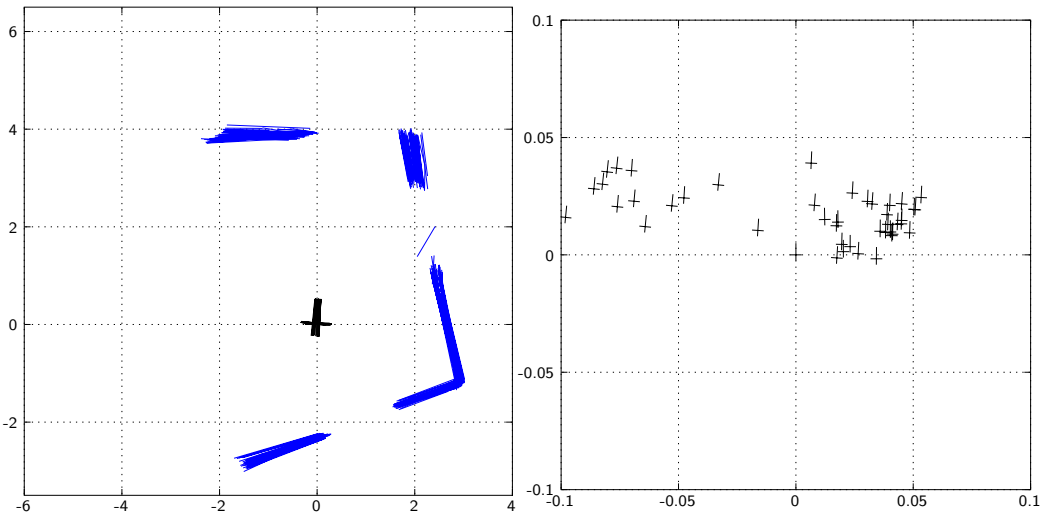


Figure 7.11: Preprocessed panoramic scan taken from the zero position in the elevator hall.

7.3.1 Experiment 1: *MARVIN* Standing

Figures 7.12 and 7.13 show the results of the PLRF localization system when *MARVIN* is standing still in his zero position in the elevator hall. In addition to the graphical visualization, the respective positional variance-covariance matrices together with the standard deviations are given.

What catches the eye is that in the more hostile environment of the elevator hall the already mentioned outlier sensitivity of simply chained displacement vectors leads to an actual drift of *MARVIN* (cf. figure 7.12 b). In contrast, the approach of determining displacement vectors with respect to the best matching recorded scan proves its stability even under worse conditions (cf. figure 7.13 b). This is also reflected in the coefficients of the variance-covariance matrices and the respective standard deviations. Most remarkable is the deviation of Φ, which is smaller by almost one order of magnitude in the second case.



(a) (b)

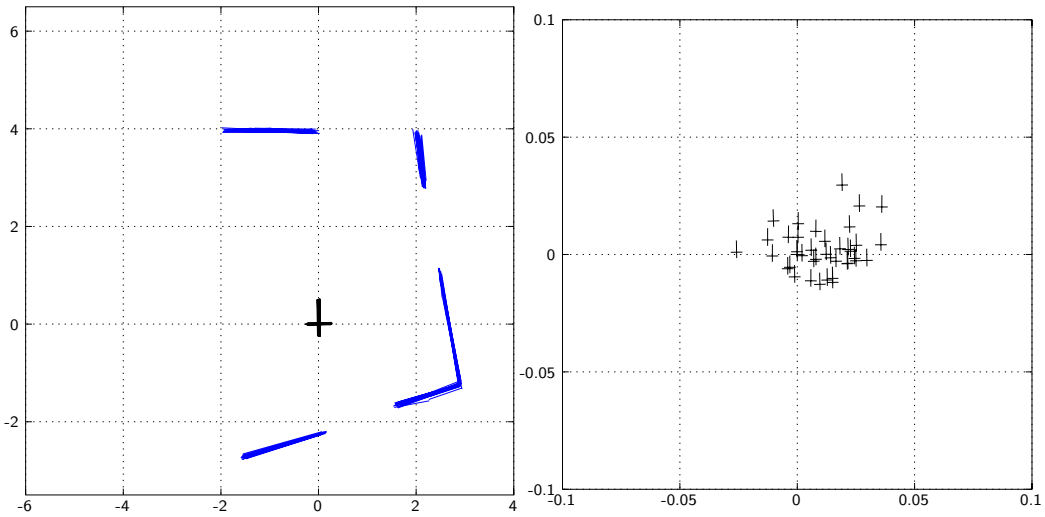
Variance-covariance matrix after about 40 concatenations:

$$\boldsymbol{C_X} = \begin{bmatrix} 3.7 \cdot 10^{-3} & 4.4 \cdot 10^{-4} & -5.9 \cdot 10^{-4} \\ 4.4 \cdot 10^{-4} & 3.2 \cdot 10^{-3} & -2.6 \cdot 10^{-3} \\ -5.9 \cdot 10^{-4} & -2.6 \cdot 10^{-3} & 3.1 \cdot 10^{-2} \end{bmatrix} \tag{7.15}$$

Standard deviations:

$$\begin{aligned} s_X &= 6.1 \cdot 10^{-2}\,\mathrm{m} \\ s_Y &= 5.7 \cdot 10^{-2}\,\mathrm{m} \\ s_\Phi &= 0.18 = 10.0^\circ \end{aligned} \tag{7.16}$$

Figure 7.12: Jitter of *MARVIN*'s calculated position around the actual zero position in case of chained displacement vectors.



(a) (b)

Typical variance-covariance matrix:

$$\boldsymbol{C_X} = \begin{bmatrix} 1.9 \cdot 10^{-4} & 1.1 \cdot 10^{-5} & -5.8 \cdot 10^{-6} \\ 1.1 \cdot 10^{-5} & 1.6 \cdot 10^{-4} & 3.1 \cdot 10^{-6} \\ -5.8 \cdot 10^{-6} & 3.1 \cdot 10^{-6} & 7.5 \cdot 10^{-4} \end{bmatrix} \tag{7.17}$$

Standard deviations:

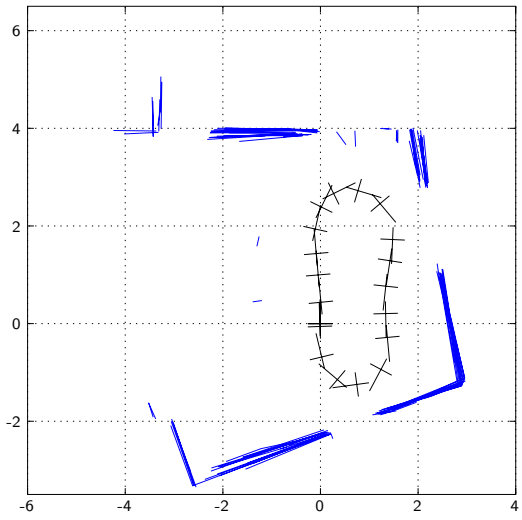
$$\begin{aligned} s_X &= 1.4 \cdot 10^{-2}\,\mathrm{m} \\ s_Y &= 1.3 \cdot 10^{-2}\,\mathrm{m} \\ s_\Phi &= 0.03 = 1.6^\circ \end{aligned} \tag{7.18}$$

Figure 7.13: Jitter of *MARVIN*'s calculated position around the actual zero position in case of displacement vectors determined with respect to the best matching recorded scan.

7.3.2 Experiment 2: *MARVIN* Moving

The following figures show the results when *MARVIN* is doing a clockwise O-shaped drive of about 8 m length in the elevator hall. Again, start and end point are exactly the zero position. The number of scans involved is about 20 in this example.

The conclusions to draw are very similar to those of the analogue experiment in the previous section. The drive of figure 7.15 features a very good accuracy when returning to the starting point, which is once again a consequence of the localization mode that minimizes the concatenation chain to length 1. A very satisfying fact is that in case of chained displacement vectors PLRF localization still performs better than *MARVIN*'s odometry, despite of the unfavourable conditions in the elevator hall (cf. figure 7.14).



Variance-covariance matrix when returned to the zero position:

$$\boldsymbol{C_X} = \begin{bmatrix} 8.2 \cdot 10^{-2} & -8.0 \cdot 10^{-3} & 2.8 \cdot 10^{-2} \\ -8.0 \cdot 10^{-3} & 2.0 \cdot 10^{-2} & -1.5 \cdot 10^{-2} \\ 2.8 \cdot 10^{-2} & -1.5 \cdot 10^{-2} & 2.7 \cdot 10^{-2} \end{bmatrix} \tag{7.19}$$

Respective standard deviations:

$$\begin{aligned} s_X &= 0.29\,\mathrm{m} \\ s_Y &= 0.14\,\mathrm{m} \\ s_\Phi &= 0.16 = 9.4^\circ \end{aligned} \tag{7.20}$$

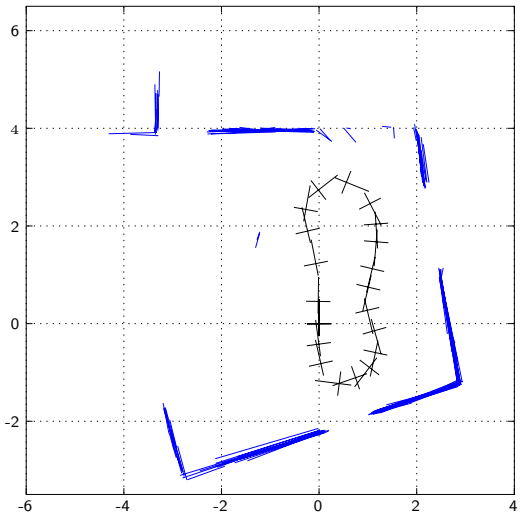
PLRF based position estimate when returned to the zero position:

$$\begin{aligned} X &= -0.009\,\mathrm{m} \\ Y &= -0.059\,\mathrm{m} \\ \Phi &= 1.9^\circ \end{aligned} \tag{7.21}$$

Odometry based position estimate when returned to the zero position:

$$\begin{aligned} X_{\mathrm{Odo}} &= 0.096\,\mathrm{m} \\ Y_{\mathrm{Odo}} &= -0.087\,\mathrm{m} \\ \Phi_{\mathrm{Odo}} &= 7.3^\circ \end{aligned} \tag{7.22}$$

Figure 7.14: Experimental drive with chained displacement vectors.



Variance-covariance matrix when returned to the zero position:

$$\boldsymbol{C_X} = \begin{bmatrix} 1.9 \cdot 10^{-5} & 2.6 \cdot 10^{-6} & 0 \\ 2.6 \cdot 10^{-6} & 3.6 \cdot 10^{-5} & 0 \\ 0 & 0 & 1.0 \cdot 10^{-4} \end{bmatrix} \tag{7.23}$$

Respective standard deviations:

$$\begin{aligned} s_X &= 4.4 \cdot 10^{-3}\,\mathrm{m} \\ s_Y &= 6.0 \cdot 10^{-3}\,\mathrm{m} \\ s_\Phi &= 0.01 = 0.6^\circ \end{aligned} \tag{7.24}$$

PLRF based position estimate when returned to the zero position:

$$\begin{aligned} X &= 0.003\,\mathrm{m} \\ Y &= -0.008\,\mathrm{m} \\ \Phi &= 0.005^\circ \end{aligned} \tag{7.25}$$

Odometry based position estimate when returned to the zero position:

$$\begin{aligned} X_{\mathrm{Odo}} &= 0.152\,\mathrm{m} \\ Y_{\mathrm{Odo}} &= -0.094\,\mathrm{m} \\ \Phi_{\mathrm{Odo}} &= 9.3^\circ \end{aligned} \tag{7.26}$$

Figure 7.15: Experimental drive with displacement vectors determined with respect to the best matching recorded scan.

7.3.3 Experiment 3: Low Scan Density

Repeating this experiment in the elevator hall yields an interesting result.

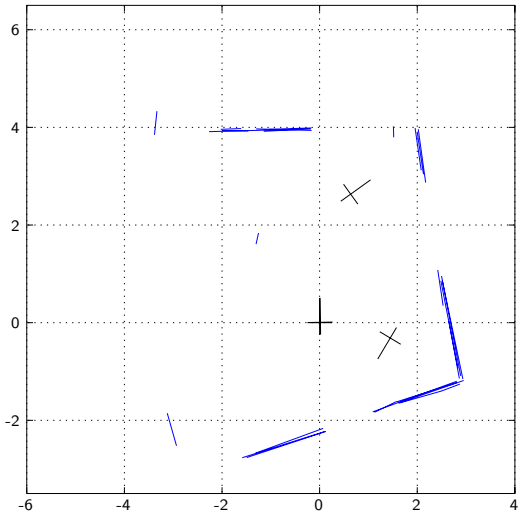


Figure 7.16: Experimental drive with only three scans taken from extremely different positions.

In this case the variance-covariance matrix $\boldsymbol{C_X}$ with the largest coefficients is related to the upper position and reads as:

$$\boldsymbol{C_X} = \begin{bmatrix} 1.0 \cdot 10^{-2} & -2.1 \cdot 10^{-3} & -3.7 \cdot 10^{-3} \\ -2.1 \cdot 10^{-3} & 7.9 \cdot 10^{-4} & 7.7 \cdot 10^{-4} \\ -3.7 \cdot 10^{-3} & 7.7 \cdot 10^{-4} & 2.1 \cdot 10^{-3} \end{bmatrix} \tag{7.27}$$

with standard deviations:

$$\begin{aligned} s_X &= 0.10\,\mathrm{m} \\ s_Y &= 0.03\,\mathrm{m} \\ s_\Phi &= 0.05 = 2.7^\circ \end{aligned} \tag{7.28}$$

Evidently, the coefficients of $\boldsymbol{C_X}$ are smaller than those of the analogue experiment in the laboratory. The reason for this is that the elevator hall is considerably less structured than the laboratory, i. e. as far as the PLRF is concerned, the appearance of the hall mostly remains the same even if a significant distance has been travelled.

7.3.4 Experiment 4: Passing a Doorway

In this fourth experiment it is shown that the PLRF localization system is able to cope with a typical doorway situation, i. e. a doorway is passed in order to leave one room and to enter another room.

Due to the complete change of the surroundings within a very short distance, the crucial point is to ensure a high scan density. Mostly, this demand can easily be satisfied as in the critical moments the door frame shadows two large sectors so that the number of line segments in a scan is reduced, rendering scan matching and localization faster. Secondly, a mobile vehicle normally slows down when approaching a narrow passage.

In the present experiment *MARVIN* starts from his zero position in the elevator hall, does a left turn around the elevator shaft and then turns right through a doorway-like passage into an adjacent small room. Displacement vectors are determined with respect to the best matching recorded scan, so ambiguities are well possible.

Figure 7.17 illustrates that during the experimental drive scans could properly be superimposed and localization was always correctly performed. The drive involved a complete change of the scene, i. e. when standing in the end position inside the small room no line segment of the elevator hall is perceivable.

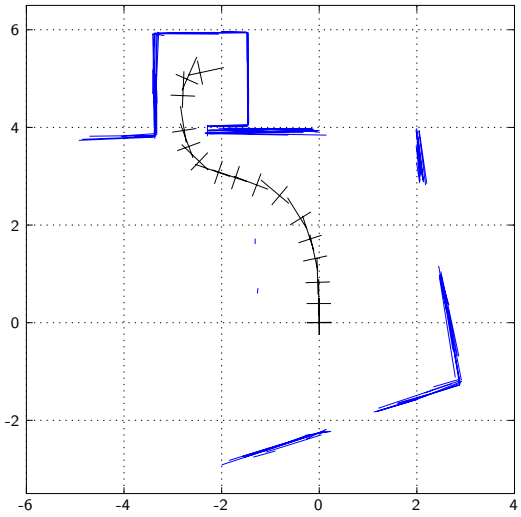


Figure 7.17: Passing a doorway.

The variance-covariance matrix in the end position and the respective stan-

dard deviations read as follows:

$$\boldsymbol{C_X} = \begin{bmatrix} 5.8 \cdot 10^{-2} & 2.7 \cdot 10^{-2} & -2.1 \cdot 10^{-2} \\ 2.7 \cdot 10^{-2} & 1.8 \cdot 10^{-2} & -8.0 \cdot 10^{-3} \\ -2.1 \cdot 10^{-2} & -8.0 \cdot 10^{-3} & 9.1 \cdot 10^{-3} \end{bmatrix} \tag{7.29}$$

$$\begin{aligned} s_X &= 0.24\,\mathrm{m} \\ s_Y &= 0.13\,\mathrm{m} \\ s_\Phi &= 0.10 = 5.5^\circ \end{aligned}$$

7.4 Scenario 3: Corridor

In the final experimental scenario *MARVIN* is passing a corridor as already outlined in section 6.4.2. Conditions are similar to those in the elevator hall: Walls made of rough, dark grey concrete and an uneven screed as flooring. Nontypical for a corridor is that a wall is only on one side. Instead of the wall on the other side there is a gallery with a window front, which is invisible to the PLRF. Another remarkable anomaly of the corridor under consideration is that it is not straight, but very slightly bent (cf. figure 7.18).

Figure 7.18: *MARVIN* passing the corridor.

Figure 7.19 shows the start of *MARVIN*'s journey along the corridor. The proper superposition of two scans while standing in the start position shows that complete localization is well possible.

Figure 7.20 shows *MARVIN* underway running along the corridor.

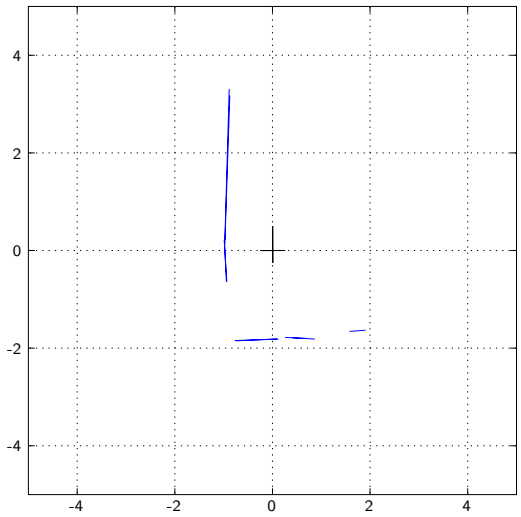


Figure 7.19: *MARVIN* in start position for the corridor drive.

An attempt to perform complete localization generates the following variance-covariance matrix and standard deviations:

$$\boldsymbol{C_X} = \begin{bmatrix} 2.7 \cdot 10^{-4} & -4.6 \cdot 10^{-3} & 0 \\ -4.6 \cdot 10^{-3} & 1.1 \cdot 10^{-1} & 0 \\ 0 & 0 & 1.3 \cdot 10^{-5} \end{bmatrix} \tag{7.30}$$

$$\begin{aligned} s_X &= 1.7 \cdot 10^{-2}\,\mathrm{m} \\ s_Y &= 0.33\,\mathrm{m} \\ s_\Phi &= 0.004 = 0.2^\circ \end{aligned}$$

This reveals that in particular localization in Y will be erroneous, whereas localization in X and Φ could still be accurately performed. At any rate, this indicates that complete localization has to be abandoned.

When *MARVIN* reaches the end of the corridor, which is illustrated in figure 7.21, complete localization is feasible again, i. e. superimposing scans can be resumed. However, as complete localization was temporarily lost, the catchment area of the original coordinate system has been left. This means that a new coordinate system with a new initial node has to be established, which results in a graph as depicted in figure 6.6 b.

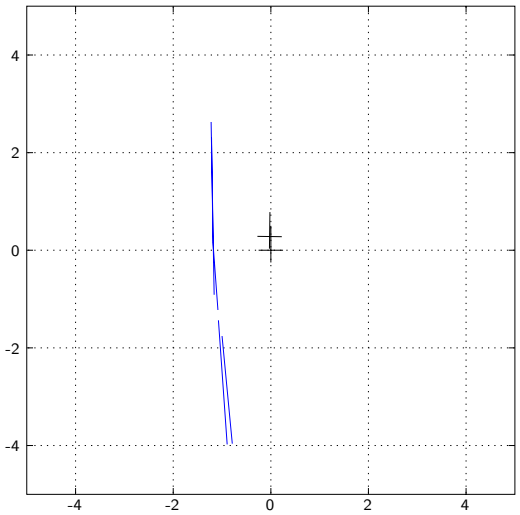


Figure 7.20: *MARVIN* running along the corridor.

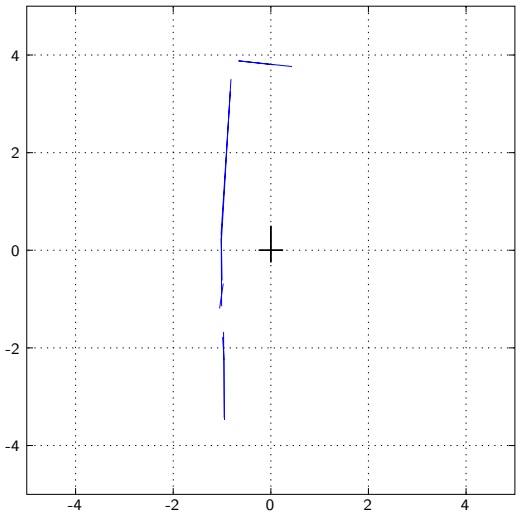


Figure 7.21: *MARVIN* at the end of the corridor.

Chapter 8

Conclusion and Future Work

This work presents a localization system suited for robots roaming in everyday indoor environments. This means the system can cope with environments that are not carefully engineered and prepared, as this is the case in a manufacturing scenario, but are typical office or residential buildings' interiors with people moving around and objects changing their place from time to time.

The employed sensor system is a panoramic laser range finder with a viewing angle of nearly 360° providing planar scans of the surroundings in a fixed height parallel to the floor. At first, the resulting range images are preprocessed by extracting straight line segments and their respective first and second order statistical moments. This procedure assures a considerable data reduction as well as an effective rejection of noisy and outlying scan points. Furthermore, the statistical nature of the scan point acquisition process is respected and preserved. Thus, the output of the preprocessing step is an ordered and cyclic sequence of line segments augmented by their first and second order moments.

The next milestone in the present approach is scan matching. In this step two preprocessed scans referred to as the previously recorded *reference scan* and the current *test scan* are considered. In general, the two scans are taken from different positions in the environment and have to be related to each other in such a way that the extracted line segments optimally coincide. The required pairwise assignment of line segments is accomplished by employing a modified version of the DP-Algorithm, which is a procedure known from the pattern matching and recognition domain. In order to make the DP-Algorithm applicable, the sequence of line segments as given from pre-

processing is interpreted as a cyclic pattern sequence. Furthermore, an appropriate similarity measure between line segments taking into account the geometrical and statistical properties of the line segments is defined. The final outcome of the DP-procedure is an assignment on line segment level and an overall measure rating the similarity of the two scans. This overall similarity measure enables the approach to be easily extended from one to several reference scans, i. e. by successively applying the DP-Algorithm it is possible to choose from a whole set of reference scans the one or, if desired, the ones that fit best to the given current scan. Another quality of this similarity measure is its proportional response to changes in the surroundings, e. g. a short line segment still visible in the reference scan and having eventually vanished in the test scan will only have little effect on the overall similarity. The proposed procedure does not rely on a priori knowledge or models of the environment: Starting with an initial scan, e. g. taken during "wake-up" of the system, scan matching is operable and as soon as a current scan is taken, assignment of line segments is provided without consulting any other source of information. Finally, the DP-Algorithm based scan matching procedure performs robust recognition of places that have been seen before and accurate matching of line segments that belong together.

Based on the matching result localization is realized by setting up and solving an overdetermined equation system, again taking into account the geometrical and statistical properties of the participating line segments. An important aspect addressed in this context is the uncertainty inherent in any position estimate and its propagation in case of successive localization cycles. A special case in this uncertainty discussion is that of partial localization, where only a subset of the existing degrees of freedom can be bound. The suggested map built from the localization result is an attributed graph, as this structure is able to combine the accuracy of a detailed map with the storage and processing efficiency of a map with a coarse degree of refinement. In addition, the graph structure in conjunction with the ability to recognize known places offers the possibility to use topological relations side by side with geometrical position estimates.

The suggested approach has been implemented on a real robot and experiments have been conducted in different, completely unprepared office environments under varying conditions. The obtained results show the suitability of the pursued method for navigational purposes: Even if surroundings are heavily structured or if sensory information becomes sparse the scans taken from different positions can properly be matched and superimposed. As can be seen from the experiments, the results would not have been that encouraging if only simple concatenation of displacement vectors were performed.

The crucial contribution to the actual performance of the approach is provided by the ability to recognize known places and to determine the current displacement vector with respect to the best matching recorded scan. This procedure offers the possibility to reduce already accumulated positional error, which otherwise increases without bounds. Another strength illustrated in the experiments is that known places can be recognized even if the sensor is located quite a distance away from the position where the original scan has been taken. As a consequence, the above mentioned error reduction can already become effective if the robot just approaches or passes by a known place. This robustness of recognition impressively becomes apparent when the robot passes a doorway into another room where a significant part of the current surroundings is abruptly replaced by a completely new environment.

A possible limitation of the presented technique may be observed if an environment comprises several places that are indistinguishable for the laser range finder and are beyond that located in immediate spatial vicinity such that the arising ambiguities cannot definitely be resolved, e. g. with the help of odometry. In such a case the localization system as presented in this work may choose the wrong of the possibilities and may in a bad case not recover from this erroneous decision. However, this problem can be overcome if the concept of Markov localization is seized, where not only one but several position estimates are allowed. This concept is well compatible with the proposed procedure as the DP-Algorithm based scan matching can not only render the best matching scan, but the best n matching scans to a given current scan together with the respective similarity measures. Along that way, in conjunction with repeating the actual localization step n times, n different and rated position estimates can be maintained. According to the technique of Markov localization the number of estimates may be reduced and ambiguities may correctly be resolved after some further roaming of the robot in its surroundings.

Bibliography

[Acu96] Acuity Research Inc., Menlo Park, CA 94025, USA. *AccuRange 4000, AccuRange High Speed Interface, AccuRange Line Scanner, User's Manual*, February 1996. 29, 31

[Ada79] D. Adams. *The Hitchhikers Guide to the Galaxy.* Crown Publishers, Inc., 1979. 103

[AS97] K. O. Arras and R. Y. Siegwart. Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building. In *Proc. SPIE, Mobile Robotics XII*, volume 3210, pages 42–53, 1997. 21

[BB99] O. Bengtsson and A.-J. Baerveldt. Localization in Changing Environments by Matching Laser Scans. In *Proc. 3rd European Workshop on Advanced Mobile Robots (EUROBOT'99)*, pages 169–176, September 1999. 25

[BBE+98] Stefan Blum, Darius Burschka, Christof Eberst, Tobias Einsele, Alexa Hauck, Norbert O. Stöffler, and Georg Färber. Autonome Exploration von Innenräumen mit der Multisensorik-Plattform MARVIN. In *Autonome Mobile Systeme*, Informatik aktuell, pages 138–147. Springer-Verlag, 1998. 103

[BCF+98] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The Interactive Museum Tour-Guide Robot. In *Proc. of the National Conference on Artificial Intelligence (AAAI'98)*, 1998. 22, 51, 95

[BD62] R. E. Bellman and S. E. Dreyfus. *Applied Dynamic Programming.* Princeton University Press, 1962. 72

[BDFC98] W. Burgard, A. Derr, D. Fox, and A.B. Cremers. Integrating Global Position Estimation and Position Tracking for Mobile Robots: The Dynamic Markov Localization Approach. In

Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'98), 1998. 22

[BEH+99] Stefan Blum, Tobias Einsele, Alexa Hauck, Norbert O. Stöffler, Georg Färber, Thorsten Schmitt, Christoph Zierl, and Bernd Radig. Eine konfigurierbare Systemarchitektur zur geometrisch-topologischen Exploration von Innenräumen. In *Autonome Mobile Systeme*, Informatik aktuell, pages 378–387. Springer-Verlag, November 1999. 103

[BF96] J. Borenstein and L. Feng. Measurement and Correction of Systematic Odometry Errors in Mobile Robots. *IEEE Trans. on Robotics and Automation*, 12(6):869–880, December 1996. 19

[BHEF98] Thorsten Bandlow, Alexa Hauck, Tobias Einsele, and Georg Färber. Recognising Objects by their Silhouette. In *IMACS Conf. on Comp. Eng. in Systems Appl. (CESA'98)*, pages 744–749, April 1998. 72

[Bor94] J. Borenstein. Internal Correction of Dead-reckoning Errors With the Smart Encoder Trailer. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'94)*, pages 127–134, September 1994. 19

[CMNT99] J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardós. The SPmap: A Probabilistic Framework for Simultaneous Localization and Map Building. *IEEE Trans. on Robotics and Automation*, 15(5):948–952, October 1999. 21

[Cox91] I. J. Cox. Blanche — An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle. *IEEE Trans. on Robotics and Automation*, 7(2):193–204, April 1991. 21, 24

[CT96] J. A. Castellanos and J. D. Tardós. Laser-Based Segmentation and Localization for a Mobile Robot. In *Proc. Int. Symposium on Robotics and Manufacturing (ISRAM'96)*, pages 101–108, 1996. 21

[CTS97] J. A. Castellanos, J. D. Tardós, and G. Schmidt. Building a Global Map of the Environment of a Mobile Robot: The Importance of Correlations. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'97)*, pages 1053–1059, 1997. 19, 21

[CW90] I. J. Cox and G. T. Wilfong, editors. *Autonomous Robot Vehicles*. Springer-Verlag, 1990.

[DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977. 22

[Ein93] T. Einsele. Objektorientierte Realisierung einer datengetriebenen Viterbi-Suche für die Spracherkennung. Master's thesis, TU München, Fakultät für Elektro- und Informationstechnik, July 1993. 72

[Ein97] T. Einsele. Real-Time Self-Localization in Unknown Indoor Environments using a Panorama Laser Range Finder. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'97)*, pages 697–703, Grenoble, France, September 1997.

[FBDT99] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In *Proc. of the National Conference on Artificial Intelligence (AAAI'99)*, 1999. 23

[FBTC98] D. Fox, W. Burgard, S. Thrun, and A.B. Cremers. Position Estimation for Mobile Robots in Dynamic Environments. In *Proc. of the National Conference on Artificial Intelligence (AAAI'98)*, 1998. 22

[FLWÅ93] J. Forsberg, U. Larsson, Å. Wernersson, and P. Åhman. Navigation in Cluttered Rooms using a Range Measuring Laser and the Hough Transform. In *Proc. Int. Conf. on Intelligent Autonomous Systems 3 (IAS-3)*, 1993. 43

[GBFK98] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An Experimental Comparison of Localization Methods. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'98)*, Victoria, Canada, 1998. 25

[GK99] J.-S. Gutmann and K. Konolige. Incremental Mapping of Large Cyclic Environments. In *International Symposium on Computational Intelligence in Robotics and Automation (CIRA'99)*, 1999. 25

[GL93] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, second edition, 1993. 88

[GOR94] J. Gonzalez, A. Ollero, and A. Reina. Map Building for a Mobile Robot equipped with a 2D Laser Rangefinder. In *Proc. IEEE*

Int. Conf. on Robotics and Automation (ICRA'94), volume 3, pages 1904–1909, 1994. 21

[GS96] J.-S. Gutmann and C. Schlegel. AMOS: Comparison of Scan Matching Approaches for Self-Localization in Indoor Environments. In *Proc. 1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT'96)*, pages 61–67, 1996. 25

[GT93] J. Gregor and M. G. Thomason. Dynamic Programming Alignment of Sequences Representing Cyclic Patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(2):129–135, February 1993. 72

[GT96] J. Gregor and M. G. Thomason. Efficient Dynamic Programming Alignment of Cyclic Strings by Shift Elimination. *Pattern Recognition*, 29(7):1179–1185, 1996. 72

[Gut99] J.-S. Gutmann. *Robuste Navigation autonomer mobiler Systeme*. PhD thesis, Universität Freiburg, Institut für Informatik, 1999. 21, 40, 41, 43, 45, 51, 95

[GWN99] J.-S. Gutmann, T. Weigel, and B. Nebel. Fast, Accurate, and Robust Self-Localization in Polygonal Environments. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'99)*, 1999. 21

[HH01] U. D. Hanebeck and J. Horn. An Efficient Method for Simultaneous Map Building and Localization. In *Proc. SPIE*, volume 4385, pages 233–243, April 2001. 19, 21

[HS96] A. Hauck and N. O. Stöffler. A Hierarchical World Model with Sensor- and Task-Specific Features. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'96)*, pages 1614–1621, 1996. 95

[JAWA00] P. Jensfelt, D. J. Austin, O. Wijk, and M. Anderson. Feature Based Condensation for Mobile Robot Localization. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'00)*, 2000. 23

[JK99] P. Jensfelt and P. Kristensen. Active Global Localisation for a Mobile Robot Using Multiple Hypothesis Tracking. In *Proc. 16th Int. Joint Conf. on Artificial Intelligence*, 1999. 26

[KB91] B. Kuipers and Y.-T. Byun. A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representa-

tions. *Journal of Robotics and Autonomous Systems*, 8:47–63, 1991. 26, 96

[KL97] Y. D. Kwon and J. S. Lee. A Stochastic Environment Modelling Method for Mobile Robot by using 2-D Laser scanner. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'97)*, volume 2, pages 1688–1693, April 1997. 21

[KWN99] C. Kunz, T. Willeke, and I. R. Nourbakhsh. Automatic Mapping of Dynamic Office Environments. *Autonomous Robots*, 7(2):131–142, September 1999. 26

[LDW91] J. J. Leonard and H. F. Durrant-Whyte. Mobile Robot Localization by Tracking Geometric Beacons. *IEEE Trans. on Robotics and Automation*, 7(3):376–382, June 1991. 21

[LDWC90] J. Leonard, H. Durrant-Whyte, and I. J. Cox. Dynamic Map Building for an Autonomous Mobile Robot. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'90)*, volume 1, pages 89–96, 1990. 20, 21

[LFW94] U. Larsson, J. Forsberg, and Å. Wernersson. On Robot Navigation Using Identical Landmarks: Integrating Measurements from a Time-of-Flight-Laser. In *Proc. 1994 IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pages 17–26, October 1994. 21

[LH74] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall Int., Inc., 1974. 86, 88

[LM94] F. Lu and E. Milios. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. In *Proc. Computer Vision and Pattern Recognition (CVPR'94)*, pages 935–938, 1994. 21, 24

[LM97] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333–349, October 1997. 21, 25, 51, 95

[LMP98] K.-M. Lee, P. Meer, and R.-H. Park. Robust Adaptive Segmentation of Range Images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(2):200–205, February 1998. 43

[Mae90] M. Maes. On a Cyclic String-To-String Correction Problem. *Information Processing Letters*, 35:73–78, June 1990. 72

[PEBR94] B. Plannerer, T. Einsele, M. Beham, and G. Ruske. A Continuous Speech Recognition System Integrating Additional Acoustic Knowledge Sources in a Data-Driven Beam Search Algorithm. In *Proc. Int. Conf. on Spoken Language Processing (ICSLP'94)*, pages 17–20, 1994. 72

[PH74] T. Pavlidis and S. Horowitz. Segmentation of planar curves. *IEEE Transactions on Computers*, C-23:860–870, 1974. 43

[Ren93] W. D. Rencken. Concurrent Localisation and Map Building for Mobile Robots Using Ultrasonic Sensors. In *IEEE Int. Workshop on Intelligent Robots and Systems (IROS '93)*, 1993. 19

[RL87] Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc., 1987. 46

[Rußß94] A. Ruß. *Sensornahe Umgebungsmodellierung mit echtzeitfähigen Zugriffsfunktionen*. PhD thesis, TU München, Fakultät für Elektro- und Informationstechnik, 1994. 95

[SK95] R. Simmons and S. Koenig. Probabilistic Navigation in Partially Observable Environments. In *Proc. 14th Int. Joint Conf. on Artificial Intelligence*, July 1995. 22, 26

[SK97] H. Shatkay and L. P. Kaelbling. Learning Topological Maps with Weak Local Odometric Information. In *Proc. 15th Int. Joint Conf. on Artificial Intelligence*, 1997. 22, 23, 26

[TBB+99] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: A Second-Generation Museum Tour-Guide Robot. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'99)*, pages 1999–2005, 1999. 23, 51, 95

[TBF98] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (joint issue)*, 31(5):1–25, 1998. 22

[TBF00] S. Thrun, W. Burgard, and D. Fox. A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'00)*, 2000. 22

[TFB00] S. Thrun, D. Fox, and W. Burgard. Monte Carlo Localization With Mixture Proposal Distribution. In *Proc. of the National Conference on Artificial Intelligence (AAAI'00)*, 2000. 23

[TGF+98] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and B. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *Proc. of the National Conference on Artificial Intelligence (AAAI'98)*, 1998. 26, 96

[TRC91] TRC, Transition Research Corporation, Danbury, CT, USA. *LABMATE User Manual, Version 5.21L-f*, 1991. 103

[TY85] W.-H. Tsai and S.-S. Yu. Attributed String Matching with Merging for Shape Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7(4):453–462, July 1985. 72

[VTGA94] S. J. Vestli, N. Tschichold-Gürman, and H. Andersson. Learning control and localisation of mobile robots. In P. Levi and Th. Bräunl, editors, *12. Fachgespräch "Autonome Mobile Systeme"*, pages 202–213. Springer-Verlag, 1994. 21, 43

[WvP95] G. Weiß and E. von Puttkamer. A Map Based On Laserscans Without Geometric Interpretation. In U. Rembold et al., editor, *Proc. Int. Conf. on Intelligent Autonomous Systems 4 (IAS-4)*, pages 403–407, March 1995. 21, 24

[WWvP94] G. Weiß, C. Wetzler, and E. von Puttkamer. Keeping Track of Position and Orientation of Moving Indoor Systems by Correlation of Range-Finder Scans. In *Proc. 1994 IEEE Int. Conf. on Intelligent Robots and Systems (IROS '94)*, pages 595–601, September 1994. 24, 43

[Zim95] U. R. Zimmer. Self Localization in Dynamic Environments. In *IEEE/SOFT International Workshop BIES'95*, 1995. 27