Odometry Estimation of a mobile Robot according to artificial landmarks

Diploma Thesis
Technische Kybernetik

presented by

Dominik Kraus

Prüfer: Prof. Dr.-Ing. A. Kistner
Tutor: Dipl.-Ing. Gastón Araguás

Institut MechB, Universität Stuttgart
Prof. Dr.-Ing. A. Kistner
Laboratorio de Investigacion CIII, Universidad Tecnológica Nacional Córdoba
Dipl.-Ing. Gastón Araguás

presented: 27.11.08

Abstract

This work is a part of the RoMAA (Robot Móvil de Arquitectura Abierta) project of the center of investigation of the Universidad Tecnologica Nacional in Córdoba Argentina. It treats the aim to estimate the odometry, the covered distance, of the mobile robot. For that it is elementary to know its position and orientation. This problem is been solved by means of vision based pose estimation. Therefore the robot is equipped with a digital camera which is used to detect artificial landmarks. The passive beacons have a quadratic form and a black and white chess-board pattern which make it easy to detect the needed reference points. The motivation of this work is to estimate the robots pose without knowing the environment nor having a global reference frame. There is also no restriction of the placement of the landmarks in the area. The robot shall be able to calculate his position by using the information recieved from the arbitrarily distributed landmarks. Beause of that there are presented two algorithms in order to achieve the estimation of the robot pose. One is used to calculate the pose and the other solves the problem of the so called pose ambiguity which can ocurre using planar targets like landmarks. In case of detecting two landmarks in one image the delivered informations are fused with the help of a kalman filter. This theoretical part is implemented in Octave, the Open Source alternative to Matlab for GNU/Linux based computers. At the end of the work simulation results of the deverse settings are demonstrated.

Contents

Chapter 1

Introduction

Since long time man tried to create machines to help him with his necessity. In the last decades with growing scientific knowledge this could be realized in more and more fields of the daily life. Not only that in mechanical construction where industrial robots already do a great part of former manual work. But also in military a space craft the robots are used in situations too dangerous for humans or simply unreachable. Another goal of robotics is to make the robot act and look like every time more similar to man. So there are sport challanges of robots yet. For all these achievements and for those which still will come the knowledge of the robots position and orientation is a main problem to solve. To get the robots pose in an often unknown surrounding area, instruments are needed to give information about the surrounding to diverse algorithms which make the calculation for the pose estimation. There are a lot of such sensors to provide the data, for example GPS, ultra sound, cameras, magnetical, rotation sensors and more. Here the mobile robot is equipped just with one fixed camera to detect the environment. The aim is to detect so called landmarks with a chess board pattern which are put arbitrarily in the ambience of the robot. With the aid of these landmarks their relative position to the robot can be estimated. In this way the desired covered distance, the odometry, of the robot can be calculated.
This work is structured as follows. In the second chapter there are given some information about the project of the research group CIII of the Universidad Tecnólogica Nacional in Córdoba Argentina. The robot and the environment are shown. After that in the third chapter the algorithms to estimate the pose of the robot to the landmark are presented. It is divided in two parts. The first is just about the pose estimation and the second about possible pose ambiguity. Then in chapter four a method to fuse the information of various sensors is explained, the Kalman filter. And at the end of this work the simulation results are illustrated.

Chapter 2

System presentation

Figure 2.1: romaa

In this chapter an understanding of the setting of this work is given. At first the used robot as mobil vehicle is described shortly, for more information see [9]. And after that the structure of the environment in which the robot moves and the shape of the landmarks which are to be detected are shown.

2.1 Robot

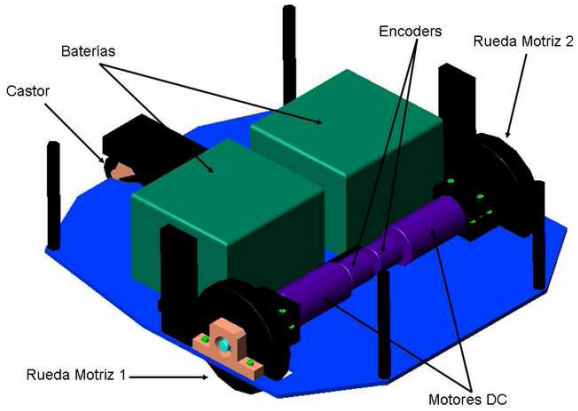

Figure 2.2: configuration of the romaa robot

The mechanical base of this work builds the mobile robot unit "RoMAA" (Robot Móvil de Arquitectura Abierta) of the Center of Investigation in Informatics for Engineering (Centro de Investigación en Informática para la Ingeniería, CIII) of the Universidad Tecnológica Nacional, Facultad Regional Córdoba in Argentina. The RoMAA was constructed to be able to implement diverse experiments in the working fields of computer vision, control and robotics. Further it shall provide a platform for an open source arquitechture. The mecanical design guarantees easy access to the components and sufficient space for sensors and actors in the front part. In terms of the computational section of the robot there can be used on-board pc or an wireless environment. The mecanical and electrical data can be read out from the tables 2.1 to 2.3.

Dimensions	570 mm length 570 mm width 200 mm height
Materials	aluminium 2024
Weight approx.	52 kg
Carga util	48 kg
Diameter of the wheels	147 mm
Wheelbase	503 mm
Turning radius	0 mm
R_A	407 mm

Table 2.1: mecanical caracteristics

Nominal tension of the motor	24 V
Maximal power of the motor	144 W (each)
Rpm of the motor	5000
Reduction	17,72:1
Maximal rpm of the axis of the motor	282,17
Maximum speed	2,22 $\frac{m}{s}$
Maximal acceleration	0,2 $\frac{m}{s^2}$
Distance resolution	0,5 mm

Table 2.2: engine carateristics

Nominal tension	12 V
Nominal capacitance @25°C, $I_{const} = 3A$	24 A-h
Dimension	175 mm length 166 mm width 126 mm height
Weight approx.	8.9 kg

Table 2.3: battery carateristics

2.2 Environment

The environment of the project is given by a planar indoor area of rectangular shape. In this area the robot shall be able to move freely without obstacles, at first, that would be part of the next step. So far the robot is driven by a remote control because the objective is just to get the right position of it.

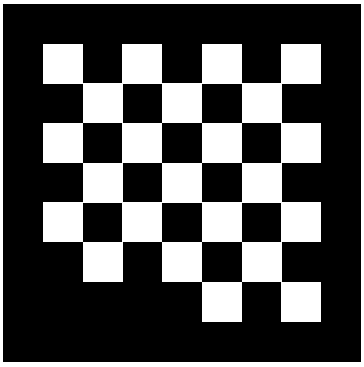

Figure 2.3: pattern of the landmark

To orientate itself the robot detects beacons which are passive landmarks like shown in figure 2.3 by means of the camera. Passive means in this case that they don't emit any signals. To put not too much restriction to the choice of the environment the landmarks are attached arbitrarily at the walls limiting the scene, but preferably at the hight of the camera. The only thing is that the robot and the camera, respectively, should be able to detect always at least one landmark. The setting of the system is exemplified for two landmarks in figure 2.4. In this figure the indices $l1$, $l2$ and c represent the first landmark, the second landmark and the camera. t is the respective translation vector in the xz plane and φ the rotation angel of the coordinate systems with respect to the camera frame.

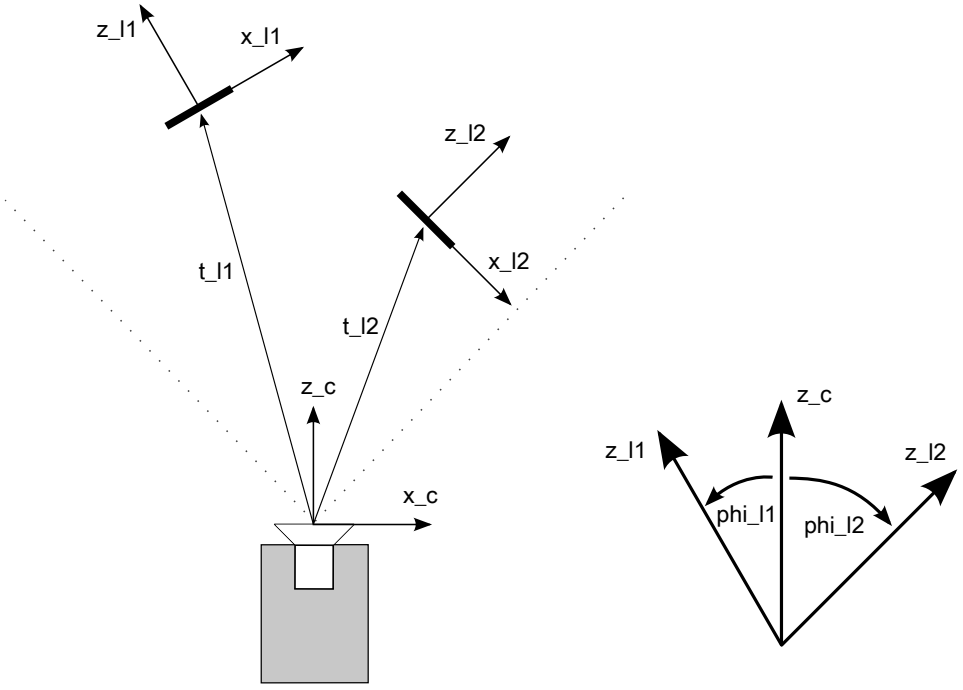

Figure 2.4: configuration of the environment

Chapter 3

Pose Estimation from video images

In times of unmanned navigation of vehicles and the increasing research in the field of mobile robots the estimation of the position and the orientation, just called pose estimation, of such a robot is a fundamental problem to be solved concerning other tasks as moving or object's tracking. Often this has to be done in real time depending the requirements of the system. To realize this challenge there are a couple of theoretical techniques available. The easiest way to estimate the pose is by means of odometry [22]. But for larger distances it results a grand error which can be improved combinating it with other sensors like in [7]. If besides the robots pose a map of the environment shall be created the socalled simultaneous localization and map building (SLAM) [17] and [1] can be applied. Another way is to use stationary active beacons, [13] and [15], emitting signals which are detected by a receiver on the robot. In this work the aim is to calculate the pose of a robot with an on-board camera on the basis of stationary landmarks shown in figure 2.3. Therefore the theory in [23] and [4] is used to develop an appropriate algorithm.
The chapter is organized as follows. In section 3.1 the algorithm for estimating the pose is presented. It is devided into 3.1.1 to 3.1.3 to demonstrate some mathematical tools and the geometrical conditions before starting with the derivation of the algorithm in 3.1.4. In the second part of this chapter another algorithm is proposed. Since there is ambiguity concerning the pose, means that there are more than one local minima of the error function, this algorithm chooses the one with the lowest error. Like in the section before, after showing the geometry in 3.2.1 the derivation of the algorithm is explained. In the following the expressions $\hat{(..)}$ represent the measured and therefore not noise free values.

3.1 Pose estimation

In this section the derivation of the algorithm to estimate the pose of a robot is the main topic. This is done by using at least four points of the landmark shown in figure 2.3. With these and the corresponding points detected in the image the algorithm operates to calculate the pose of the robot. But before, some other items have to be discussed which are needed to explicate the algorithm. At first a short explanation about the so called absolute orientation problem is given followed by the orthogonal projection. After that the geometrical conditions in which the system is resided. At the end of this section and past the explained terms the algorithm is presented.
In terms of notation it is said that in the following the expressions A and a describe matrices and vectors, respectively, without noise, whereas the expressions $\hat{A}$ and $\hat{a}$ are the observed or measured matrices and vectors.

3.1.1 Absolut Orientation Problem

The absolute orientation problem is the process of estimating the rotation and translation between two different coordinate systems given by the matrix R and the vector t. The estimation is based on the corresponding pairs of the camera space coordinates q and the object space coordinates p. It is supposed that p is known and q can be read out from the captured image. Another way to calculate the camera space coordinates is to use the relation between the systems given by

$$q = Rp + t. \tag{3.1}$$

With the goal obtaining R and t there can be further the least-squares problem proposed

$$\min_{R,t} \sum_{i=1}^{n} \|Rp_i + t - q_i\|^2 \tag{3.2}$$

where the minimization of the error with respect to R and t between the two obtained manners of determining q must be estimated. To solve the least-squares problem in a closed form there are various ways for example by using quaternions like in [12] or using the sigular value decomposition presented in [2] and [14]. In this work the sigular

value decomposition is used. In the following a short explaination of the solution of the absolute orientation problem is given. The equation 3.1 is supposed as the relation between p_i and q_i and the center of mass of the set of points is calculated to

$$\bar{p} = \frac{1}{n}\sum_{i=1}^{n} p_i \tag{3.3}$$

$$\bar{q} = \frac{1}{n}\sum_{i=1}^{n} q_i. \tag{3.4}$$

Further are defined

$$p_i' = p_i - \bar{p}, \quad q_i' = q_i - \bar{q} \tag{3.5}$$

as the distance of the particular points to the defined center and the matrix

$$M = \sum_{i=1}^{n} q_i' p_i'^t \tag{3.6}$$

which coincide with the sample cross-covariance matrix when multiplied by $\frac{1}{n}$. [2] shows that with the given conditions R^* and t^* minimize equation 3.2 and can be estimated like

$$R^* = \arg\max_R tr(R^t M) \tag{3.7}$$

$$t^* = \bar{q} - R^*\bar{p}. \tag{3.8}$$

Suppose the matrices U, Σ and V are the singular value decomposition of M, which means that acording to the theory in [2] they satisfy

$$U^t M V = \Sigma \tag{3.9}$$

where the columns of V represtent the eigenvectors of M^*M, the ones of U the eigenvectors of MM^* and the diagonal values of Σ the square roots of the eigenvalues that

correspond with the same columns in U and V. With 3.9 it can be estimated a rotation matrix

$$R^* = VU^t \tag{3.10}$$

which coincides whith R^* calculated in equation 3.7. So finally the searched rotation matrix which minimizes equation 3.2 can be obtained. Further the corresponding translation can be calculated according to equation 3.8.

3.1.2 Orthogonal Projection

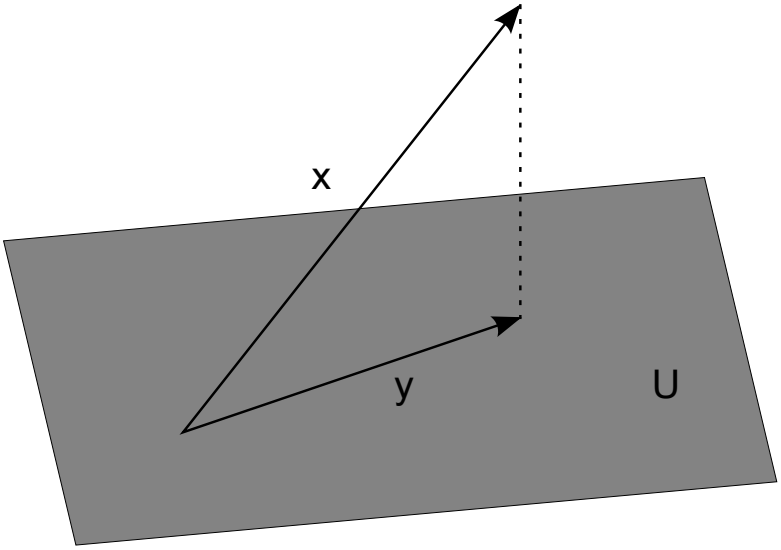

Figure 3.1: general orthogonal projection

The orthogonal projection describes a linear projection of a three dimensional vector upon a two dimensional subspace U where the view direction is orthogonal to the projection plane as shown in figure 3.1. The output vector of the projection is calculated according to

$$y = P_v x = \sum_{k=1}^{j} \frac{v_k' x}{v_k' v_k} v_k = \sum_{k=1}^{j} v_k \frac{v_k' x}{v_k' v_k} \tag{3.11}$$

The equal sign holds because of the linearity. The important step to get the projection matrix is to rewrite equation 3.11 as

$$y = P_v(x) = \sum_{k=1}^{j} \frac{v_k v_k'}{v_k' v_k} x \tag{3.12}$$

with the orthogonal projection matrix

$$V_i = \frac{v_i v_i^t}{v_i^t v_i}. \tag{3.13}$$

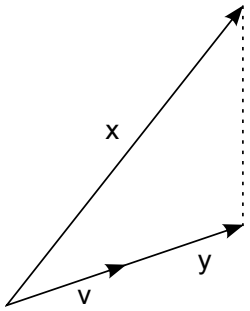

Figure 3.2: orthogonal projection upon a vector

where v_1,..,v_j is the orthogonal base of U. Generally speaking this means that the result of the projection is a vector composed of multiples of the subspace base vectors. Since here the subspace is a vector the result is just a multiple of v which is demonstrated in figure 3.2. The orthogonal projection has two extremal results, the zero vector if x and v are perpendicular and x itself if they point in the same direction.

3.1.3 Geometrical conditions

In order that the information provided by an image of a camera can be processed according to estimate the pose with respect to an object the geometrical conditions are to be clarified. This means that the relation between the camera and the object and its coordinate systems respectively. Figure 3.3 shows how the configuration in this work is

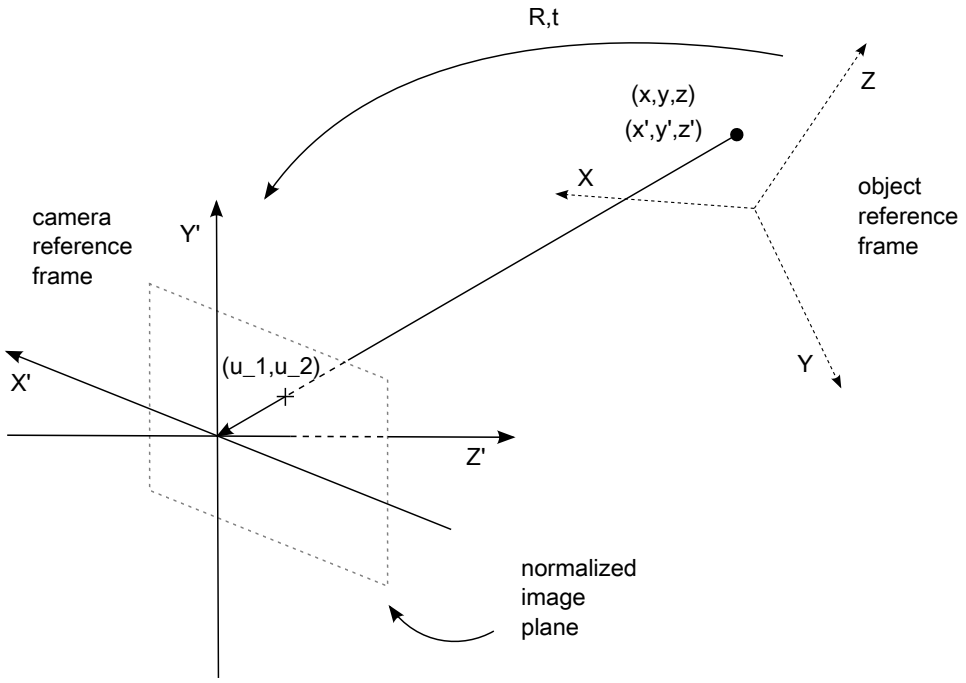

Figure 3.3: relation between camera and landmark

set. At this $p = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ specifies the three dimensional coordinate system of the landmark like it is brought out in figure 3.3, $q = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$ the one of the camera and $v = \begin{pmatrix} u_1 \\ u_2 \\ 1 \end{pmatrix}$ the two dimensional coordinate system of the so called normalized image plane. That describes the plane in the camera coordinate system with $z' = 1$. Hence the connection of p and q results in

$$q = Rp + t \tag{3.14}$$

whith the rotation matrix R and the translation vector t. Further the relation of q and v adds up to the projection equation acording to the pin-hole theory

$$v = \begin{pmatrix} \frac{x'}{z'} f_x + c_x \\ \frac{y'}{z'} f_y + c_y \\ 1 \end{pmatrix} \tag{3.15}$$

with the vector of the focal length

$$f = \begin{pmatrix} f_x \\ f_y \end{pmatrix} \tag{3.16}$$

and the vector

$$c = \begin{pmatrix} c_x \\ c_y \end{pmatrix} \tag{3.17}$$

describing the optical center of the camera. Since the relation of the coordinate systems is linear v_i, q_i and the optical center of the camera are collinear, shown in figure 3.3. Thus it can be proposed that the orthogonal projection of q_i on v_i should be q_i itself which can be expressed by the so called object space collinearity equation

$$Rp_i + t = V_i \left(Rp_i + t\right) \tag{3.18}$$

with

$$V_i = \frac{v_i v_i^t}{v_i^t v_i} \tag{3.19}$$

as the orthogonal projection matrix according to equation 3.13. Now all the information of the observed points v_i are stored in the matrices V_i. Due to measuring errors the collinearity is not given exactly and therefore equation 3.18 is not satisfied. Figuratively speaking, this means that the vectores q_i and v_i do not point exactly in the same direction. This error between the orthogonal projection $P_{v_i}(q_i)$ and q_i itself is called object space error

$$e_i = (I - \hat{V}_i)(Rp_i + t) \tag{3.20}$$

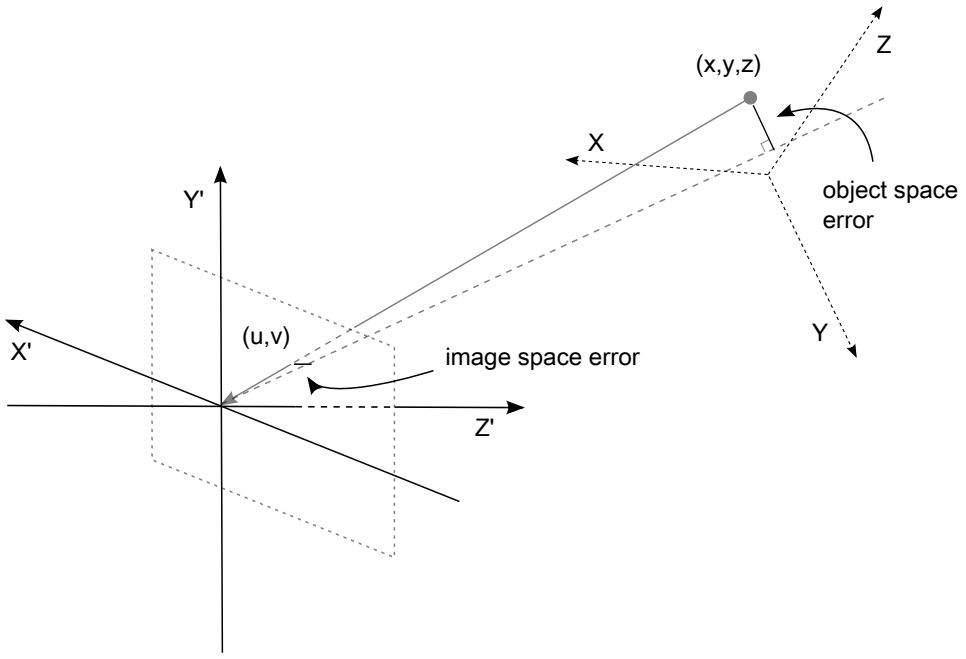

Figure 3.4: objectspace and image-space collinearity error

and is displayed in figure 3.4, where $\hat{V}_i$ is the observed orthogonal projection matrix which is defined in equation 3.19, R is the rotation matrix, t the translation vector and p_i are the points which represent the landmark. Taking the sum of the squared errors of all n pairs of points p_i and q_i the aim of the algorithm explained in the following section can be proposed to find a rigid transformation (R, t) to minimize the summation of the squared errors.

3.1.4 Algorithm

To begin with the derivation of the algorithm estimating the pose the object-space collinearity error from equation 3.20, which is explained in the recent section, is taken as the starting point. As explained in section 3.1.1, satisfying the conditions it is easy to find a solution for the absolute orientation problem as well as for the problem of estimation the pose of the robot. So starting with the error equation 3.14 the first step is to convert this error equation into a least square problem to be able to apply the SVD theory shown in 3.1.1. This is done by regarding the sum of the squared error

$$E(R,t) = \sum_{i=1}^{n} \|e_i\|^2 = \sum_{i=1}^{n} \left\| (I - \hat{V}_i)(Rp_i + t) \right\|^2 \tag{3.21}$$

with respect to the matrix R and the vector t. Now it is possible to obtain the pose by minimizing equation 3.21. With a fixed rotation R there can be calculated the optimal translation t like

$$t(R) = \frac{1}{n} \left(I - \frac{1}{n} \sum_j \hat{V}_j \right)^{-1} \sum_j \left(\hat{V}_j - I \right) Rp_j \tag{3.22}$$

so that in the following it is tried to find the minimization of equation 3.21 only with respect to the rotation R. Further it is attemped to convert equation 3.21 to an absolute orientation problem according to equation 3.2. In the first step the definitions

$$q_i(R) = \hat{V}_i \left(Rp_i + t(R) \right) \tag{3.23}$$

$$\bar{q}(R) = \frac{1}{n} \sum_{i=1}^{n} q_i(R), \tag{3.24}$$

where $\bar{q}(R)$ represents the center of mass of the set of camera frame points, are made in order to rewrite equation 3.21 to

$$E(R) = \sum_{i=1}^{n} \|Rp_i + t(R) - q_i(R)\|^2 . \tag{3.25}$$

Like this the equation 3.25 seems the same like equation 3.2 but still it isn't. That becomes apparent if the sample cross-covariance matrix

$$M(R) = \sum_{i=1}^{n} q_i'(R) p_i'^t \tag{3.26}$$

with

$$p'_i = p_i - \bar{p} \tag{3.27}$$
$$q'_i(R) = q_i(R) - \bar{q}(R) \tag{3.28}$$

is considered. It can be seen that M still depends on the rotation matrix R so that the SVD theory can not be applied and equation 3.25 can not be solved for R in a closed form. Another way to estimate R which minimize the error function is to calculate it iteratively. In the kth step of the iteration the following equations are given

$$t^{(k)} = t(R^{(k)}) \tag{3.29}$$
$$q_i^{(k)} = R^{(k)} p_i + t^{(k)}. \tag{3.30}$$

The rotation in the next iteration step is evaluated like

$$R^{(k+1)} = \arg\min_R \sum_{i=1}^{n} \left\| R p_i + t - \hat{V}_i q_i^{(k)} \right\|^2 \tag{3.31}$$
$$= \arg\max tr\left(R^t M(R^{(k)})\right) \tag{3.32}$$

where $t^{(k)}$ is not dependent of the actual rotation matrix $R^{(k+1)}$ but was calculated in the step before. So both $q_i^{(k)}$ and $M^{(k)}$ are independent of the actual rotation matrix $R^{(k+1)}$. Therefor in each iteration step the equations 3.31 and 3.32 respectively, can be considered as an absolute orientation problem and can be solved like shown in section 3.1.1. $\hat{V}_i q_i^{(k)}$ in equation 3.31 is regarded as a hypothesis of the set of scene points q_i in equation 3.2. Having found $R^{(k+1)}$ which minimizes equation 3.31 the new optimal translation vector can be updated to

$$t^{(k+1)} = t\left(R^{(k+1)}\right) \tag{3.33}$$

according to equation 3.22 and a new iteration step can be started. A final solution R^* is found if it represents a fixed point of equation 3.31 which means that the values of R don't change and

$$R^{(j)} = R^{(j+1)} \tag{3.34}$$

is satisfied. The abort criterion can also displayed as

$$R^* = \arg\min_R \sum_{i=1}^{n} \left\| Rp_i + t - \hat{V}_i(R^* p_i + t(R^*)) \right\|^2 . \tag{3.35}$$

3.2 Pose ambiguity

In this part of the work the problem of the pose ambiguity is discussed which can occure in the estimation of the robot pose according to two dimensional landmarks. The vision-based pose estimation is not very precise so that jitter, pose jumps and gross pose outliers can occure. Due to this uncertainty the corresponding error function of the algorithms may deliver two distinct local minima. And for that in many cases a wrong pose is estimated. The aim in this section is to derivate an algorithm which calculates all existing minima of the error function and choose the pose which minimizes it globally. The starting point of this algorithm is a first known pose (R, t) which is estimated with the pose estimation algorithm presented in sction 3.1. In the following the geometrical conditions of the camera landmark system are illustrated before presenting the derivation of the algorithm.

3.2.1 Geomatrical conditions

Figure 3.5 shows the two coordinate systems with their origins in the point C_C for the camera and C_M for the model which represents the landmark in our case. The relation between them is given by the rotation matrix R and the translation vector t which describes its distance. The optical axis of the camera points in the direction of the z-axis of the coordinate systems. On the right we see the model points P_1 and P_2 in the plane Π. By rotation the model about the y-axis for an angel α we obtain the model points $P_{1\alpha}$ and $P_{2\alpha}$ in the plane Π_α. These points are projected to the normalized image plane as v_1 and v_2. Like in the section 3.1.3 the idealiyed pinhole model is used to describe the relation between the camera and the landmark which leads to the same equations.
In sectoin II of [23] it is shown that for every angle α there may axist an angle β,

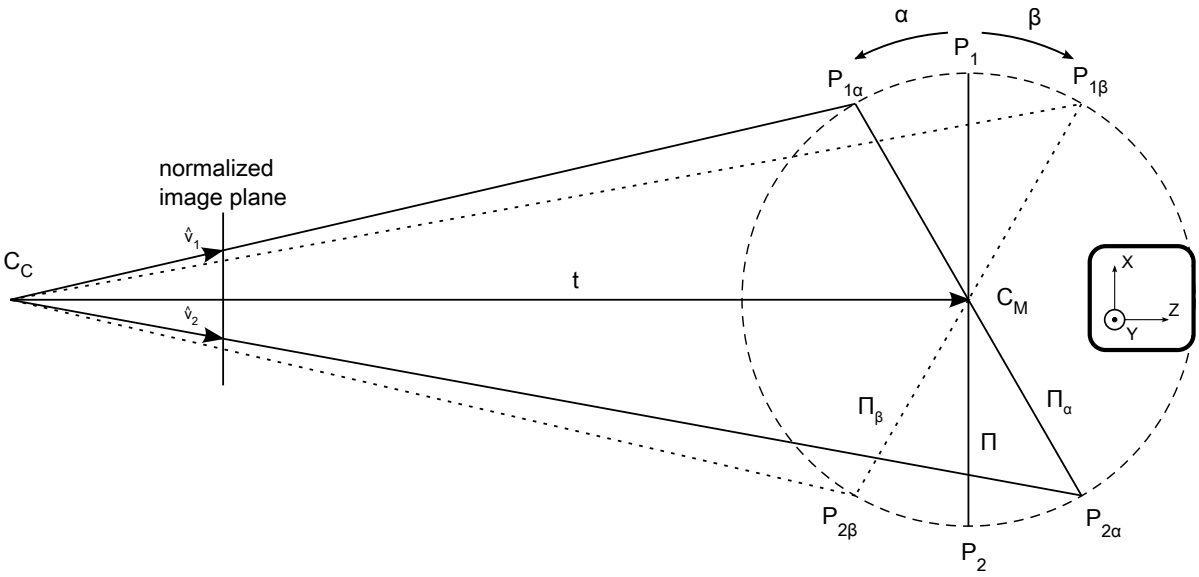

Figure 3.5: geometrical conditions

depending on the parameters, which also leads to a local minimum of an error function. Knowing this in the next section an algorithm is presented which estimates all the appearing minima of the corresponding error function.

3.2.2 Derivation of the algorithm

Derivating this algorithm it is begun with the same equations 3.14 and 3.22 like in the algorithm presented in section 3.1. It is tried to get an error function in which the rotation matrix R only depends on a rotation about the y-axis. To reach this there are applied two algebraic steps which are shown in the following. The relation between the object and the image points is

$$q = Rp + t \tag{3.36}$$

and the error function is

$$E(\hat{R}, \hat{t}) = \sum_{i=1}^{n} \left\| (I - \hat{V}_i)(\hat{R}p_i + \hat{t}) \right\|^2 \tag{3.37}$$

where

$$\hat{V}_i = \frac{\hat{v}_i \hat{v}_i^t}{\hat{v}_i^t \hat{v}_i} \tag{3.38}$$

is the observed orthogonal projection matrix with $\hat{v}_i$ describing the measurements of the vectors to the normal image plane. The first step in order to get to an error function which only depends on the rotation around the y-axis is to define a matrix R_t that satisfies

$$R_t \hat{t} = \begin{bmatrix} 0 \\ 0 \\ \|\hat{t}\| \end{bmatrix}. \tag{3.39}$$

This step is done because further on in equation 3.50 the rotation about the z-axis $R(\gamma)$ can be factored out from the error function so that only the desired rotation around the y-axis is left. Without loss of generality equation 3.36 can be rewritten as

$$\tilde{q} = \tilde{R}p + \tilde{t} \tag{3.40}$$

where

$$\tilde{q}_i = R_t \hat{q}_i \quad \tilde{t} = R_t \hat{t} \quad \tilde{R} = R_t \hat{R} \tag{3.41}$$

is defined. After that transformation the error function in equation 3.37 converts into

$$E(\tilde{R}, \tilde{t}) = \sum_{i=1}^{n} \left\| (I - \hat{V}_i)(\tilde{R}p_i + \tilde{t}) \right\|^2. \tag{3.42}$$

The second step is about to eliminate the part of R which represents the rotation about the x-axis the matrix $R_x(\alpha)$. Therefor the rotation matrix $\tilde{R}_z$ is introduced such that

$$E(\tilde{R}, \tilde{t}) = \sum_{i=1}^{n} \left\| (I - \hat{V}_i)(\tilde{R}\tilde{R}_z\tilde{R}_z^{-1}p_i + \tilde{t}) \right\|^2. \tag{3.43}$$

With this transformation it is further defined

$$\tilde{p}_i = \tilde{R}_z^{-1} p_i \tag{3.44}$$

which has as result a rotation of the objects points only about the z-axis. That doesn't change the fact that the model still stays planar with $z = 0$. In equation 3.43 remains the rotation matrix $\tilde{R}\tilde{R}_z$ which can be decomposed into the three elementary rotation about the the x, y and z axis respectively as follows

$$\tilde{R}\tilde{R}_z = R_z(\tilde{\gamma})R_y(\tilde{\beta})R_(\tilde{\alpha}). \tag{3.45}$$

Now the rotation $R_x(\tilde{\alpha})$ about the x axis can be eliminated by selecting $\tilde{R}_z$ such that $\tilde{\alpha} = 0$. With the modifications so far equation 3.36 has changed into

$$\tilde{q}_i = R_z(\gamma)R_y(\beta)\tilde{p}_i + \tilde{t} \tag{3.46}$$

and the error function 3.37 into

$$E(\gamma, \beta, \tilde{t}) = \sum_{i=1}^{n} \left\| (I - \hat{V}_i)(R_z(\gamma)R_y(\beta)\tilde{p}_i + \tilde{t}) \right\|^2. \tag{3.47}$$

As shown in section II of [23] there can ocurre two minima of the error function with respect to the rotation about the y-axis. Like it can be seen in equation 3.47 $E(\gamma, \beta, \tilde{t})$ still depends on the rotation about the z-axis. To eliminate this rotation so that the error function only depends on β it can be made use of the rotation matrix R_t which was indroduced in equation 3.39 in this section. From equation 3.39 is known that

$$\tilde{t} = \begin{bmatrix} 0 \\ 0 \\ \|\hat{t}\| \end{bmatrix} \tag{3.48}$$

and with $R_z(\gamma)$ as a rotation matrix about the z-axis it is effective that

$$R_z(\gamma)\tilde{t} = \tilde{t}. \tag{3.49}$$

Thus equation 3.46 can be rewitten as

$$\tilde{q}_i = R_z(\gamma)\left(R_y(\beta)\tilde{p}_i + \tilde{t}\right). \tag{3.50}$$

By placing $R_z(\gamma)$ outside the brackets it can be seen that it becomes a rotation about the optical axis of the camera and so it only effects image coordinates and doesn't change the geometric relation between the image plane and the landmark plane. With this step the final relation between image coordinates and landmark coordinates shown in equation 3.50 as well as the error function

$$E(\beta, \tilde{t}) = \sum_{i=1}^{n} \left\| (I - \hat{V}_i) R_z(\gamma)\left(R_y(\beta)\tilde{p}_i + \tilde{t}\right) \right\|^2 \tag{3.51}$$

are given and the estimation of the local minima can be started. At first the optimal translation $\tilde{t}_{opt}$ is calculated by setting the first derivative of $E(\beta, \tilde{t})$ with respect to $\tilde{t}$ to zero

$$\frac{\partial E}{\partial \tilde{t}} = 0. \tag{3.52}$$

Thereby we get a solution for $\tilde{t}_{opt}$ to

$$\begin{aligned} \tilde{t}_{opt}(\beta) &= \frac{1}{n}\left(I - \frac{1}{n}\sum_j \tilde{V}_j\right)^{-1} \sum_j \left(\tilde{V}_j - I\right) R_z(\gamma) R_y(\beta)\tilde{p}_j & (3.53)\\ &= G \sum_j \left(\tilde{V}_j - I\right) R_z(\gamma) R_y(\beta)\tilde{p}_j & (3.54) \end{aligned}$$

where G is a constant which depends only on measured image points $\tilde{v}_i$. By plugging in $\tilde{t}_{opt}(\beta)$ in equation 3.51 we obtain an error function which now only depends on β and the rotation about the y-axis respectively

$$E(\beta) = \sum_{i=1}^{n} \left\| (I - \hat{V}_i) R_z(\gamma) R_y(\beta)\tilde{p}_i + G \sum_j \left(\tilde{V}_j - I\right) R_z(\gamma) R_y(\beta)\tilde{p}_j \right\|^2. \tag{3.55}$$

Because it is very troublesome to calculate with trigonometrical functions

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{3.56}$$

is simplified by substituting

$$\beta_t = \tan(\frac{1}{2\beta}) \tag{3.57}$$

and we get

$$R_y(\beta_t) = \frac{1}{1+\beta_t^2} \begin{bmatrix} 1-\beta_t^2 & 0 & \beta_t \\ 0 & 1+\beta_t^2 & 0 \\ -2\beta_t & 0 & 1-\beta_t^2 \end{bmatrix}. \tag{3.58}$$

To estimate the optimal β_t we plug in equation 3.58 into equation 3.55 and after that the resulting error function is derivated such that

$$\frac{\partial E(\beta_t)}{\partial \beta_t} = 0 \tag{3.59}$$

which delivers an polynomial equation of degree four and can be easily solved. Naturally the solution of this polynomial consists of four values for β_t where two of them represent maxima and the other two minima. The two minima in which we are interested can be sorted out by using the condition

$$\frac{\partial^2 E(\beta_t)}{\partial \beta_t^2} > 0. \tag{3.60}$$

With the values of the solution for β_t we get by resubstituating according to equation 3.57 the solution for the rotation about the y-axis $R_y(\beta)_i$ and the translation $t(\beta)_i$. To obtain the pose $\hat{P}_i = (\hat{R}_i, \hat{t}_i)$ which describe the relation between the camera and the landmark the transformations of equation 3.45 and 3.41 are undone. These received poses are used as starting points for the algorithm presented in section 3.1 to get the final poses P_i^*. At last as the correct pose among the P_i^* is decided the one with the lowest value of the error function.

Chapter 4

Data fusion

The sensor based pose estimation for autonomous robots inherents some problem. If there is only the information of one sensor available you have to rely on its correctness. If there are various data at your disposal as possible different kinds of sensors you are able to minimize the error of estimation. Therefore it can be said that the more senors or information are available the more accurate the result will be. Of course there is the question of computing power and real time data processing which limits the use of an arbitrary amount of sensors. Needless to say that for fusing data from diffenent sensors or different data from the same sensor a separate algorithm is needed to combine these information. An introduction to this theory and some applications are given in [20] and [18]. There are a few approaches which give a solution of the fusion problem. One of them is stochastic weighting function according to the so called Monte Carlo method which its definition in [5] and some implementation in [25] and [3]. Another approach is the maximum likelihood method which estimates the stochastic characteristics according to a sample of the total [10]. It can also be applied to fuse sensor data [21] and [26]. The Bayesian probability theory like it is shown in [16], [8] and [6] can be seen as a method to reach the data fusion. Also filters can be used to solve this problem how it is done in this work. Here the data fusion is applied according to the kalman filter theory based on [11] and [19].
The first part of the chapter forms the estimation of the covariance in 4.1 which is needed in 4.2 to implement the kalman filter.

4.1 Covariance

Before the algorithm of fusing data according to the theory of Kalman can be applied it is needed to know the statistical distribution of the signals in form of the covariance matrix C_R. The covariance matrix represents some kind of weight to give more importance to those informations which are more precise. In our case these are the informations from the images captured closer to the landmark and with less distorsion to it. In general it is very difficult to account for all of the parameters which play a roll in the estimation of C_R. After taking some measurements and comparing whose results with the covariance matrix calculated with the algorithm it can be seen that they behave the same way. Therefore the calculation of this uncertainty matrix C_R is derived with the help of [24] which discusses the uncertainty model of robots with on board cameras observing artificial landmarks in general. The landmarks have the shape shown in figure 2.3.

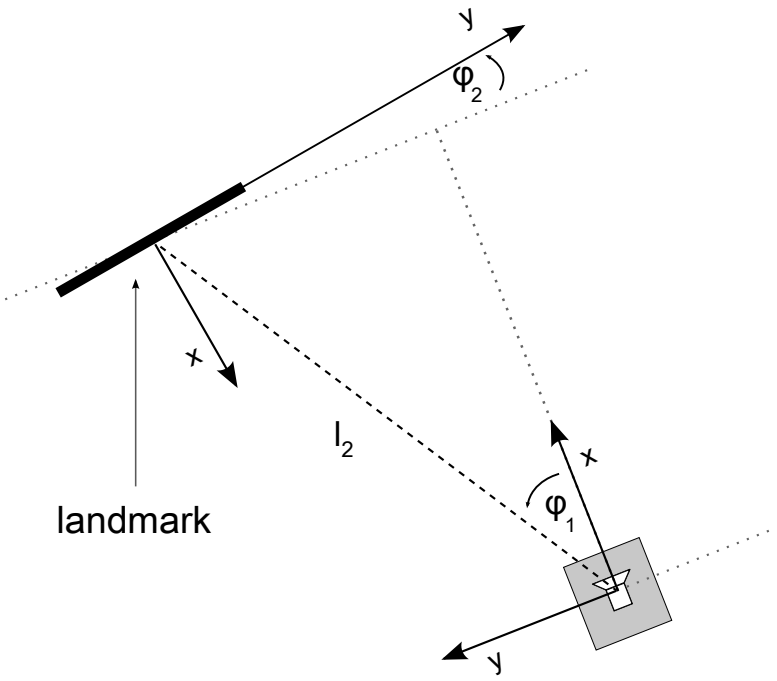

Figure 4.1: robot position relative to the landmark

In this section a manner of estimating the covariance matrix, as a measure of the uncertainty of the robot pose, is presented. In a first part the equations to calculate the pose

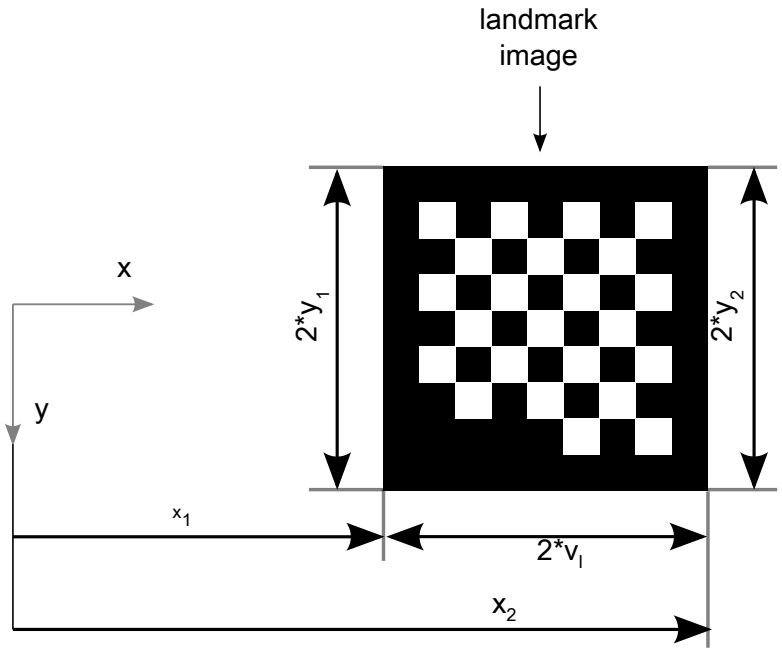

Figure 4.2: landmark image dimension

of the robot relative to the landmark are shown like it is demonstrated in figure 4.1. This way to calculate the robots position isn't used in combinatin with the algorithm derived in sction 3.2 because the translation vector and the rotation matrix are needed instead of just the distance and the angels φ_1 and φ_2.

At first the vector

$$P = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix} \tag{4.1}$$

with the x coordinates as the centers of the left and right boarder of the landmark image and the y coordinates as the half length of the left and right boarder. They can be read out of the image in the way illustrated in figure 4.2. [24] shows that with the vector P obtained from the captured landmark image the equations to calculate the pose of the robot result to

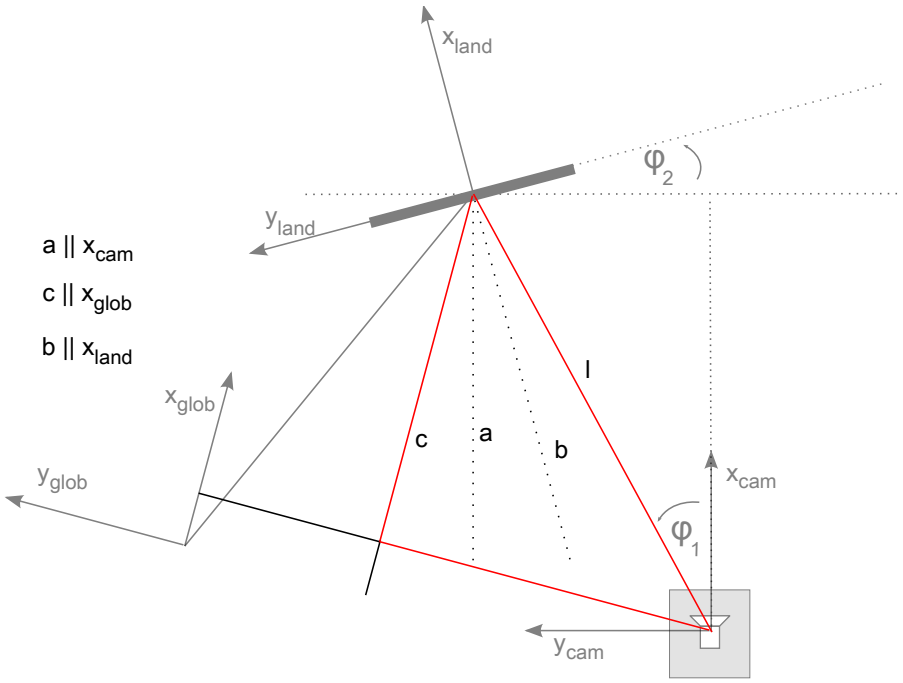

Figure 4.3: geometrical relations

$$l = v_l \frac{\sqrt{\lambda^2 + \left(\frac{x_1 y_2 + x_2 y_1}{y_1 + y_2}\right)^2}}{\frac{2 y_1 y_2}{y_1 + y_2}} \tag{4.2}$$

$$\varphi_1 = -\arctan\left(\frac{\frac{x_1 y_2 + x_2 y_1}{y_1 + y_2}}{\lambda}\right) \tag{4.3}$$

$$\varphi_2 = \arctan\left(\lambda \frac{y_1 - y_2}{-x_1 y_2 + x_2 y_1}\right) \tag{4.4}$$

where λ is the focal length and v_l the half width of the landmark frame, see figure 4.2. These transformations can be expressed as

$$L = f_{pl}(P, \lambda, v_l). \tag{4.5}$$

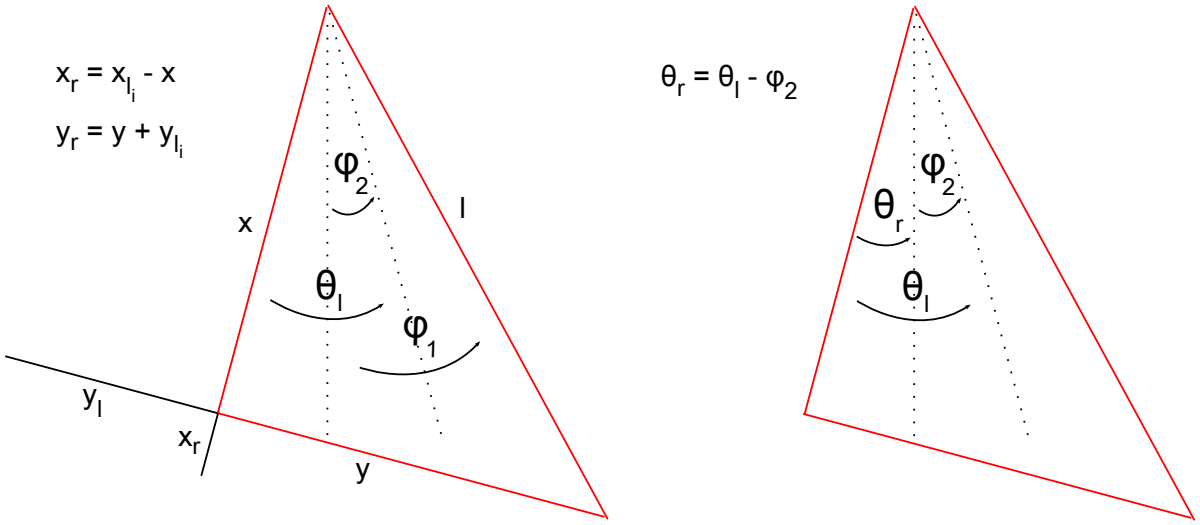

Figure 4.4: mathematical relations

The origin and orientation of the global coordinate system are described by the camera coordinate system at the point where the first image is captured. In figure 4.3 and 4.4 the deviation of the equations to calculate the robot pose in the global frame is shown. At first the geometrical relations a demostrated before the mathematical relations are given. So the location of the landmarks is known in the global frame and the location of the robot can be estimated to

$$x_r = x_{l_i} - l\cos(\phi_{l_i} + \varphi_1 - \varphi_2) \tag{4.6}$$
$$y_r = y_{l_i} + l\sin(\phi_{l_i} + \varphi_1 - \varphi_2) \tag{4.7}$$
$$\phi_r = \phi_{l_i} - \varphi_2. \tag{4.8}$$

where x_r and y_r are the coordinates of the robot and the origin of the camera frame respectively and ϕ_r is its orientation. x_{l_i}, y_{l_i} and ϕ_{l_i} mean the same with respect to the frame of the detected landmark. φ_1 and φ_2 are ilustrated in figure 4.3. The recent presented equations can be merged to

$$X_R = f_{lx}(X_{L_i}, L) \tag{4.9}$$

where

$$X_{L_i} = \begin{bmatrix} x_{l_i} \\ y_{l_i} \\ \phi_{l_i} \end{bmatrix} \tag{4.10}$$

is the vector of the coordinates of the i-th landmark in the global frame. So far the equations to locate the position of the robot are shown. This is not the main item in this section but they are needed to estimate the the uncertainty of the pose described by the covariance matrix C_R in which we are interested. C_R has the following general appearance

$$C_R = \begin{bmatrix} \sigma_x^2 & \sigma_{yx} & \sigma_{\phi x} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{\phi y} \\ \sigma_{x\phi} & \sigma_{y\phi} & \sigma_\phi^2 \end{bmatrix} \tag{4.11}$$

where the diagonal shows the statistical dispersions of the parameters and the non diagonal values describe the relation between the two different parameters. There are three parts which have to be observed concerning the uncertainties. At first there is the error of the calculation of the coordinates of the camera points P from equation 4.1 caused by the limited resolution. After this there is the calculation of the relation between the robot and the landmark and at last the estimation of the robots position in the global frame. Starting with the primary uncertainty a first matrix

$$C_P = \begin{bmatrix} \sigma_{x_1}^2 & 0 & 0 & 0 \\ 0 & \sigma_{y_2}^2 & 0 & 0 \\ 0 & 0 & \sigma_{x_2}^2 & 0 \\ 0 & 0 & 0 & \sigma_{y_2}^2 \end{bmatrix} \tag{4.12}$$

can be indicated. σ_x describes the horizontal discretization error and can be estimated as

$$\sigma_x = \frac{w'}{R_w} \tag{4.13}$$

with R_w as the horizontal resolution and w' as the width of the camera. On the other hand σ_y describes the vertical discretization error and can be estimated as

$$\sigma_y = \frac{h'}{R_h} \tag{4.14}$$

with R_h as the vertical resolution and h' as the height of the camera. The next step is to express the uncertainty of the transformation L given by the equations 4.2 to 4.4. This is done by the covariance matrix

$$C_L = \begin{bmatrix} \sigma_l^2 & \sigma_{\varphi_1 l} & \sigma_{\varphi_2 l} \\ \sigma_{l\varphi_1} & \sigma_{\varphi_1}^2 & \sigma_{\varphi_2\varphi_1} \\ \sigma_{l\varphi_2} & \sigma_{\varphi_1\varphi_2} & \sigma_{\varphi_2}^2 \end{bmatrix}. \tag{4.15}$$

Since the only variables of the transformation L are the points given in P the matrix C_L is estimated taken into account the recent shown insecurity C_P in equation 4.12. And because of the fact that the equations 4.2 to 4.4 are nonlinear further it has to be used the first order linearization

$$C_L = J_P C_P J_P^T \tag{4.16}$$

where J_P is the jacobian matrix of equation 4.5 defined as

$$J_P = \frac{\partial f_{pl}}{\partial P}. \tag{4.17}$$

The last step to reach the desired matrix C_R is to estimate the effect of the error distribution on the estimation of the position of the robot in the global frame given by 4.9. Again it is used the first order approximation caused by the same reason of nonlinearity of the transformation. So the final result yields to

$$C_R = J_L C_L J_L^T \tag{4.18}$$

where J_L describes the jacobian matrix of equation 4.9 defined as

$$J_L = \frac{\partial f_{lx}}{\partial L}. \tag{4.19}$$

4.2 Kalman

In this section the data fusion by means of the Kalman filter is shown. The base of the Kalman filter is the set of the state space models of the state and the measurement dynamics. Beginning with these equation the Kalman filter is able to give conclusions about the exact state from the knowledge of observations solely containing errors. To put it simply it just eliminates the noise caused by the measurement device. The Kalman filter can be applied in real time systems for example in the case of tracking or to fuse several information from different sensors.
At first there is explained the derivation in general of the Kalman filter. After that it is adapted to the static case before it is extended by considering the velocity to the dynamic case.

4.2.1 General derivation

At first the derivation of the common discrete kalman filter is presented here. The starting point of the kalman filter theory is a linear stochastic time discrete difference equation

$$x_k = Ax_{k-1} + Bu_k + w_k \tag{4.20}$$

where x_k is the actual state and the state transition matrix A gives the relation to the previous state x_{k-1}. The control-input matrix B describes the influence of the known input u_k. The unknown inputs w_k are assumed to be a stochastic signal with a normal distribution

$$p(w) \sim N(0, Q) \tag{4.21}$$

with zero mean and covariance Q. The second part of the system builds the measurement equation

$$z_k = Hx_k + v_k \tag{4.22}$$

with the actual measurement z_k, the observation matrix H which represents the relation between the state and the measurement. v_k is the term of the unknown signals of the measure equation also with a normal distribution

$$p(v) \sim N(0, R) \tag{4.23}$$

with zero mean and covariance R. The idea of the kalman filter is to predict a new state $\hat{x}_k^-$ at the step k at first which is called the *a priori* state estimate. In the second step it is corrected by means of the measurement to the so called *a posteriori* state estimate $\hat{x}_k$. Starting with the prediction, also called time-update, step the state in the k-th step yields to

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{4.24}$$

and the corresponding covariance to this expected value of the state is calculated to

$$P_k^- = AP_{k-1}A^T + Q. \tag{4.25}$$

In equation 4.24 $\hat{x}_k^-$ is just estimated from the known values of the state equation 4.20. The covariance P_k^- of the state prediction is computed from the covariance of the unknown process signals Q and the uncertainty P_{k-1} of the estimation of the previous state. The second step is the correction step or measurement-update. In order to be able to correct the predicted state the right way a weight matrix K is introduced

$$K_k = P_k^- H^T \left(HP_k^- H^T + R\right)^{-1} \tag{4.26}$$

giving more consideration to the real measurement or to its estimation. R specifies the covariance of the measurement noise in equation 4.22. With the Kalman gain the *a posteriori* state estimation it's error covariance can be calculated to

$$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H\hat{x}_k^-\right) \tag{4.27}$$

$$P_k = \left(I - K_k H\right) P_k^-. \tag{4.28}$$

With equation 4.26 to 4.28 the two extrema can be demonstrated which means an exact estimation or an exact measurement. If there is an exact estimation of the prediction

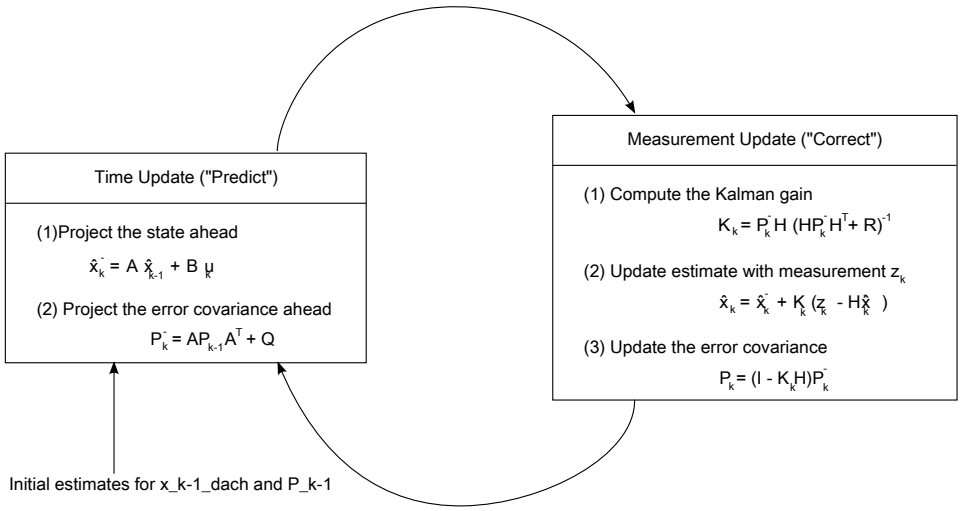

Figure 4.5: Kalman predictor-corrector

$\hat{x}_k^-$ the *a priori* estimate error covariance P_k^- tends to zero. For that the kalman gain converts to

$$\lim_{P_k^- \to 0} K_k = 0 \tag{4.29}$$

and therewith the equations 4.27 and 4.28 result to

$$\hat{x}_k = \hat{x}_k^- \tag{4.30}$$
$$P_k = P_k^-, \tag{4.31}$$

which means that there is no accumulation of the error. On the other hand if there is an exact measurement without error, means that R tends to zero the kalman gain yields to

$$\lim_{R \to 0} K_k = H^{-1} \tag{4.32}$$

which causes after equations 4.27 and 4.28 that

$$\hat{x}_k = z_k \tag{4.33}$$
$$P_k = 0. \tag{4.34}$$

For recapitulation figure 4.5 shows the whole circuit of the predictor-corrector in one diagram.

4.2.2 Static Kalman filter

To begin with the algorithm to fuse the information of two landmarks captured in the same image primarily the static version of the kalman filter will be used, because of the fact that every picked image can be seen as a new situation in which the pose has to be estimated. So the starting point are the two pose estimations z_1 and z_2 from the algorithm presented in chapter 3. The corresponding covariance matrices calculated in section 4.1 are σ_1 and σ_2. About the covariance is to say that it depends on the distance and the rotation between the landmark and the robot. The lower the value of the distance and the angel the lower are the values of the covariance matrix which means that the estimation is more precise. To assure that the estimation which is more exact is assessed with a greater value the weighting function results to

$$w = \sigma_1 \left(\sigma_1 + \sigma_2\right)^{-1} z_2 + \sigma_2 \left(\sigma_1 + \sigma_2\right)^{-1} z_1 \tag{4.35}$$

where w is the fused pose estimation. This means that if the measurement z_1 has a greater uncertainty and therefore greater values of σ the second measurement z_2 gets a higher weight. Further it can be seen that if the estimates are of equal precision the optimal estimation is just the mean of z_1 and z_2. The appropriate covariance matrix to w can be calculated to

$$\frac{1}{\sigma_w^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}. \tag{4.36}$$

The calculated uncertainty matrix σ_w is smaller than the smallest one of the estimations z_i. Now, if equation 4.35 is regarded more closely it converts into

$$w = z_1 + \sigma_1 \left(\sigma_1 + \sigma_2\right)^{-1} \left(z_2 - z_1\right) \tag{4.37}$$

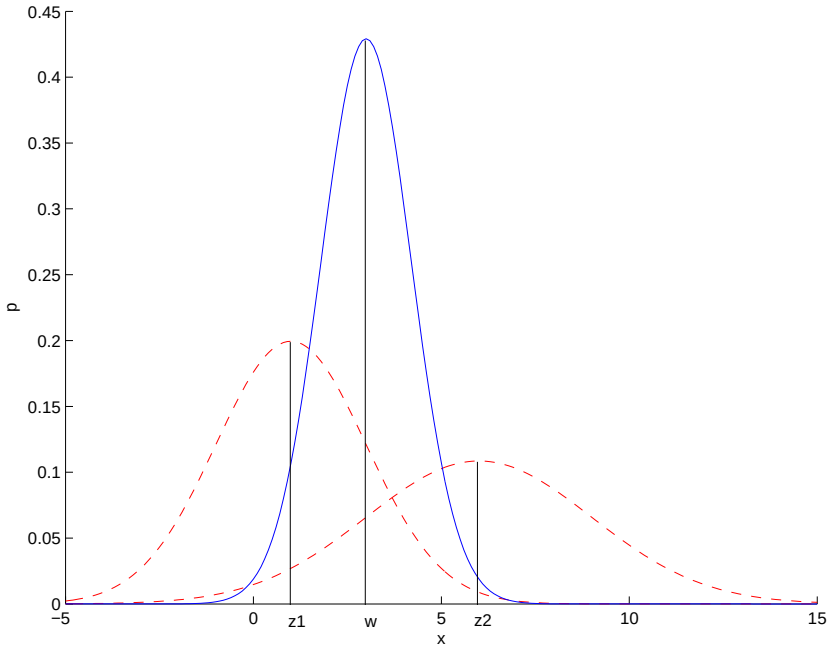

Figure 4.6: data fusion with Kalman filter

after outlining. Thus equation 4.37 can be considered as the *a posteriori* kalman filter equation 4.27

$$\hat{x}_k = \hat{x}_k^- + K_k \left(z_k - H\hat{x}_k^- \right) \tag{4.38}$$

with z_1 as $\hat{x}_k^-$ and the kalman gain defined as

$$K_k = \sigma_1 \left(\sigma_1 + \sigma_2 \right)^{-1} . \tag{4.39}$$

In figure 4.6 can be seen the fusion of two arbitrary signals z_1 and z_2 with their expected values $E(z_1) = 1$ and $E(z_2) = 6$ and the coresponding variances $\sigma_{z_1} = 2$ and $\sigma_{z_2} = 3$. The figure shows that the result w lies nearer to z_1 and has a smaller variance.

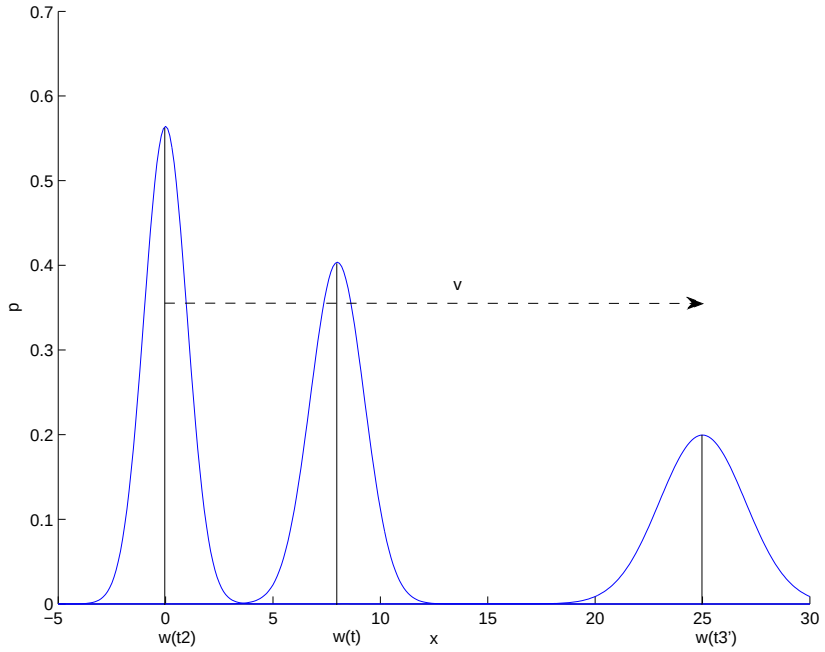

Figure 4.7: behavior considering velocity

4.2.3 Dynamic Kalman filter

So far just the static case is been observed which means that the kalman filter only combines the information of the two landmarks captured in one image.

In this case every image taken is considered as a new situation without taking in account the data of former estimations. With the dynamic kalman filter the velocity of the robot is calculated to make an *a priori* prediction of the pose which will be corrected in a following step by the fused information of the measurement, compare to figure 4.5. Since the time of each captured image is known the velocity can be calculated by the time difference of the two images and the covered distance. This results in

$$v_j = \frac{x_{j-1} - x_{j-2}}{t_{j-1} - t_{j-2}} \tag{4.40}$$

where v_j is the velocity used for the estimation in the actual step j and x and t are the

position and the time, respectively, of the two former steps. Because of the fact that the time information is given the uncertainty of the velocity estimation is just the addition of the cavariance matrices of the pose estimations

$$\sigma_{v_j} = \sigma_{z_{j-2}} + \sigma_{z_{j-1}}. \tag{4.41}$$

With the estimated velocity and its covariance matrix the pose prediction results into

$$\mu_{pred,j} = x_{j-1} + v * (t_j - t_{j-1}) \tag{4.42}$$

and the corresponding covariance

$$\sigma_{\mu_{pred,j}} = \sigma_{j-1} + \sigma_v. \tag{4.43}$$

The next step is just like explained in subsection 4.2.2. The prediction of the pose as result of the equation 4.42 is combined with the fused pose estimation from the landmarks. This happenes according to the equations 4.35 and 4.36. Figure 4.7 shows the behavior of the covariance during time. Here $w(t_2)$ is the *a posteriori* value of the Kalman filter fusion at the time step 2. With the time passing the prediction of the position gets more and more imprecise until the t_3' which is the time right before the next measurements are taken.

Chapter 5

Simulation results

Figure 5.1: robot - landmark setting

In this final chapter the simulation results of the recent shown algorithm are presented. The simulation was operated by GNU/Octave and Matlab. At first, in section 5.1, there is demonstrated the performance of the pose estimation algorithm derived in chapter 3. After that the estimated odometry of the robot by means of the mentioned algorithm is compared with the measured odometry only considering one landmark. And finally in section 5.3 the results of the algorithm with and without using the data fusion of two landmarks realized by a Kalman filter is illustrated.

5.1 Pose estimation

In the first section of the chapter there are presented the results of the algorithm to estimate just the position and orientation of the robot to one landmark. The robot or the camera respectively stands still and doesn't cover any distance. In figure 5.1 the setup of the robot landmark composition is demonstrated. The exact distance to the landmark is measured with a laser distance meter, see figure 5.1.

Figure 5.2: distance camera - landmark: 1m

Figure 5.3: distance camera - landmark: 1.5m

The results can be seen in figures 5.2 to 5.6 where the distances of the camera to the landmark are 1m, 1.5m, 2m, 2.5m and 3m. The output of the algorithm is represented by the vector

$$Q = \begin{bmatrix} 1.01876 \\ 1.52708 \\ 2.03572 \\ 2.54425 \\ 3.0522 \end{bmatrix}, \tag{5.1}$$

so that the error yields to

Figure 5.4: distance camera - landmark: 2m

Figure 5.5: distance camera - landmark: 2.5m

Figure 5.6: distance camera - landmark: 3m

$$E = \begin{bmatrix} 0.01678 \\ 0.01962 \\ 0.02472 \\ 0.03525 \\ 0.0522 \end{bmatrix} \tag{5.2}$$

in meter.

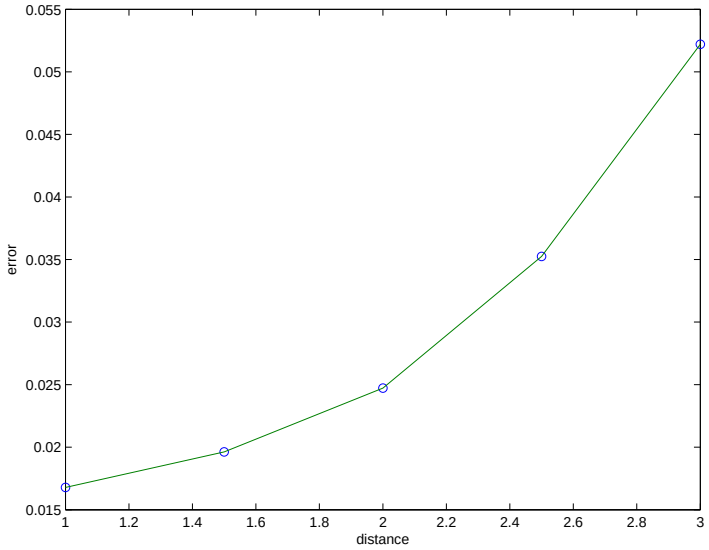

Figure 5.7: error chart

If we examine figure 5.7 it can be seen that the error of pose estimation algorithm grows exponentially acording to the growing distance between camera and landmark. The influence of the angle can be seen if the figure 5.3 and 5.8 are compared. Both have the distance of 1.5m to the landmark. But figure 5.8 delivers 1.6452 as the value of the distance and so a greater error. The fact that the algorithm has an error already from short distances and that this error has always the same sign does not play a significant role because the aim of the work is to estimate the odometry of the covered distance. Therefore the diferences between results are taken and so the errors cancel each other.

Figure 5.8: distance camera - landmark: 3m angular

5.2 Estimated and measured odometry with one landmark

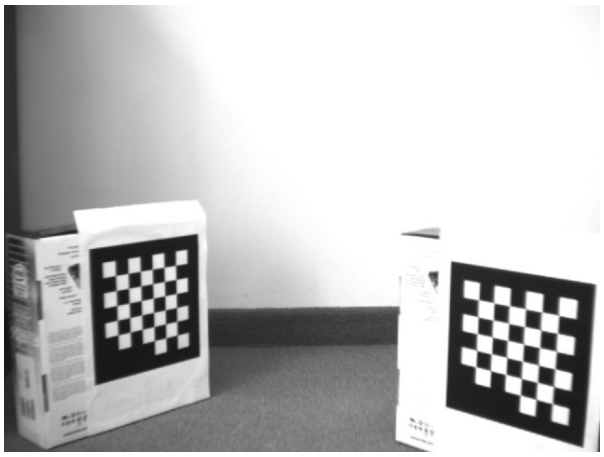

Figure 5.9: first image of the sequenz

Figure 5.10: second image of the sequenz

In this second section the estimation of the odometry is examined. Therefore the pictures taken by the camera are the input to the algorithm and its result is compared to the measurement of the device shown in figure 5.1. In figure 5.9 to 5.13 a few positions of the robot are illustrated during its motion away from the landmark to give an understanding

of the path covered by the robot. In this case just one of the two landmarks are taken into account for the estimation. So it can be compared the results of not using data fusion and the case of using it, presented in section 5.3. The movement done by the robot is to start close by the landmark and move straight backwards.

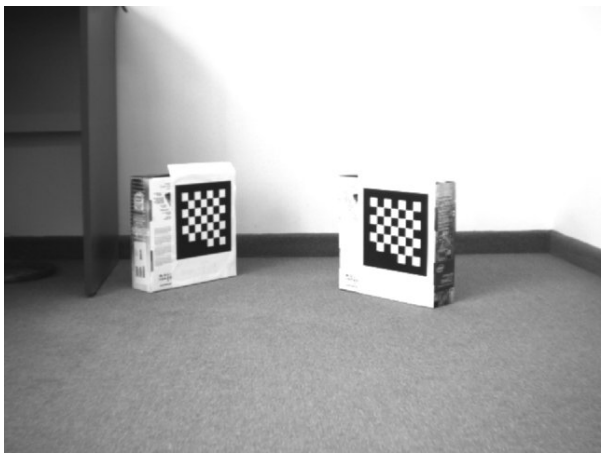

Figure 5.11: third image of the sequenz

Figure 5.12: fourth image of the sequenz

Figure 5.13: last image of the sequenz

Figure 5.14 shows the lateral movement towards the x-axis of the camera frame. As can be seen, the values stay quite close to zero. That's because the robots execute a linear movement backwards along the negative direction of the z-axis. This is shown in figure 5.15 The curve starts at zero and proceeds like a straight with a constant slope. In the

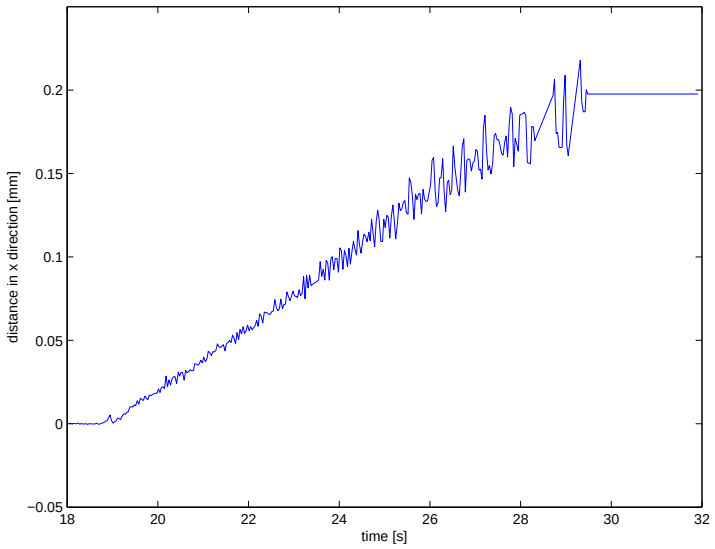

Figure 5.14: developing of the x-coordinate

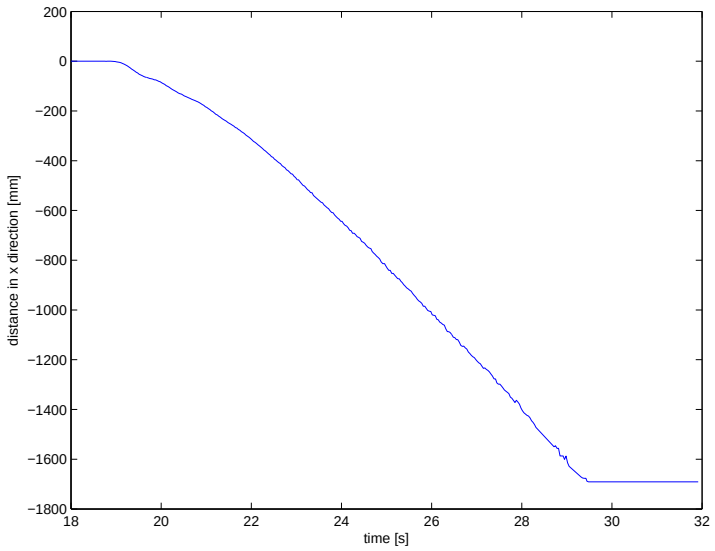

Figure 5.15: developing of the z-coordinate

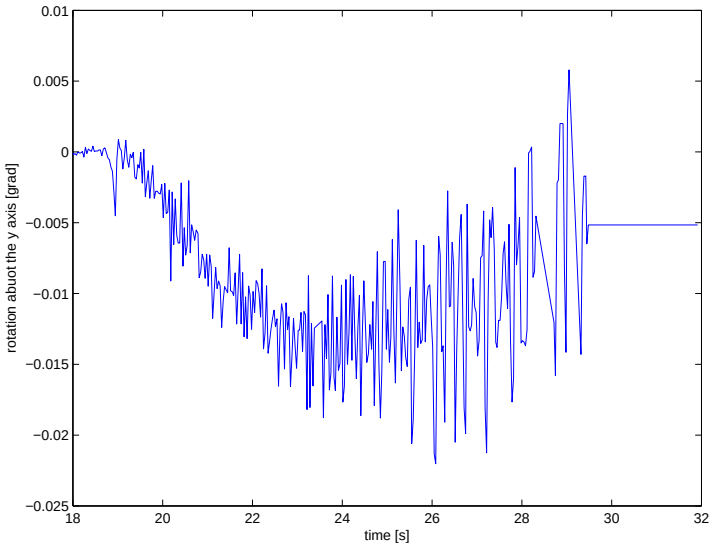

Figure 5.16: developing of the angle φ

experiment it was tried to drive the robot at a constant velocity. The last figure, 5.16, demonstrate the developing of the rotation angle φ. Here it is the same reason like with the x-axis. The value of φ stays around zero because of the motion without turning. The initial and the final value of the estimated pose of the robot deliver a result for the distance covered of

$$Odo_{est} = 1.6911m. \tag{5.3}$$

The odometry measured is

$$Odo_{meas} = 1.861m. \tag{5.4}$$

And so the odometry using only one landmark gives an error of 0.0913 and 9.13% respectively.

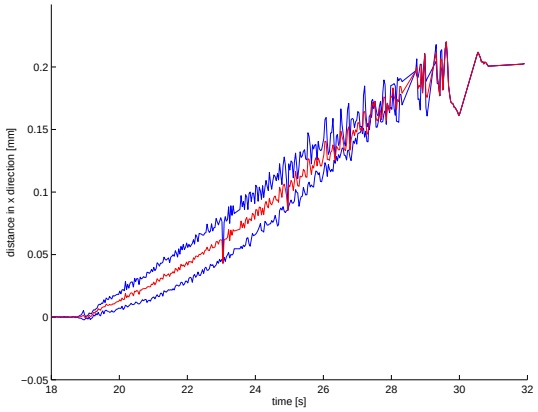

Figure 5.17: data fusion from two landmarks (x-direction)

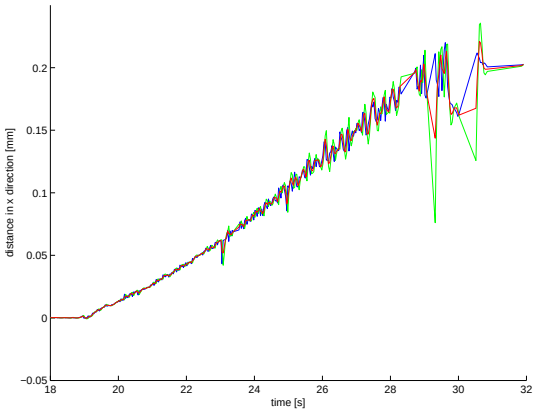

Figure 5.18: data fusion of the velocity estimation and the landmark fusion (x-direction)

5.3 Estimated and measured odometry by means of data fusion

At last the simulation results using the data from two landmarks fused by the Kalman filter are shown in section 4.2. Here the same sequence of pictures is used as in section 5.2. The diference is that now the information of both landmarks is used, if they are detected. In the figures 5.17 and 5.18 the process of the movement in direction of the x-axis is illustrated. It describes the displacement to the left or right respectavely acording to the starting point. Since the covered distance is done in form of a straight line the values of the x coordinate should stay in a small area around the zero. In figure 5.17 the resultant graph is the fusion, red, of the information from the two landmarks, blue. Figure 5.18 shows the result of the fusion, red, combined with the pose prediction by means of velocity estimation. In figure 5.17 the data of the two landmarks is replanished well according to the fusion. In the right figure the fusion with the estimated velocity is disturbed at the end of the simulation because the robot was driven by remote control and so stopped abruptly.

In figure 5.19 and 5.20 the results concerning the z coordinate are expressed. This is the direction of the robot movement away from or to the position of the first picture taken. This graph should demonstrate a nearly linear curve because the robot was tried to drive with a canstant velocity. Like in the case of the x coordinate the left figure shows

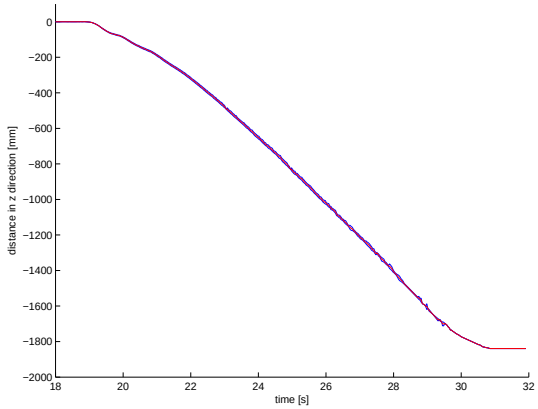

Figure 5.19: data fusion from two landmarks (z-direction)

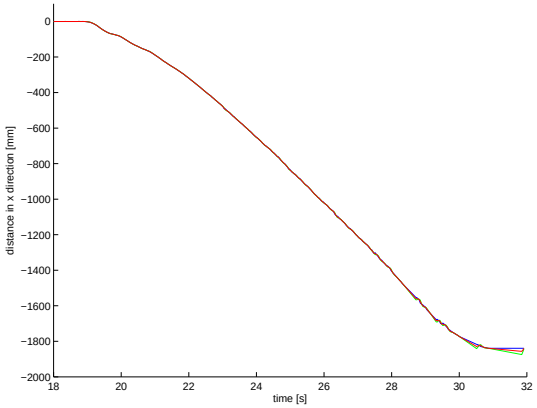

Figure 5.20: data fusion of the velocity estimation and the landmark fusion (z-direction)

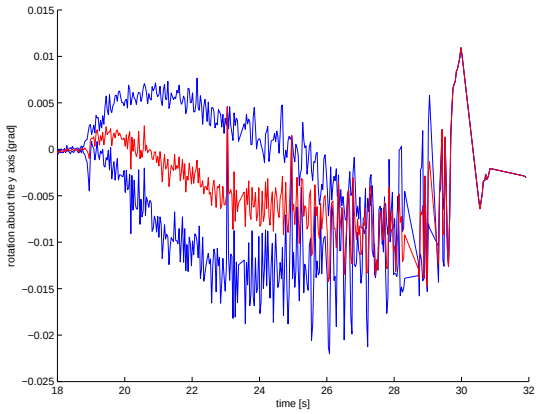

Figure 5.21: data fusion from two landmarks (rotation about y-axis)

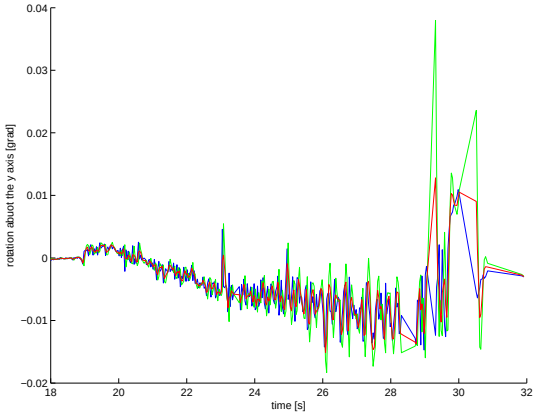

Figure 5.22: data fusion of the velocity estimation and the landmark fusion (rotation about y-axis)

the fusion without considering the velocity estimation. In the right one the calculated velocity as prediction is fused with the result given by the two landmarks. It can be seen that during this motion the pose estimation according to the two landmarks as well as to the velocity estimation is very good because there is only a little difference between the curves. This shows the accurate work of the algorithm calculating the distance.

In the last two figures, 5.21 and 5.22, the angle of rotation is shown. Like in the figures before figure 5.21 shows the fusion of the estimated poses acording to the landmarks and figure 5.22 illustrates the combination of this result with the angular velocity. Like said before, since it is a srtraight movement of the robot the angle value has to stay around zero. Figure 5.21 is a good example of the data fusion of the two landmarks. With the help of the fusion the angle dynamic results to the espected signal around zero.
The desired result of the odometry of the robot is estimated

$$Odo_{est} = 1.8403 \tag{5.5}$$

and measured

$$Odo_{meas} = 1.861. \tag{5.6}$$

And so the estimated result has only an error of 0.0111 or 1.11%. If the figures 5.20 and 5.15 and the equations 5.5 and 5.3 respectively are compared it can be seen that the estimation with the help of data fusion gives a better result. To give a review of the simulation results shown in this chapter it can be said that the algorithm to estimate the pose of a robot gives, absolutely seen, only sufficient performance. This means that if only one picture is taken the error of the pose estimation is quite big. But if the purpose is to calculate a distance covered by or the odometry of a vehicle it delivers a good result because the diference of the measurements are taken. And further it can be improved by taking the information of the velocity estimation into account.

Chapter 6

Conclusion

In this thesis an important problem in the field of robot science is discussed. A vehicle equipped with a camera for orientation is the object of interest, see figure 2.1. The posed aim was to apply and examine an algorithm to estimate the distance and rotation of the robot in relation to a landmark, figure 2.3, and further to estimate the odometry of the distance travelled by the robot. Therefore two papers are taken as the basis to realize the postulated algorithm. In the first step there is calculated the relative pose of the robot in respect of the landmark. As result it marks a minimum of an error function. Since in some cases there can occur two local minima the second algorithm guarantees that the correct one is chosen as result. Using the relative position at each moment where a picture is taken the distances between them can be calculated. If the first pose is set as initial point the odometry of the robot can be estimated. To reach a maximum degree of accuracy there is used the Kalman Filter theory to fuse data from different origins.

The topics that could not be realized in this work can be seen as an outlook of further works. The algorithm can be expanded to treat a robot which drives an arbitrary path. Which means that it has to handle curves and changes in the velocity. If larger distances are covered another issue is to manage the crossing from one landmark to another and to consider the case if one gets lost or isn't detected.

Affirmation

Hereby I affirm that I wrote the present thesis without any inadmissible help by a third party and without using any other means than indicated.

This thesis has not been presented to any other examination board in this or a similar form, neither in Germany nor in any other country.

Stuttgart, December 22, 2009 _____

List of Figures

List of Tables

Bibliography

[1] Andrea Garulli, Antonio Giannitrapani, Andrea Rossi Antonio Vicino: *"Mobile robot SLAM for line-based environment representation"*. IEEE Conference on Decision and Control, and the European Control Conference, 2005.

[2] Berthold K. P. Horn, Hugh M. Hilden, Shahriar Negahdaripourt: *"Closed-form solution of absolute orientation using orthonormal matrices"*. Journal of the Optical Society of America, 1988.

[3] Carine Hue, Jean-pierre Le Cadre, Patrick Perez: *"Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion"*. IEEE Transactions on Signal Processing, 2002.

[4] Chien-Ping Lu, Gregory D. Hager, Eric Mjolsness: *"Fast and Globally Convergent Pose Estimation from Video Images"*. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2000.

[5] Christian P. Robet, George Casella: *"Monte Carlo Statistical Methods"*. Springer, 2005.

[6] Christophe Coué, Thierry Fraichard, Pierre Bessiere Emmanuel Mazer: *"Multi-Sensor Data Fusion Using Bayesian Programming : an Automotive Application"*. hyper articles en ligne, 2002.

[7] Cox, Ingemar J.: *"Blanche- An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle"*. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION,, 1991.

[8] D. Fasbender, D. Fasbender, P. Bogaert A. Dassargues: *"Bayesian data fusion applied to water table spatial mapping"*. Water resources research, 2008.

[9] David Gaydou, Gonzalo Perez Paina, Javier Salomone Guillermo Steiner: *"Plataforma móvil de arquitectura abierta"*. V Jornadas Argentinas de Robótica, 2008.

[10] Eliason, Scott R.: *"Maximum Likelihood Estimation: Logic and Practice"*. Sage Publications, 1993.

[11] Grag Welch, Gary Bishop: *"An introduction to the Kalman Filter"*. SIGGRAPH, 2001.

[12] Horn, Berthold K. P.: *"Closed-form solution of absolute orientation using unit quaternions"*. Journal of the Optical Society of America, 1987.

[13] Jaemu Yun, Sungbu Kim, Jangmyung Lee: *"Robust Positioning a Mobile Robot with Active Beacon Sensors"*. International Conference on Knowledge-Based Intelligent Information and Engineering Systems, 2006.

[14] K. S. Arun, T. S. Huang, S. D. Blostein: *"Least-Squares Fitting of Two 3-D Point Sets"*. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 1987.

[15] Kleeman, Lindsay: *"Ultrasonic autonomous robot localisation system"*. IEEE/RSJ International Workshop on Intelligent Robots and Systems, 1989.

[16] Lawrence D. Stone, Thomas L. Corwin, Carl A. Barlow: *"Bayesian Multiple Target Tracking"*. Artech Print on Demand, 1999.

[17] M. W. M. Gamini Dissanayake, Paul Newman, Hugh F. Durrant-Whyte M. Csorba: *"A Solution to the Simultaneous Localization and Map Building (SLAM) Problem"*. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, 2001.

[18] Martin E. Liggins, David L. Hall, James Llinas: *"Handbook of Multisensor Data Fusion: Theory and Practice"*. Crc Pr Inc, 2008.

[19] Maybeck, Peter S.: *"Stochastic Models, Estimation and Control"*. Academic Pr, 1979.

[20] Mitchell, H. B.: *"Multi-Sensor Data Fusion: An Introduction"*. Springer, 2007.

[21] Piero Gerbino, Msaruf Ali: *"A Novel Local Likelihood Approach to Data Fusion in Passive Target Tracking"*. IEEE Colloquium. Target Tracking: Algorithms and Applications, 1999.

[22] Polotski, V.: *"Vehicle position estimation using structured light and odometry"*. IFAC international workshop on motion control, 1998.

[23] Schweighöfer, Gerald and Axel Pinz: *"Robust Pose Estimation from a Planar Target"*. IEEE[PAMI], 2005.

[24] Skrzypczynski, Piotr: *"Uncertainty models of vision sensors in mobile robot positioning"*. Int. J. Appl. Math. Comput. Sci., 2005.

[25] Y. F. Wong, J. K. Wu, L. H. Ngoh W. C. Wong: *"Collaborative Data Fusion Tracking in Sensor Networks using Monte Carlo Methods"*. Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, 2004.

[26] Yifeng Zhou, Henry Leung: *"Maximum-likelihood approach for multisensor data fusion applications"*. SPIE proceedings series, 1998.