

Simultaneous Localization and Mapping (SLAM) of a Mobile Robot Based on Fusion of Odometry and Visual Data Using Extended Kalman Filter

André M. Santana[†] and Adelardo A. D. Medeiros[‡]

[†] *Federal University of Piauí – UFPI*

Department of Informatics and Statistics – DIE

Teresina - Piauí - Brasil

[‡] *Federal University of Rio Grande do Norte – UFRN*

Department of Computer Engineering and Automation - DCA

Natal - Rio Grande do Norte - Brasil

1. Introduction

The term SLAM is used as an abbreviation for Simultaneous Localization and Mapping, and was originally developed by Leonard & Durrant-Whyte (1991) based on previous work by Smith et al. (1987). In the SLAM problem, a mobile robot uses its sensors to explore the environment, gains knowledge about it, interprets the scenario, builds an appropriate map and then calculates its location relative to it. The maps can be illustrated in several ways, such as occupation grids and characteristic maps. We are interested in the second illustration. A detailed theoretical description about the topic can be found in Durrant & Bailey (2006).

In addition to perception reliability, for the general acceptance of applications, the technologies used must provide a solution at a reasonable cost, that is, the components must be inexpensive. A solution is to use optical sensors in the robots to solve environment perception problems. Due to the wide use of personal digital cameras, cameras on computers and cell phones, the price of image sensors has decreased significantly, making them an attractive option. Furthermore, the cameras can be used to solve a series of key problems in robotics and in other automatized operations, as they provide a large variety of environmental information, use little energy, and are easily integrated into the robot hardware.

The main challenges are to take advantage of this powerful and inexpensive sensor to create reliable and efficient algorithms that can extract the necessary information for the solution of problems in robotics. When cameras are used in robots as the main sensors for solving SLAM problems, the literature uses the term visual SLAM to denote this process, the objective of this study.

The major challenges in visual SLAM are: a) how to detect characteristics in images; b) how to recognize if a detected characteristic is or is not the same as one previously detected; c)

how to decide if a newly detected characteristic will or will not be adopted as a new landmark; d) how to calculate the 3D position of the landmarks from 2D images; and e) how to estimate the uncertainty associated with the calculated values. In general, all of these aspects must be resolved. However, in special situations, it is possible to develop specific strategies to overcome all of these problems. This is the aim of this work.

The system that will be presented shows a visual SLAM technique equipped for flat and closed environments with floor lines. This is not a very limiting pre-requisite, as many environments such as universities, shopping malls, museums, hospitals, homes and airports, for example, have lines as floor components.

The algorithm used in visual SLAM is based on the Extended Kalman Filter (EKF), to allow the robot to navigate in an indoor environment using odometry and pre-existing floor lines as landmarks. The lines are identified using the Hough transform. The prediction phase of EKF is done using the geometric model of the robot. The update phase uses the parameters of the lines detected by the Hough transform directly in Kalman's equations without any intermediate calculation stage. Using existing lines as landmarks reduces the total complexity of the problem as follows: a) lines can be easily detected in images; b) floor lines are generally equally well spaced, reducing the possibility of confusion; c) since the number of lines in the images are not very large, each new line detected can be defined as a new landmark; d) a flat floor is a 2D surface and thus there is a constant and easy-to-calculate conversion matrix between the image plane and the floor plane, with uncertainties about 3D depth information; and e) after processing the number of pixels in the image that belong to the line is a good reliability measure of the landmark detected.

2. Related Work

Recent extensions to the overall SLAM problem have focused on the possibility of using cameras instead of sonar or laser. Examples are the works of Davison et al. (2002) , Jung (2004) and Herath et al. (2007), using stereo vision, as well as the studies conducted by Davidson et al. (2004) and Kwok et al. (2005), using a single camera.

Mansinghka (2004) presented a visual SLAM for dynamic environments using the SIFT transform and optical flow. Estrada et al. (2005) proposed a hierarchical mapping method that enables the obtaining of accurate metric maps of large environments in real time. The lower level of the map is composed of a set of local maps that are statistically independent. The upper part of the map is an adjacency graph whose arches are labeled with the relationship between the location of the local maps and a relative estimate of these local maps is kept at this level in a stochastic relation.

A solution using geometric information of the environment is proposed by Chen (2006). He reports that redundancy in SLAM may reinforce the reliability and accuracy of the characteristics observed and for this reason the geometric primitives common in indoor environments, lines and squares, for example, are incorporated into an Extended Kalman Filter (EKF) to raise the knowledge level of the characteristic observed.

Frintrop et al. (2006) introduced a new method to detect landmarks that consists of a biologically inspired attention system to detect contrasting regions in the image. This approach enables the regions to be easily redetected, thereby providing more ease of communication. Dailey & Parnichkun (2006) used stereo vision for a visual SLAM based on a particle filter. Choi et al. (2006) used an approach based on sonar information with stereo

vision in an extended Kalman filter. In this work, object recognition is accomplished with Harris corners, using SIFT and RANSAC to eliminate false-positives.

Automatic recognition and a record of objects as visual landmarks are proposed by Lee & Song (2007). SIFT transform and contour algorithms are used to distinguish objects in the background of the image. When objects are detected and considered adequate for robot navigation, they are stored for later use to correct position.

Clemente et al. (2007) demonstrated for the first time that SLAM with a single camera providing input data can reach a large scale outdoors while functioning in real time. Jing Wu & Zhang (2007) conducted a study on a camera model for visual SLAM. The focus of this work is on how to model optical sensor uncertainty and how to build probabilistic components of the camera model. The deterministic component of the camera calibration process, an intrinsic parameter, is used to re-project the error. The errors are then found according to bivariate Gaussian distribution and the measure of covariance can be calculated when the characteristics are at different distances from the camera.

More recent studies in visual SLAM address several specific points. Esteban (2008) attacked the problem of illumination for omnidirectional vision. Steder et al. (2008) studied visual SLAM for aerial vehicles and Angeli et al. (2008) investigated the closed-loop problem. The latter used color information to resolve the closed-cycle problem with a voting system.

Lemaire & Lacroix (2007) proposed the use of 3D lines as landmarks. They report the following advantages of using 3D lines: first, these primitives are very numerous in indoor environments; second, in contrast to sparse point maps, which are only useful for location purposes, a relevant segmentation map provides information on the structure of the environment. Also using lines, in this case, vertical, Fu et al. (2007) carried out a study on the fusion of laser and camera information in an extended Kalman filter for SLAM. In this work, the lines are extracted from the image using Canny. Ahn et al. (2007) built a map with characteristics of 3D points and lines for indoor environments. Considering the last three studies, our approach differed by using 2D straight landmarks existing on the floor of the environment. In addition, the extraction of the characteristics is based on the Hough transform.

3. Proposed System

The system proposed in this study shows an adequate visual SLAM technique for flat and closed environments with pre-existing floor lines and is an evolution of the robot location study conducted by Santana et al. (2008).

The algorithm used in visual SLAM is based on the extended Kalman filter (EKF) to allow the robot to navigate in indoor environments using odometry and pre-existing floor lines as landmarks. The lines are identified using the Hough transform. The prediction phase of the EKF is done using the geometric model of the robot. The update phase uses the parameters of the lines detected by the Hough transform directly in the Kalman equations without any intermediate calculation stage. Figure 1 shows the scheme of the proposed system.

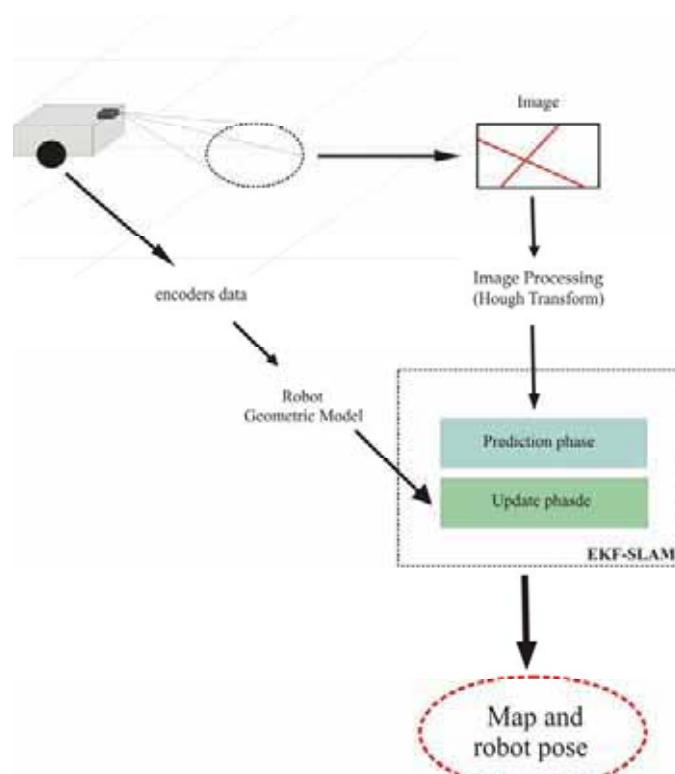


Fig. 1. Proposed System.

3.1 Theoretical Background

Extended Kalman Filter

In this work, the Extended Kalman Filter (EKF) deals with a system modeled by System (1), whose variables are described in Table (1). ε_t and δ_t are supposed to be zero-mean Gaussian white noises.

$$\begin{cases} s_t = p(s_{t-1}, u_{t-1}, \varepsilon_{t-1}) \\ z_t = h(s_t) + \delta_t \end{cases} \quad (1)$$

At each sampling time, the EKF calculates the best estimate of the state vector in two phases: a) the prediction phase uses System (2) to predict the current state based on the previous state and on the applied input signals; b) the update phase uses System (3) to correct the predicted state by verifying its compatibility with the actual sensor measurements.

$$\begin{cases} \bar{\mu}_t = p(\mu_{t-1}, u_{t-1}, 0) \\ \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t M_t V_t^T \end{cases} \quad (2)$$

$$\begin{cases} K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \\ \mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t)) \\ \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \end{cases} \quad (3)$$

where:

$$G_t = \frac{\partial p(s, u, \varepsilon)}{\partial s} \Big|_{s=\mu_{t-1}, u=u_{t-1}, \varepsilon=0} \quad (4)$$

$$V_t = \frac{\partial p(s, u, \varepsilon)}{\partial \varepsilon} \Big|_{s=\mu_{t-1}, u=u_{t-1}, \varepsilon=0} \quad (5)$$

$$H_t = \frac{\partial h(s)}{\partial s} \Big|_{s=\mu_{t-1}} \quad (6)$$

s_t	state vector (order n) at instant t
$p(\cdot)$	non-linear model of the system
u_{t-1}	input signals (order l), instant $t-1$
ε_{t-1}	process noise (order q), instant $t-1$
z_t	vector of measurements (order m) retourned by the sensors
$h(\cdot)$	non-linear model of the sensors
δ_t	measurement noise
μ_t	mean (order n) of the state vector s_t , before and after the update phase
Σ_t	covariance ($n \times n$) of the state vector s_t
G_t	Jacobian matrix ($n \times n$) that linearizes the system model $p(\cdot)$
V_t	Jacobian matrix ($n \times q$) that linearizes the process noise ε_t
M_t	covariance ($q \times q$) of the process noise ε_t
K_t	gain of the Kalman filter ($n \times m$)
H_t	Jacobian matrix ($m \times n$) that linearizes the model of the sensors $h(\cdot)$
Q_t	covariance matrix ($m \times m$) of the measurement noise δ_t

Table 1. Symbols in Equations (1), (2) and (3).

EKF-SLAM

In SLAM, besides estimating the robot pose, we also estimate the coordinates of all landmarks encountered along the way. This makes necessary to include the landmark coordinates into the state vector. If ${}^i c$ is the vector of coordinates of the i -th landmark and there are k landmarks, then the state vector is:

$$s_t = \begin{bmatrix} x_t, y_t, \theta_t, {}^1 c_t^T, {}^2 c_t^T, \dots, {}^k c_t^T \end{bmatrix}^T \quad (7)$$

When the number of marks (k) is a priori known, the dimension of the state vector is static; otherwise, it grows up when a new mark is found.

3.2 Modeling

Prediction phase: Process Model

Consider a robot with differential drive in which $\Delta\theta_R$ and $\Delta\theta_L$ are the right and left angular displacement of the respective wheels, according to Figure 2.

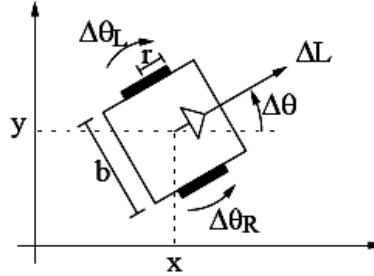


Fig. 2. Variables of the kinematic model.

Assuming that the speeds can be considered constant during one sampling period, we can determine the kinematic geometric model of the robot's movement (System 3.8):

$$\begin{cases} x_t = x_{t-1} + \frac{\Delta L}{\Delta\theta} [\sin(\theta_{t-1} + \Delta\theta) - \sin(\theta_{t-1})] \\ y_t = y_{t-1} - \frac{\Delta L}{\Delta\theta} [\cos(\theta_{t-1} + \Delta\theta) - \cos(\theta_{t-1})] \\ \theta_t = \theta_{t-1} + \Delta\theta \end{cases} \quad (8)$$

in which:

$$\begin{cases} \Delta L = (\Delta\theta_R r_R + \Delta\theta_L r_L) / 2 \\ \Delta\theta = (\Delta\theta_R r_R + \Delta\theta_L r_L) / b \end{cases} \quad (9)$$

where ΔL and $\Delta\theta$ are the linear and angular displacement of the robot; b represents the distance between wheels and r_R and r_L are the radii of the right and the left wheels, respectively. When $\Delta\theta \rightarrow 0$, the model becomes the one in Equation (10), obtained from the limit of System (8).

$$\begin{cases} x_t = x_{t-1} + \Delta L \cos(\theta_{t-1}) \\ y_t = y_{t-1} + \Delta L \sin(\theta_{t-1}) \\ \theta_t = \theta_{t-1} \end{cases} \quad (10)$$

Adopting the approach advocated by Thrun et al. (2005), we consider odometric information as input signals to be incorporated to the robot's model, rather than as sensorial measurements. The differences between the actual angular displacements of the wheels ($\Delta\tilde{\theta}_R$ and $\Delta\tilde{\theta}_L$) and those ones measured by the encoders $\Delta\tilde{\theta}_R$ and $\Delta\tilde{\theta}_L$ are modeled by a zero mean Gaussian white noise, accordingly to System (11).

$$\begin{cases} \Delta\theta_R = \Delta\tilde{\theta}_R + \varepsilon_R \\ \Delta\theta_L = \Delta\tilde{\theta}_L + \varepsilon_L \end{cases} \quad (11)$$

The measured $\Delta\tilde{L}$ and $\Delta\tilde{\theta}$ are defined by replacing $(\Delta\theta_R$ and $\Delta\theta_L)$ by $(\Delta\tilde{\theta}_R$ and $\Delta\tilde{\theta}_L)$ in Equations (9). If the state vector s is given by Equation (7), the system model $p(\cdot)$ can be obtained from Systems (8) or (10) and from the fact that the landmarks coordinates ${}^i c$ are static:

$$p(\cdot) = \begin{cases} x_t = \dots \\ y_t = \dots \\ \theta_t = \dots \\ {}^1 c_t = {}^1 c_{t-1} \\ \vdots \\ {}^k c_t = {}^k c_{t-1} \end{cases} \quad n = 3 + k \cdot \text{order}({}^i c) \quad (12)$$

$$u_t = [\Delta\tilde{\theta}_R \Delta\tilde{\theta}_L]^T \quad l = 2 \quad (13)$$

$$\varepsilon_t = [\varepsilon_R \varepsilon_L]^T \quad q = 2 \quad (14)$$

The \mathbf{G} and \mathbf{V} matrices are obtained by deriving the $p(\cdot)$ model using Equation (4) and Equation (5) respectively:

$$G = \begin{pmatrix} 1 & 0 & g_{13} & 0 & \dots & 0 \\ 0 & 1 & g_{23} & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix} \quad (15)$$

$$V = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \\ r_R / b & -r_L / b \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix} \quad (16)$$

where,

$$g_{13} = \frac{\Delta\tilde{L}}{\Delta\tilde{\theta}} [\cos(\theta_{t-1} + \Delta\tilde{\theta}) - \cos(\theta_{t-1})] \quad g_{23} = \frac{\Delta\tilde{L}}{\Delta\tilde{\theta}} [\sin(\theta_{t-1} + \Delta\tilde{\theta}) - \sin(\theta_{t-1})]$$

and considering $r_R = r_L = r$:

$$\begin{aligned}
 v_{11} &= V_1 \cos(\beta) + V_2 [\sin(\beta) - \sin \theta_{t-1}] & v_{12} &= -V_1 \cos(\beta) + V_3 [\sin(\beta) - \sin \theta_{t-1}] \\
 v_{21} &= V_1 \sin(\beta) - V_2 [\cos(\beta) - \cos \theta_{t-1}] & v_{22} &= -V_1 \sin(\beta) - V_3 [\cos(\beta) - \cos \theta_{t-1}] \\
 \beta &= \theta_{t-1} + \frac{r(\Delta\tilde{\theta}_R - \Delta\tilde{\theta}_L)}{b} & V_1 &= \frac{r(\Delta\tilde{\theta}_R + \Delta\tilde{\theta}_L)}{2(\Delta\tilde{\theta}_R - \Delta\tilde{\theta}_L)} \\
 V_2 &= \frac{-b\Delta\tilde{\theta}_L}{(\Delta\tilde{\theta}_R - \Delta\tilde{\theta}_L)^2} & V_3 &= \frac{b\Delta\tilde{\theta}_R}{(\Delta\tilde{\theta}_R - \Delta\tilde{\theta}_L)^2}
 \end{aligned}$$

It is well known that odometry introduces accumulative error. Therefore, the standard deviation of the noises ε_R and ε_L is assumed to be proportional to the module of the measured angular displacement. These considerations lead to a definition of the matrix \mathbf{M} given by Equation (17).

$$\mathbf{M} = \begin{pmatrix} (M_R |\Delta\tilde{\theta}_R|)^2 & 0 \\ 0 & (M_L |\Delta\tilde{\theta}_L|)^2 \end{pmatrix} \quad (17)$$

Update phase: Sensor Model

The landmarks adopted in this work are lines formed by the grooves of the floor in the environment where the robot navigates. The system is based on a robot with differential drive and a fixed camera, as in Figure 3.

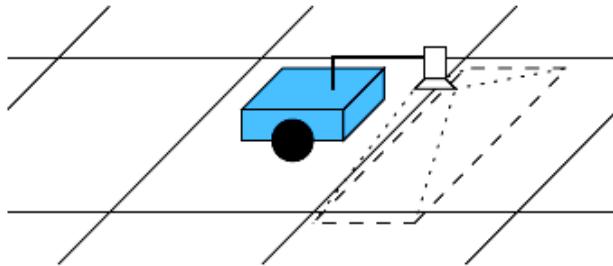


Fig. 3. Robotic system.

The landmarks are detected by processing images using the Hough transform. The detected lines are described by the parameters ρ and α in Equation (18).

$$\rho = x \cos(\alpha) + y \sin(\alpha) \quad (18)$$

Figure 4 shows the geometric representation of these parameters: ρ is the module and α is the angle of the shortest vector connecting the origin of the system of coordinates to the line.

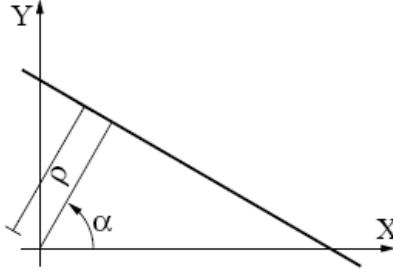


Fig. 4. Line parameters ρ and α .

We define a fixed coordinate system (F) and a mobile one (M), attached to the robot, both illustrated in Figure 5. The origin of the mobile system has coordinates (x_M^F, y_M^F) in the fixed system. θ_M^F represents the rotation of the mobile system with respect to the fixed one.

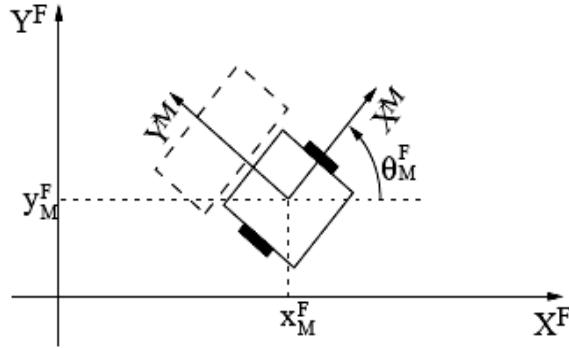


Fig. 5. Mobile and fixed coordinate systems.

One should note that there is a straight relation among these variables $(x_M^F, y_M^F, \theta_M^F)$ and the robot's pose (x_t, y_t, θ_t) , which is given by Equations (19).

$$x_t = x_M^F \quad y_t = y_M^F \quad \theta_t = \theta_M^F + \pi / 2 \quad (19)$$

Each line on the floor is described by two static parameters $(\tilde{\rho}, \tilde{\alpha})$. The map to be produced by the SLAM process is composed of a set of these pairs of parameters. So, the ${}_i c$ vector of coordinates of the i -th landmark that appears in Equations (7) and (12) is given by Equation (20):

$${}_i c = \begin{bmatrix} {}_i \rho^F \\ {}_i \alpha^F \end{bmatrix} \quad (20)$$

At each step the robot captures an image and identifies the parameters $(\tilde{\rho}, \tilde{\alpha})$ of the detected lines. These image parameters are then converted to the corresponding parameters $(\tilde{\rho}^M, \tilde{\alpha}^M)$ in the mobile coordinate system M attached to the robot, using the camera parameters. The vector of measurements \mathbf{z}_t to be used in the update phase of the EKF algorithm (Equation 3) is defined by Equation (21):

$$\mathbf{z}_t = \begin{bmatrix} \tilde{\rho}^M \\ \tilde{\alpha}^M \end{bmatrix} \quad (21)$$

To use information directly obtained by image processing $(\tilde{\rho}^M, \tilde{\alpha}^M)$ in the update phase of the EKF-SLAM, one must deduct the sensor model $h(\cdot)$, that is, the expected value of these parameters in function of the state variables.

We use the relation between coordinates in the M and F systems (Sistem 22) and Equation (18) in both coordinate systems (Equations 23 and 24):

$$\begin{cases} x^F = \cos(\theta_F^M)x^M - \sin(\theta_F^M)y^M + x_M^F \\ y^F = \sin(\theta_F^M)x^M + \cos(\theta_F^M)y^M + y_M^F \end{cases} \quad (22)$$

$${}^i\rho^F = x^F \cos({}^i\alpha^F) + y^F \sin({}^i\alpha^F) \quad (23)$$

$$\rho^M = x^M \cos(\alpha^M) + y^M \sin(\alpha^M) \quad (24)$$

By replacing Equations (22) in Equation (23), doing the necessary equivalences with System (24) and replacing some variables using Equations (19), we obtain the Systems (25) and (26), which represent two possible sensor models $h(\cdot)$ to be used in the filter. To decide about which model to use, we calculate both values of α^M and use the model which generate the value closer to the measured value $\tilde{\alpha}^M$.

$$\begin{cases} \rho^M = {}^i\rho^F - x_t \cos({}^i\alpha^F) - y_t \sin({}^i\alpha^F) \\ \alpha^M = {}^i\alpha^F - \theta_t + \pi/2 \end{cases} \quad (25)$$

$$\begin{cases} \rho^M = -{}^i\rho^F + x_t \cos({}^i\alpha^F) + y_t \sin({}^i\alpha^F) \\ \alpha^M = {}^i\alpha^F - \theta_t - \pi/2 \end{cases} \quad (26)$$

The sensor model is incorporated into the EKF through the \mathbf{H} matrix (Equation 6), given by Equation (27):

$$\mathbf{H} = \begin{pmatrix} -\cos({}^i\alpha^F) & -\sin({}^i\alpha^F) & 0 & \dots & 1 & 0 & \dots \\ 0 & 0 & -1 & \dots & 0 & 1 & \dots \end{pmatrix} \quad (27)$$

The final columns of the \mathbf{H} matrix are almost all null, except for the columns corresponding to the landmark in the vector state that matchs the detected line on the image.

3.3 Matching

A crucial aspect of the SLAM algorithm is to establish a match between the detected line on the image and one of the landmarks represented in the state vector. To choose the correct landmark, we first calculate the predicted values of (ρ^F, α^F) using the measured values of $(\tilde{\rho}^M, \tilde{\alpha}^M)$ and the model in Equations (25), if $\tilde{\alpha}^M \geq 0$, or in Equations (26), if $\tilde{\alpha}^M < 0$. Then,

these predicted values are compared to each one of the values (${}^i\rho^F$, ${}^i\alpha^F$) in the state vector. If the difference between the predicted values and the best (${}^i\rho^F$, ${}^i\alpha^F$) is small enough, a match was found. If not, we consider that a new mark was detected and the size of the state vector is increased.

3.4 Image processing

Detection of lines

Due to the choice of floor lines as landmarks, the technique adopted to identify them was the Hough transform (Gonzalez & Woods, 2007). The purpose of this technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a accumulator grid that is constructed by the algorithm for computing the Hough transform. In our case, the shapes are lines described by Equation (18) and the parameter space has coordinates (ρ , α). The images are captured in grayscale and converted to black and white using a threshold level, determined off-line. Figure 6 shows a typical image of the floor and the lines detected by the Hough transform.

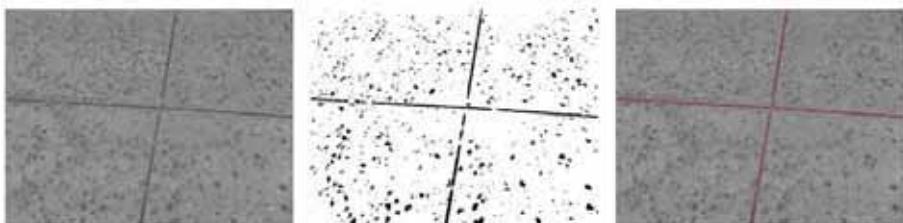


Fig. 6. Detection of lines.

From images to the world

We assume that the floor is flat and that the camera is fixed. So, there is a constant relation (a homography A) between points in the floor plane (x , y) and points in the image plane (u , v):

$$s \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = A \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (28)$$

The scale factor s is determined for each point in such a way that the value of the third element of the vector is always 1. The homography can be calculated off-line by using a pattern containing 4 or more remarkable points with known coordinates (see Figure 7). After detecting the remarkable point in the image, we have several correspondences between point coordinates in the mobile coordinate system M and in the image. Replacing these points in Equation (28), we obtain a linear system with which we can determine the 8 elements of the homography matrix A .



Fig. 7. Calibration pattern.

Once calculated the homography, for each detected line we do the following:

1. using the values of (ρ, α) in the image obtained by the Hough transform, calculate two point belonging to the image line;
2. convert the coordinates of these two points to the mobile coordinate system M ;
3. determine $(\tilde{\rho}^M, \tilde{\alpha}^M)$ of the line that passes through these two points.

4. Results

The experiments were carried out using the Karel robot, a reconfigurable mobile platform built in our laboratory (see Figure 7). The robot has two wheels that are driven by DC motors with differential steering. Each motor has an optical encoder and a dedicated card based on a PIC microcontroller that controls local velocity. The cards communicate with the computer through a CAN bus, receiving the desired wheel velocities and encoder data.

The computer that controls the robot is a notebook with a USB-to-CAN bridge and a color webcam connected to its USB bus. The camera captures 160×120 images (Figure 6) and each image is processed at 300 ms.

The experiment was conducted in an environment with pre-existing floor lines and proceeded as follows: the robot executes a forward movement, turns around and tries to return to the departure point.

Figure 8 shows the position of the lines, the approximate trajectory of the robot, the comparative size of the camera's field of view and the "name" of each straight line. The initial position of the robot is approximately $(3, 3.1, 110^\circ)$, but it is not used in the SLAM algorithm: the robot assumes that its initial position is $(0, 0, 0^\circ)$.

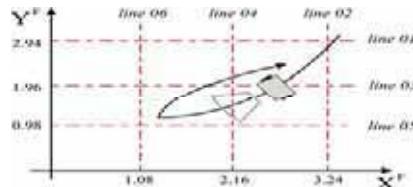


Fig. 8. Description of the experiment performed.

The name of the lines is related to the temporal order in which the robot identified them. Figure 9 shows the moment of identification of each straight line. Initially, the robot identifies two lines (Figure 9-a); advancing it identifies the third (Figure 9-b) and a little further on identifies the fourth (Figure 9-c). The fifth line is only seen when the robot has already negotiated around half of the trajectory (Figure 9-d) and the sixth is only identified when the robot begins to rotate on its own axis (Figure 9-e)..

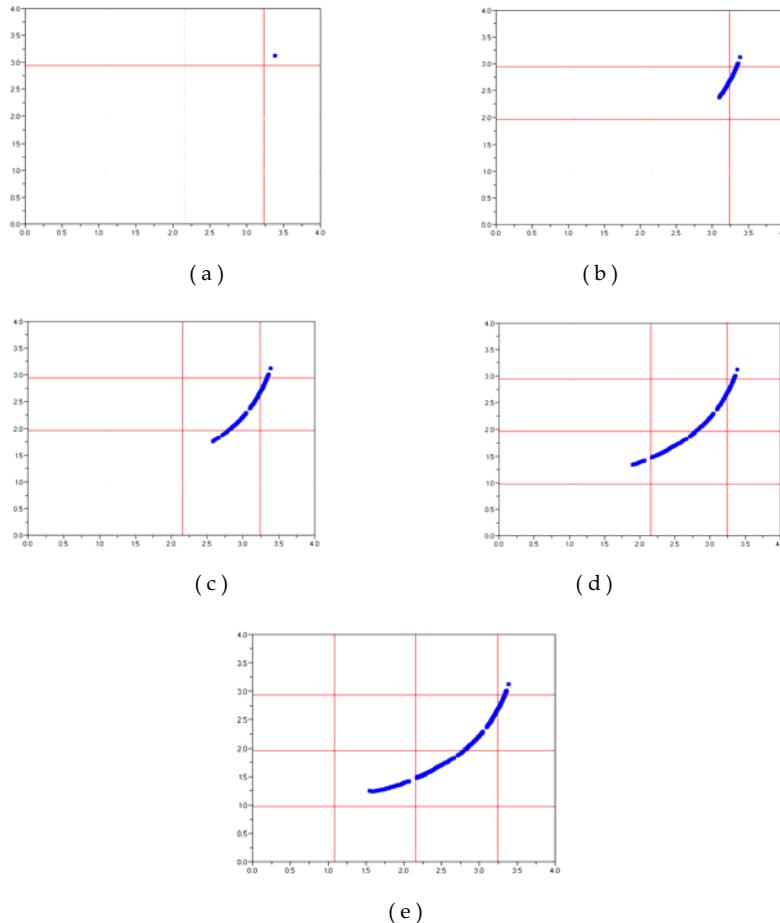


Fig. 9. Temporal process of identifying the floor lines.

Considering the total movement of the robot, we found that it identifies lines 56% of the time (one line 65% of the time and two lines 35% of the time), as shown in Figure 10. This figure shows that at a particular instant the robot identified three lines. This incorrect information originated in an image where the shadow of an obstacle was identified as a line.

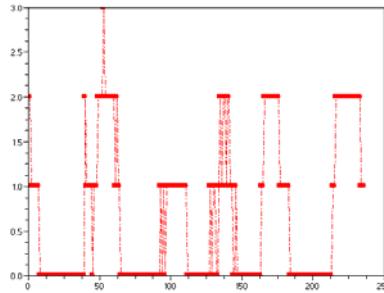


Fig. 10. Temporal process of identifying floor lines.

The process of matching all the experience was successful in 93% of the cases, as illustrated in Table 2. Figure 11 shows the straight lines identified during the robot's trajectory.

Lines	Identifications	Correct	Incorrect	Hit rate
Line 01	2	2	0	100%
Line 02	29	28	1	96,50%
Line 03	44	41	3	93,20%
Line 04	34	32	2	94,10%
Line 05	23	22	1	95,60%
Line 06	13	11	2	84,65%

Table 2. Matching.

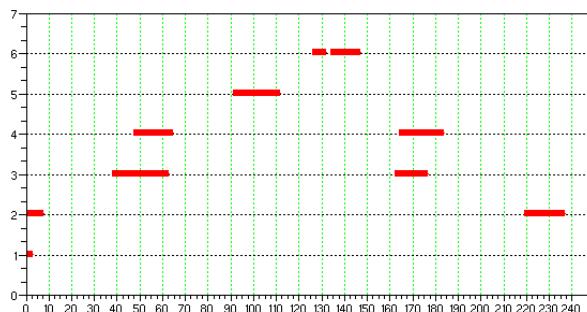


Fig. 11. Straight lines identified during the robot's trajectory.

Based on line 3, the line most seen during the experiment, Figure 12 shows the behavior of the p and α variance for this line. It can be seen that the behavior is in accordance with the theory and is always minimized when it and other lines are observed.

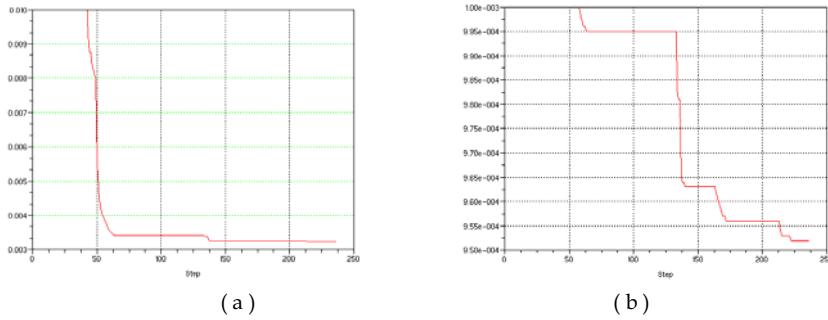
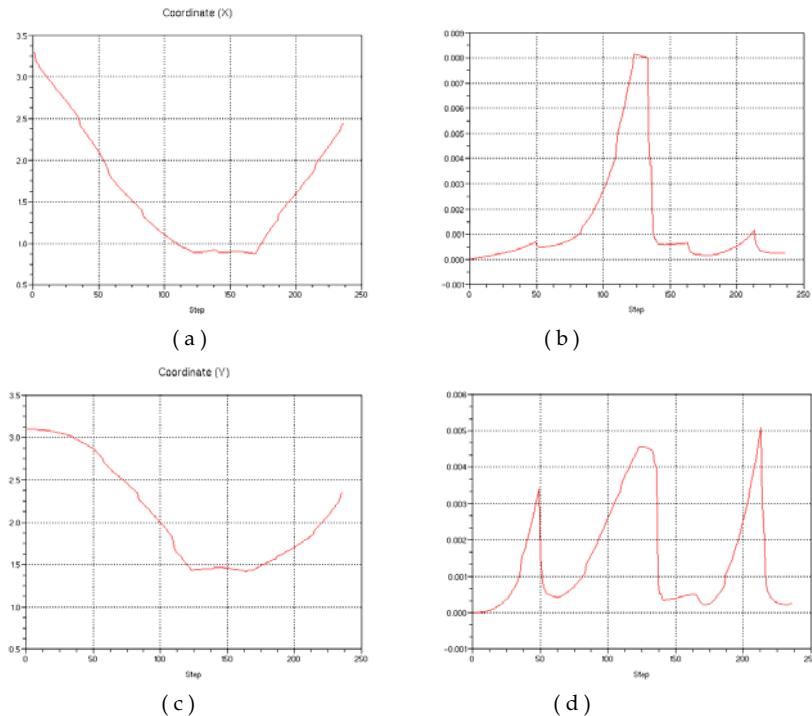


Fig. 12. Variance behavior: (a) p of line 3 and (b) a of line 3.

The coordinates x , y and θ and their respective variances are shown in Figure 13..



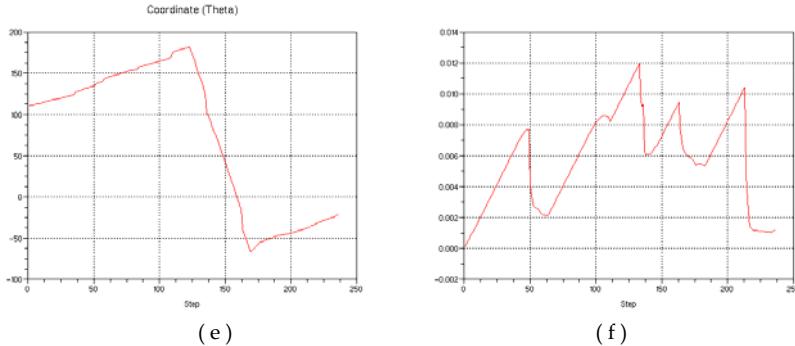


Fig. 13. (a) coordinate X, (b) variance in X, (c) coordinate Y, (d) variance in Y, (e) coordinate θ, (f) variance in θ.

Table 2 shows the values of the landmarks detected after around 250 iterations. The values are expressed in the system of coordinates, assuming that the initial position of the robot is (0, 0, 0°).

Actual		Calculated		Corrected	
ρ	α	ρ	α	ρ	α
1.08	0	1.88	108	1.13	1.7
2.16	0	0.82	108	2.19	1.3
3.24	0	0.11	110	3.26	0
0.98	90	2.57	23	0.97	87
1.96	90	1.24	22	1.95	87
2.94	90	0.20	22	2.93	88

Table 2. Feature map.

5. Conclusions and Perspectives

This chapter presents a new perspective for solving SLAM problems using optical sensors, given that they are increasingly cheaper and provide a wealth of information on the environment.

The main contribution of this work is an optical sensor modeling that enables the use of parameters obtained by the image processing algorithm directly in the Kalman filter equations without intermediate phases to calculate position or distance.

The values presented in Table 2 demonstrate that the proposed approach can obtain good results even with a relatively small number of samples.

As future works, we intend to improve the real-time properties of the image processing algorithm, by adopting some of the less timeconsuming variants of the Hough transform. Another required improvement is to deal with line segments with finite length, incorporating this characteristics to the step of matching lines.

6. References

- Ahn, S.; Chung, W. K. & Oh, S. (2007), Construction of Hybrid Visual Map for Indoor SLAM, *IEEE/RSJ International Conference on Intelligent Robots and Systems* .
- Angeli, A.; Filliat, D.; Doncieux, S. & Meyer, J. (2008), Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words, *IEEE Transaction on Robotics*, Vol. 24, No. 5, pp. 1027-1037.
- Chen, Z. (2006), A Visual SLAM Solution Based on High Level Geometry Knowledge and Kalman Filtering, *IEEE CCECE/CCGEI*, pp. 1283-1286 .
- Choi, Y. & Oh, S. (2006), Grid-based Visual SLAM in Complex Environment, *IEEE/RSJ International Conference on Intelligent Robots and Systems* .
- Clemente, L.; Davison, A.; Reid, I.; Neira, J. & Tardos, J. (2007), Mapping Large Loops with a Single Hand-Held Camera', *Proceedings of Robotics: Science and Systems* .
- Dailey, M. N. & Parnichkun, M. (2006), Simultaneous Localization and Mapping with Stereo Vision', *International Conference on Control, Automation, Robotics and Vision*.
- Davison, A. J. & Murray, D. W. (2002), Simultaneous Localization and Map-Building Using Active Vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 865-880.
- Davison, A. J.; Cid, Y. G. & Kita, N. (2004), Real-time 3d SLAM with Wide-Angle Vision, *15th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*.
- Durrant-Whyte, H. & Bailey, T. (2006). Simultaneous Localization and Mapping: Part I, *IEEE Robotics & Automation Magazine*, pp. 99-108.
- Esteban, I. (2008), Visual Trajectory Based SLAM, *Master Thesis*, Universiteite van Amsterdam.
- Estrada, C.; Neira, J. & Tards, J. D. (2005), Hierarchical SLAM: Realtime Accurate Mapping of Large Environments, *IEEE Transactions on Robotics*, pp. 588-596.
- Frintrop, S.; Jensfelt, P. & Christensen, H. I. (2006), Attentional Landmark Selection for Visual SLAM, *IEEE/RSJ Int. Conference on Intelligent Robots and Systems* .
- Fu, S.; Liu, C.; Gao, L. & Gai, Y. (2007), SLAM for Mobile Robots Using Laser Range Finder and Monocular Vision, *IEEE Int. Conference on Robotics and Automation* .
- Gonzalez, R. C. & Woods, R. E. (2007). *Digital Image Processing*, Prentice Hall, ISBN: 13168728X, USA.
- Herath, D. C.; Kodagoda, K. R. S. & Dissanayake, G. (2007), Stereo Vision Based SLAM Issues and Solutions, *IEEE International Conference on Robotics and Automation* .
- Jing Wu, J. & Zhang, H. (2007), 'Camera Sensor Model for Visual SLAM, *Conference on Computer and Robot Vision* .
- Jung, I. K. (2004), Simultaneous Localization and Mapping in 3D Environments with Stereovision, *PhD Thesis*, Institut National Polytechnique - Toulouse - France.
- Kwok, N. M.; Dissanayake, G. & Ha, Q. P. (2005), Bearing-only SLAM Using a Sport Based Gaussian Sum Filter, *IEEE International Conference on Robotics and Automation* .
- Lemaire, T. & Lacroix, S. (2007), Monocular SLAM as a Graph of Coalesced Observations, *IEEE Int. Conference on Robotics and Automation*, pp.2791-2796 .
- Lee, Y. & Song, J. (2007), Autonomous selection, Registration, and Recognition of Objects for Visual SLAM in Indoor Environments', *International Conference on Control, Automation and Systems* .

- Leonard, J. J. & Durrant-Whyte, H. F. (1991), Mobile Robot Localization by Tracking Geometric Beacons, *IEEE Transactions on Robotics and Automation*, Vol. 1, No. 3, pp. 376-382 .
- Mansinghka, V. K. (2004), Towards Visual SLAM in Dynamic Environments, *Springer* .
- Steder, B.; Grisetti, G.; Stachniss, C. & Burgard, W. (2008), Visual SLAM for Flying Vehicles, *Academic Publish* .
- Santana, A. M.; de Souza, A. A. S.; Britto, R. S.; Alsina, P. J.; Medeiros, A. A. D. Localization of a Mobile Robot Based on Odometry and Natural Landmarks Using Extended Kalman Filter. *International Conference on Informatics in Control, Automation and Robotics*. Funchal, Madeira, Portugal, 08/2008.
- Smith, R. & Cheesman, P. (1987), On the Representation of Spatial Uncertainty, *The International Journal of Robotics Research* , Vol. 5, No. 4, pp. 56-68 .
- Thrun, S.; Burgard, W. & Fox, D. (2005). *Probabilistic Robotics*, MIT Press, ISBN: 9780262201629, USA.