Fusion of Odometry and Visual Datas to Localization a Mobile Robot

André M. Santana[†], Anderson A. S. Souza[‡], Luiz M. G. Gonçalves[‡],
Pablo J. Alsina[‡], Adelardo A. D. Medeiros[‡]

[†] *Federal University of Piauí – UFPI*
Department of Informatics and Statistic – 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

Applications involving wheeled mobile robots have been growing significantly in recent years thanks to its ability to move freely through space work, limited only by obstacles. Moreover, the wheels allow for greater convenience of transportation in environments plans and give greater support to the static robot.

In the context of autonomous navigation of robots we highlight the localization problem. From an accumulated knowledge about the environment and using the current readings of the sensors, the robot must be able to determine and keep up its position and orientation in relation to this environment, even if the sensors have errors and / or noise. In other words, to localize a robot is necessary to determine its pose (position and orientation) in the workspace at a given time.

Borenstein et al. (1997) have classified the localization methods in two great categories: relative localization methods, which give the robot's pose relative to the initial one, and absolute localization methods, which indicate the global pose of the robot and do not need previously calculated poses.

As what concerns wheel robots, it is common the use of encoders linked to wheel rotation axes, a technique which is known as odometry. However, the basic idea of odometry is the integration of the mobile information in a determined period of time, what leads to the accumulation of errors (Park et al., 1998). The techniques of absolute localization use landmarks to locate the robot. These landmarks can be artificial ones, when introduced in the environment aiming at assisting at the localization of the robot, or natural ones, when they can be found in the proper environment.

It´s important to note that, even the absolute location techniques are inaccurate due to noise from the sensors used. Aiming to obtain the pose of the robot with the smallest error possible an efficient solution is to filter the information originated by its sensors. A mathematical tool to accomplish this task is the Kalman filter.

Still on autonomous robots, a key attribute is a reliable perception of the world. Besides the 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.

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

The algorithm used is based on the Extended Kalman Filter (EKF), to allow the robot to navigate in an indoor environment using odometry and preexisting floor. 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.

The use of lines is justified as follows: a) lines can be easily detected in images; b) floor lines are generally equally well spaced, reducing the possibility of confusion; c) 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 d) after processing the number of pixels in the image that belong to the line is a good reliability measure of the landmark detected.

Literature shows works using distance measures to natural landmarks to locate the robot. Bezerra (2004) used in his work the lines of the floor composing the environment as natural landmarks. Kiriy and Buehler (2002) have used extended Kalman Filter to follow a number of artificial landmarks placed in a non-structured way. Launay et al. (2002) employed ceiling lamps of a corridor to locate the robot. Odakura et al. (2004) show the location of the robot using Kalman filter with partial observations. More recent studies show a tendency to solve the problem of simultaneous localization and mapping - SLAM. Examples of work in this area: Amarasinghe et al. (2009), Marzorati et al. (2009) and Wu et al. (2009).

2. Proposed System and Theoretical Background

The system proposed in this study presents an adequate technique to localization a mobile robot in flat and closed environments with pre-existing floor lines. The algorithm used is based on Extended Kalman Filter (EKF) to allow the robot to navigate in an indoor environment by fusing odometry information and image processing. The prediction phase of the EKF is done using the odometric model of the robot and the update phase uses the parameters of the lines detected by Hough directly in the Kalman equations without any intermediate calculation stage. Figure 1 shows the scheme of the proposed system.

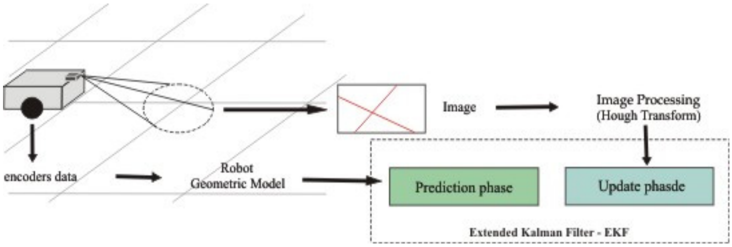

Fig. 1. Proposed System.

2.1 Kalman Filter

In 1960, Rudolph Emil Kalman published a famous paper describing a recursive process for solving problems related to linear discrete filtering (Kalman 1960). His research has provided significant contributions by helping to establish solid theoretical foundation in many areas of the engineering systems.

With the advance computing, the Kalman filter and its extensions to nonlinear problems represent a product widely used in a modern engineering. Next will be described in summary form, the Kalman filter applied to linear and nonlinear systems.

2.2 Discrete Kalman Filter - DKF

Aiube et al. (2006) define the Kalman filter as a set of mathematical equations that provides an efficient recursive process of estimation, since the square error of estimation is minimized. Through the observation of the variable named " observation variable" another variable, not observable, the "state variable" can be estimated efficiently The modeling of the Discrete Kalman Filter-DKF presupposes that the system is linear and described by the model of the equations of the System (1):

$$\begin{cases} \boldsymbol{s}_t = \mathbf{A}_t \boldsymbol{s}_{t-1} + \mathbf{B}_t \boldsymbol{u}_{t-1} + \gamma_{t-1} \\ \boldsymbol{z}_t = \mathbf{C}_t s_t + \delta_t \end{cases} \tag{1}$$

in which **s** ϵ $\boldsymbol{R}^n$ is the state vector; **u** ϵ $\mathbf{R}^l$ is the vector of input signals; **z** ϵ $\mathbf{R}^m$ is the vector of measurements; the matrix $n \ x \ n$, **A**, is the transition matrix of the states; **B**, $n \ x \ l$, is the coefficient matrix on entry; matrix **C**, $m \ x \ n$, is the observation matrix; γ ϵ $\mathbf{R}^n$ represents the vector of the noises to the process and δ ϵ $\mathbf{R}^m$ the vector of measurement errors. Indexes *t* and *t-1* represent the present and the previous instants of time.

The Filter operates in prediction-actualization mode, taking into account the statistical proprieties of noise. An internal model of the system is used to updating, while a retro-alimentation scheme accomplishes the measurements. The phases of prediction and actualization to DKF can be described by the Systems of Equations (2) and (3) respectively.

$$\begin{cases} \bar{\mu}_t = \mathbf{A}_t \, \mu_{t-1} + \mathbf{B}_t \boldsymbol{u}_{t-1} \\ \bar{\Sigma}_t = \mathbf{A}_t \Sigma_{t-1} \mathbf{A}_t^T + \mathbf{R}_t \end{cases} \tag{2}$$

$$\begin{cases} \mathbf{K}_t = \bar{\Sigma}_t \mathbf{C}_t^T (\mathbf{C}_t \bar{\Sigma}_t \mathbf{C}_t^T + \mathbf{Q}_t)^{-1} \\ \mu_t = \bar{\mu}_t + \mathbf{K}_t (\boldsymbol{z}_t - \mathbf{C}_t \bar{\mu}_t) \\ \Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \bar{\Sigma}_t \end{cases} \tag{3}$$

The Kalman Filter represents the state vector $\mathbf{s}_t$ by its mean μ_t and co-variance Σ_t. Matrixes **R**, *n x n*, and **Q**, *l x l*, are the matrixes of the covariance of the noises of the process (γ) and measurement (δ) respectively, and matrix **K**, *n x m*, represents the gain of the system.

2.3 Extended Kalman Filter - EKF

The idea of the EKF is to linearize the functions around the current estimation using the partial derivatives of the process and of the measuring functions to calculate the estimations, even in the face of nonlinear relations. The model of the system to EKF is given by the System (4):

$$\begin{cases} \boldsymbol{s}_t = g(\boldsymbol{s}_{t-1}, \boldsymbol{u}_{t-1}) + \gamma_t \\ \boldsymbol{z}_t = h(\boldsymbol{s}_t) + \delta_t \end{cases} \tag{4}$$

in which $g(\mathbf{u}_{t-1}; \mathbf{s}_{t-1})$ is a non-linear function representing the model of the system, and $h(\mathbf{s}_t)$ is a nonlinear function representing the model of the measurements. Their prediction and actualization phases can be obtained by the Systems of Equations (5) and (6) respectively.

$$\begin{cases} \bar{\mu}_t = g(\mu_{t-1}, \boldsymbol{u}_{t-1}) \\ \bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{R}_t \end{cases} \tag{5}$$

$$\begin{cases} \mathbf{K}_t = \bar{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1} \\ \mu_t = \bar{\mu}_t + \mathbf{K}_t \big(\boldsymbol{z}_t - h(\bar{\mu}_t)\big) \\ \Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t \end{cases} \tag{6}$$

The matrix **G**, *n x n*, is the jacobian term linearizes the model and **H**, *l x n*, is the jacobian term linearizes the measuring vector. Such matrixes are defined by the Equations (7) e (8).

$$\mathbf{G}_t = \frac{\partial g(\mu_{t-1}, \boldsymbol{u}_{t-1})}{\partial \mathbf{s}_{t-1}} \tag{7}$$

$$\mathbf{H}_t = \frac{\partial h(\mathbf{s}_{t-1})}{\partial \mathbf{s}_t} \tag{8}$$

3. Modeling

3.1 Prediction phase: process model

Traditionally, the behavior of the robot motion is described by its dynamic model. Modeling this type of system is quite complex because there are many variables involved (masses and moments of inertia, friction, actuators, etc.). Even in the most elaborate systems cannot faithfully portray the behavior of the robot motion.

A classic method used to calculate the pose of a robot is the odometry. This method uses sensors, optical encoders, for example, which measure the rotation of the robot's wheels.

Using the cinematic model of the robot, its pose is calculated by means of the integration of its movements from a referential axis.

As encoders are sensors, normally their reading would be implemented in the actualization phase of the Kalman Filter, not in the prediction phase. Thrun et al. (2005) propose that odometer information does not function as sensorial measurements; rather they suggest incorporating them to the robot's model. In order that this proposal is implemented, one must use a robot's cinematic model considering the angular displacements of the wheels as signal that the system is entering in the prediction phase of the Kalman Filter.

Consider a robot with differential drive in which the control signals applied and its actuators are not tension, instead angular displacement, according to Figure 2.

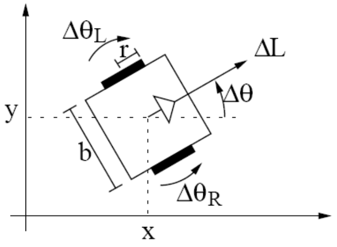

Fig. 2. Variables of the kinematic model.

With this idea, and supposing that speeds are constant in the sampling period, one can determine the geometric model of the robot's movement (System 9).

$$\begin{cases} x_t = x_{t-1} + \frac{\Delta L}{\Delta\theta}[\sin(\theta_{t-1} + \Delta\theta) - \sin(\theta_{t-1})] \\ y_t = y_{t-1} + \frac{\Delta L}{\Delta\theta}[\cos(\theta_{t-1} + \Delta\theta) - cos(\theta_{t-1})] \\ \theta_t = \theta_{t-1} + \Delta\theta \end{cases} \tag{9}$$

The turn easier the readability of the System (9) representing the odometry model of the robot, two auxiliary variables have been employed ΔL and Δθ.

$$\begin{cases} \Delta L = (\Delta\theta_R r_R + \Delta\theta_L r_L)/2 \\ \Delta\theta = (\Delta\theta_R r_R - \Delta\theta_L r_L)/b \end{cases} \tag{10}$$

in which $\Delta\theta_R$ is the reading of the right encoder and functions relatively the robot by means of the angular displacement of the right wheel; $\Delta\theta_L$ is the reading of the left encoder and functions as a displacement applied to the left wheel; *b* represents the distance from wheel to wheel of the robot; r_L and r_R are the spokes of the right and the left wheels respectively.

It is important to emphasize that in real applications the angular displacement effectively realized by the right wheel differs of that measured by the encoder. Besides that, the supposition that the speeds are constant in the sampling period, which has been used to obtain the model 9, is not always true. Hence, there are differences between the angular displacements of the wheels ($\Delta\hat{\theta}_R$ and $\Delta\hat{\theta}_L$) and those ones measured by the encoders ($\Delta\theta_R$ and $\Delta\theta_L$). This difference will be modeled by a Gaussian noise, according to System (11).

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

It is known that odometry possesses accumulative error. Therefore, the noises ε_R and ε_L do not possess constant variance. It is presumed that these noises present a proportional standard deviation to the module of the measured displacement. With these new considerations, System (9) is now represented by System (12):

$$\begin{cases} x_t = x_{t-1} + \frac{\Delta\hat{L}}{\Delta\hat{\theta}}\left[\sin\left(\theta_{t-1} + \Delta\hat{\theta}\right) - \sin(\theta_{t-1})\right] \\ y_t = y_{t-1} + \frac{\Delta\hat{L}}{\Delta\hat{\theta}}\left[\cos\left(\theta_{t-1} + \Delta\hat{\theta}\right) - cos(\theta_{t-1})\right] \\ \theta_t = \theta_{t-1} + \Delta\hat{\theta} \end{cases} \quad (12)$$

in which:

$$\begin{cases} \Delta\hat{L} = (\Delta\hat{\theta}_R r_R + \Delta\hat{\theta}_L r_L)/2 \\ \Delta\hat{\theta} = (\Delta\hat{\theta}_R r_R - \Delta\hat{\theta}_L r_L)/b \end{cases} \quad (13)$$

One should observe that this model cannot be used when $\Delta\hat{\theta} = 0$. When it occurs, one uses an odometry module simpler than a robot (System 14), obtained from the limit of System (12) when $\Delta\hat{\theta} \rightarrow 0$.

$$\begin{cases} x_t = x_{t-1} + \Delta\hat{\theta}\cos(\theta_{t-1}) \\ y_t = y_{t-1} + \Delta\hat{\theta}\sin(\theta_{t-1}) \\ \theta_t = \theta_{t-1} \end{cases} \quad (14)$$

Thrun's idea implies a difference as what concerns System (4), because the noise is not audible; rather, it is incorporated to the function which describes the model, as System (15) shows:

$$\begin{cases} \boldsymbol{s}_t = p(\boldsymbol{s}_{t-1}, \boldsymbol{u}_{t-1}, \varepsilon_{t-1}) \\ \boldsymbol{z}_t = h(\boldsymbol{s}_t) + \delta_t \end{cases} \quad (15)$$

in which $\varepsilon_t = [\varepsilon_R \quad \varepsilon_L]^T$ is the noise vector connected to odometry.

It is necessary, however, to bring about a change in the prediction phase of the System (6) resulting in the System (16) equations:

$$\begin{cases} \bar{\mu}_t = \mu_{t-1} + p(\mathbf{u}_{t-1}, \mu_{t-1}, 0) \\ \bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{V}_t \mathbf{M}_t \mathbf{V}_t^T \end{cases} \quad (16)$$

in which, **M**, $l \ x \ l$, is the co-variance matrix of the noise sensors (ε) and **V**, $n \ x \ m$, is the jacobian mapping the sensor noise to the space of state. Matrix **V** is defined by equation (17).

$$\mathbf{V}_t = \frac{\partial p(\mathbf{s}_{t-1}, \mathbf{u}_{t-1}, 0)}{\partial \mathbf{u}_{t-1}} \quad (17)$$

Making use of the odometry model of the robot described in this section and the definitions of the matrixes used by the Kalman Filter, we have:

$$\mathbf{G}_t = \begin{pmatrix} 1 & 0 & \frac{\Delta\hat{L}}{\Delta\hat{\theta}}\left[\cos\left(\theta_{t-1} + \Delta\hat{\theta}\right) - \cos(\theta_{t-1})\right] \\ 0 & 1 & \frac{\Delta\hat{L}}{\Delta\hat{\theta}}\left[\sin\left(\theta_{t-1} + \Delta\hat{\theta}\right) - \sin(\theta_{t-1})\right] \\ 0 & 0 & 1 \end{pmatrix} \tag{18}$$

$$\mathbf{V}_t = \begin{pmatrix} K_1\cos(K_2) - K_3[\sin(K_2) - \sin\,(\theta_{t-1})] & -K_1\cos(K_2) + K_3[\sin(K_2) - \sin\,(\theta_{t-1})] \\ K_1\sin(K_2) - K_3[-\cos(K_2) + \cos\,(\theta_{t-1})] & -K_1\sin(K_2) + K_3[-\cos(K_2) + \cos\,(\theta_{t-1})] \\ r_R/b & -r_L/b \end{pmatrix} \tag{19}$$

$$\mathbf{M}_t = \begin{pmatrix} \alpha_1|\Delta\hat{\theta}_R| & 0 \\ 0 & \alpha_1|\Delta\hat{\theta}_L| \end{pmatrix} \tag{20}$$

Elements m_{11} and m_{22} in the Equation (20) represent the fact that the standard deviations of ε_R and ε_L are proportional to the module of the angular displacement. The variables k_1, k_2 and k_3 are given by System (21), considering $rd = re = r$.

$$\begin{cases} K_1 = \frac{r(\Delta\hat{\theta}_R + \Delta\hat{\theta}_L)}{b(\Delta\hat{\theta}_R - \Delta\hat{\theta}_L)} \\ K_2 = \theta_{t-1} + \frac{r(\Delta\hat{\theta}_R - \Delta\hat{\theta}_L)}{b} \\ K_3 = \frac{b\Delta\hat{\theta}_L}{2[r(\Delta\hat{\theta}_R - \Delta\hat{\theta}_L)/b]^2} \end{cases} \tag{21}$$

3.2 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.

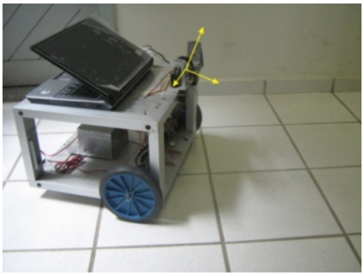

Fig. 3. Robotic System.

Due to the choice of the straight lines as landmarks, the technique adopted to identify them was the Hough transform [Hough, 1962]. This kind of transform is a method employed to identify inside a digital image a class of geometric forms which can be represented by a parametric curve [Gonzales, 2007]. As what concerns the straight lines, a mapping is provided between the Cartesian space (X, Y) and the space of the parameters (ρ, α) where the straight line is defined.

Hough defines a straight line using its common representation, as Equation (22) shows, in which parameter (ρ) represents the length of the vector and (α) the angle this vector forms with axis X. Figure 4 shows the geometric representation of these parameters.

$$\rho = x.\cos(\alpha) + y.\sin(\alpha) \quad (22)$$

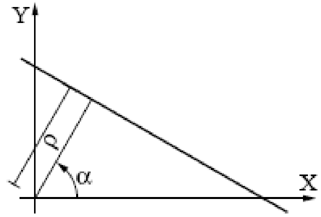

Fig. 4. Line parameters: ρ and α.

The robot navigates in an environment where the position of the lines in the world is known and every step identifies the descriptors of the lines contained in the image ρ^I e α^I. These descriptors are mapped to the plane of a moving coordinate system and obtaining ρ^M e α^M. This transformation is easy and relies only on the correct calibration of camera parameters. Figure 5 illustrates the coordinate systems used in mathematical deduction of the sensor model.

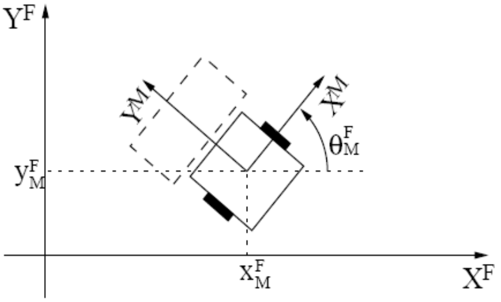

Fig. 5. Mobile (*M*) and Fixed (*F*) coordinate systems.

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. 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 (23).

$$x_t = x_M^F \qquad y_t = y_M^F \qquad \theta_t = \theta_M^F + \pi/2 \tag{23}$$

We use the relation between coordinates in the (*M)* and (*F*) systems (System 24) and Equation (22) in both coordinate systems (Equations 25 and 26).

$$\begin{cases} x^F = \cos(\theta_M^F)\, x^M - \sin(\theta_M^F)\, y^M + x_M^F \\ y^F = \sin(\theta_M^F)\, x^M + \cos(\theta_M^F)\, y^M + y_M^F \end{cases} \tag{24}$$

$$\rho^F = x^F \cos(\alpha^F) + y^F \sin(\alpha^F) \tag{25}$$

$$\rho^M = x^M \cos(\alpha^M) + y^M \sin(\alpha^M) \tag{26}$$

By replacing Equations (24) in Equation (25), doing the necessary equivalences with Equation (26) and replacing some variables using Equations (23), we obtain the Systems (27) and (28), which represent two possible sensor models $h(.)$ 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 generates the value closer to the measured value.

$$\begin{cases} \rho^M = \rho^F - x_t \cos(\alpha^F) - y_t \sin(\alpha^F) \\ \alpha^M = \alpha^F - \theta_t + \frac{\pi}{2} \end{cases} \tag{27}$$

$$\begin{cases} \rho^M = -\rho^F + x_t \cos(\alpha^F) + y_t \sin(\alpha^F) \\ \alpha^M = \alpha^F - \theta_t - \frac{\pi}{2} \end{cases} \tag{28}$$

The sensor model is incorporated into the EKF through the matrix H (Equation 8). Representation for **H** obtained from the System (27) is given by Equation (29) and, using the System (28), **H** is described by Equation (30).

$$\mathbf{H} = \begin{pmatrix} -\cos(\alpha^F) & -\sin(\alpha^F) & -x_t \sin(\alpha^F) + y_t \cos(\alpha^F) \\ 0 & 0 & -1 \end{pmatrix} \tag{29}$$

$$\mathbf{H} = \begin{pmatrix} \cos(\alpha^F) & \sin(\alpha^F) & x_t \sin(\alpha^F) - y_t \cos(\alpha^F) \\ 0 & 0 & -1 \end{pmatrix} \tag{30}$$

4. Image Processing

4.1 Detection of lines

Due to the choice of floor lines as landmarks, the technique adopted to identify them was the Hough transform [Hough, 1962]. 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 an accumulator grid that is constructed by the algorithm for computing the Hough transform [Bradski and Kaehler, 2008].

In our case, the shapes are lines described by Equation (22) and the parameter space has coordinates (ρ,α). The images are captured in grayscale and converted to black and white using the edge detector Canny [Canny, 1986]. Figure 6.a shows a typical image of the floor, Figure 6.b shows the image after applying the Canny detector and Figure 6.c shows lines detected by Hough.

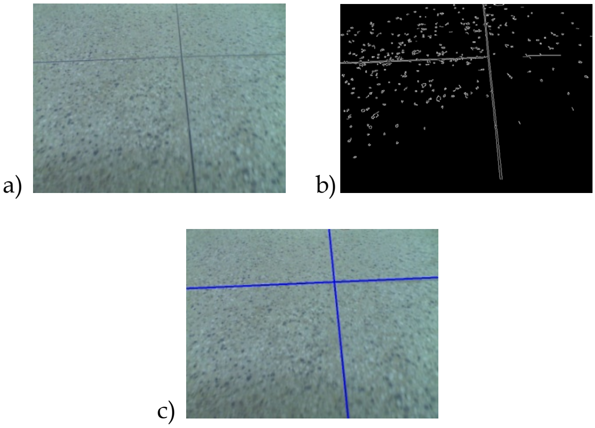

Fig. 6. Image processing.

4.2 From images to the word

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):

$$s.\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{A}\,.\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{31}$$

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.a). 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 (31), we obtain a linear system with which we can determine the 8 elements of the homography matrix **A**.

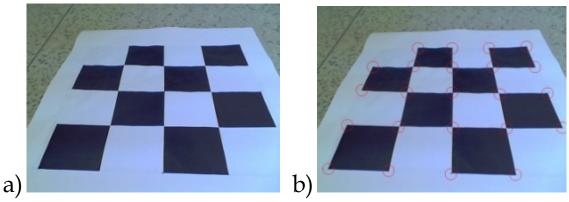

Fig. 7. Calibration pattern.

Once calculated the homography, for each detected line we do the following: a) using the values of $(\tilde{\rho}, \tilde{\alpha})$ in the image obtained by the Hough transform, calculate two point belonging to the image line; b) convert the coordinates of these two points to the mobile coordinate system M using $\mathbf{A}$; c) determine $(\tilde{\rho}^M, \tilde{\alpha}^M)$ of the line that passes through these two points.

To verify the correctness of the homography found, we calculated the re-projection error using the points detected in the image and their counterparts worldwide. The average error was calculated at *e = 1.5 cm*. To facilitate interpretation of this value, the figure shows a circle of *e* radius drawn on the pattern used.

4.3 Sensor noise

As it is showed in Figure 3, the camera position used in this work is not parallel to the floor, but at a slope. The resulting effect caused by the camera slope can be seen in Figures 6 and 7. From experimentation, one observed that existing information at the top of the image suffered higher noise if compared to the bottom area, what made us consider that noise variance must be proportional to the distance (ρ) of the straight line on the image. Besides, one noticed that quality of horizontal lines which appeared on the image was better than that of vertical ones, what allowed us to understand that noise variance was also related to the straight line angle (α) on the image.

If those aspects above are taken into consideration, then the sensor noise variance adopted in this work is in accordance with the Equation (22). The values of the constants a , b and c were calculated through experimentation and their values are: a = 0:004, b = 0:3 and c = 45.

$$\sigma(\rho,\alpha) = a + b.\sin(\alpha).(exp^{\frac{\rho}{c}} - 1) \tag{22}$$

In this equation, the term $[exp^{\frac{\rho}{c}} - 1]$ represents the distance proportionality, and the term $[sin(\alpha)]$, the angle influence. Figure 8 shows the behavior of the function described by Equation (22) using the values of a, b and c already submitted, and given ρ in meters and α in degrees.

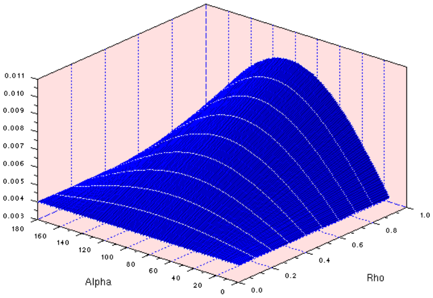

Fig. 8. Noise variance function.

5. Results

The experiments were carried out using the Karel robot, a reconfigurable mobile platform built in our laboratory that has coupled to the structure, a webcam and a laptop for information processing (Figure 3). 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.

To validate the proposed system, results were obtained in two different environments: one containing only a pattern of lines and another containing two patterns of lines. The first experiment was carried out by making the robot navigate in an environment where there are vertical lines on the floor: (I) was command the robot to move forward by 25m, (II) rotate 90 degrees around its own axis (III) move forward 5m, (IV) 180 rotate around its own axis (V) move forward 5m, (VI) rotate 90 degrees around its axis and, finally, walking forward 25m. Figure 9 shows the map of the environment and the task commanded to the robot.

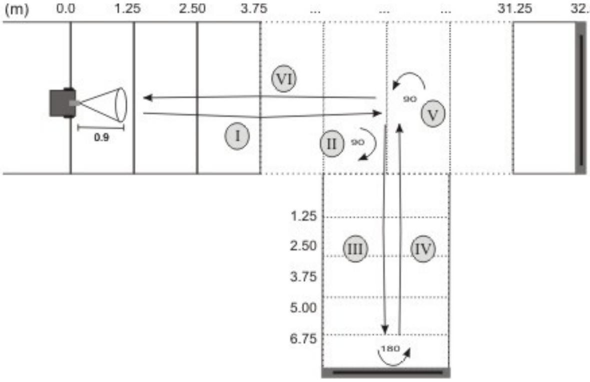

Fig. 9. Experience 01.

In this experiment, during the full navigation of the robot 1962 images were processed and the matching process was successful in 93% of cases. The average observation of each line was 23 times.

In this work the sensors used have different sampling rates. We decided to use the encoders reading in a coordinated manner with the image capture. The camera captures images 640 x 480 (Figure 6) and each image is processed, on average, 180 ms. Figure 10 shows the graphs of the acquisition time (image and encoder), processing time and total time of the system, including acquisition, processing and calculations of the localization algorithm. The average time of acquisition was 50 ms, the processing was 125 ms and the average total time the system was 180 ms. The peaks on the graph made themselves available after the first turning motion of the robot (II), or after it enters a new corridor with different lighting.

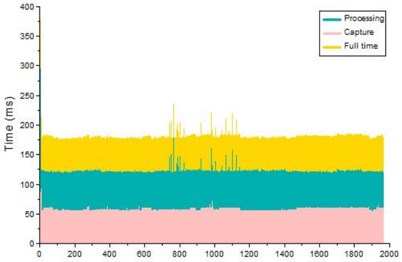

Fig. 10. Times of the system.

About the homography, Figure 7.a shows the pattern that was used at the beginning of the experiment to calculate it. The camera was positioned so that it was possible to have a viewing angle of about twice the size of the robot. It is important to remember that the camera position is such that the image plane is not parallel to the floor plan. Equation (23) shows the homography matrix used.

$$\mathbf{A} = \begin{pmatrix} 0.1417 & 0.0009 & -49.2065 \\ 0.0073 & -0.0761 & 98.6323 \\ 0.0001 & 0.0029 & 1 \end{pmatrix} \tag{23}$$

Besides the proposed system, another location system was also implemented: location system using geometric correction. In this system, every step, the lines are identified and used to calculate the robot pose using trigonometry. When there are no lines identified, the robot pose is calculated by odometry. Figure 11 shows the trajectories calculated using EKF, Geometric Correction and Odometry. It is easy to see that the behavior of the system based on Kalman filter (proposed system) was more satisfactory. The final error, measured *in-loco*, was 0.27m to the system using EKF, 0.46m using the geometric correction system and 0.93m using only odometry.

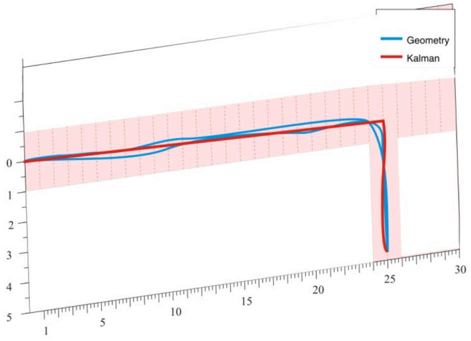

Fig. 11. Trajectories.

As previously mentioned, another experiment was performed in an environment where there are two patterns of lines on the floor, horizontal and vertical. In this environment, the robot was commanded to move forward for 25m (I), rotate 180 degrees around its axis (II) and move forward by 25m (III). Figure 12 shows the position of the lines and controlled the robot trajectory.

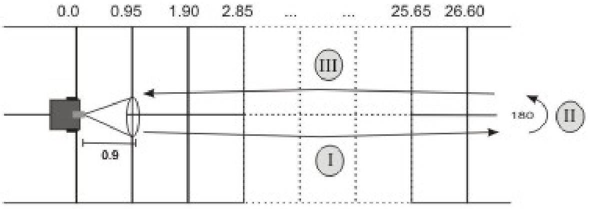

Fig. 12. Experience 02.

In this second experiment, the matching process was successful in 95% of cases. Considering the full navigation of the robot, 2220 images were processed and found that in 87% of step lines were observed (61% and 26% a line two lines). The final error, measured *in-loco* was lower than that found in Experiment 0.16m and allowed us to infer that for greater precision of the proposed system is not enough just a lot of lines in the environment, but rather, they are competitors.

6. Conclusions and Perspectives

This paper presented a localization system for mobile robots using fusion of visual data and odometer data. The main contribution is the modeling of the optical sensor made such a way that it permits using the parameters obtained in the image processing directly to equations of the Kalman Filter without intermediate stages of calculation of position or distance.

Our approach has no pretension to be general, as it requires a flat floor with lines. However, in the cases where can be used (malls, museums, hospitals, homes, airports, etc.) when compared with another approach using geometric correction was more efficient.

As future works, we intend: to improve the real-time properties of the image processing algorithm, by adopting some of the less time consuming variants of the Hough transform; Replace the Kalman Filter by a Filter of Particles, having in view that the latter incorporates more easily the nonlinearities of the problem, besides leading with non-Gaussian noises; Develop this strategy of localization to a proposal of SLAM (Simultaneous Localization and Mapping), so much that robot is able of doing its localization without a previous knowledge of the map and, simultaneously, mapping the environment it navigates.

7. References

Aiube, F. , Baidya T. and Tito, E. (2006), Processos estocásticos dos preços das commodities: uma abordagem através do filtro de partículas, *Brazilian Journal of Economics*, Vol.60, No.03, Rio de Janeiro, Brasil.

Amarasinghe, D., Mann, G. and Gosine, R. (2009), Landmark detection and localization for mobile robot applications: a multisensor approach, *Robotica Cambridge*.

Bezerra, C. G. (2004), Localização de um robô móvel usando odometria e marcos naturais. *Master Thesis*, Federal University of Rio Grande do Norte, Natal, RN, Brasil.

Borenstein, J., Everett, H., Feng, L., and Wehe, D. (1997), Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems*, pp. 231–249.

Bradski, G. and Kaehler, A. (2008), *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media.

Canny, J. (1986), A computational approach to edge detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 679 -698.

Gonzalez, R. C. and Woodes, R. E. (2007), *Digital Image Processing*. Prentice Hall.

Hough, P. V. C (1962), Method and means for recognizing complex patterns, *US Pattent 3069654*, Dec. 18.

Kalman, R. E. (1960), A new approach to linear filtering and predictive problems, *Transactions ASME, Journal of basic engineering*.

Kiriy, E. and Buehler, M. (2002), Three-state extended Kalman filter for mobile robot localization. *Report Centre for Intelligent Machines - CIM*, McGill University.

Launay, F., Ohya, A., and Yuta, S. (2002), A corridors lights based navigation system including path definition using a topologically corrected map for indoor mobile robots. *IEEE International Conference on Robotics and Automation*, pp.3918-3923.

Marzorati, D., Matteucci, M., Migliore, D. and Sorrenti, D. (2009), On the Use of Inverse Scaling in Monocular SLAM, *IEEE Int. Conference on Robotics and Automation*, pp. 2030-2036.

Odakura, V., Costa, A. and Lima, P. (2004), Localização de robôs móveis utilizando observações parciais, *Symposium of the Brazilian Computer Society*.

Park, K. C., Chung, D., Chung, H., and Lee, J. G. (1998), Dead reckoning navigation mobile robot using an indirect Kalman filter. *Conference on Multi-sensor fusion* and Integration for Intelliget Systems, pp. 107-118.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press.

Wu, E., Zhou, W., Dail, G. and Wang, Q. (2009), Monocular Vision SLAM for Large Scale Outdoor Environment, *IEEE Int. Conference on Mechatronics and Automation,* pp. 2037-2041, (2009).