Int. J. Appl. Math. Comput. Sci., 2012, Vol. 22, No. 2, 409–421
DOI: 10.2478/v10006-012-0031-8

amcs

STRAIGHT–LINES MODELLING USING PLANAR INFORMATION FOR MONOCULAR SLAM

ANDRÉ M. SANTANA *, ADELARDO A.D. MEDEIROS *,**

* Department of Informatics and Statistics
Federal University of Piauí, Teresina, PI, 64049-550, Brasil
e-mail: andremacedo@ufpi.edu.br

** Department of Computer Engineering and Automation
Federal University of Rio Grande do Norte, Natal, RN, 59078-970, Brasil
e-mail: adelardo@dca.ufrn.br

This work proposes a SLAM (*Simultaneous Localization And Mapping*) solution based on an Extended Kalman Filter (EKF) in order to enable a robot to navigate along the environment using information from odometry and pre-existing lines on the floor. These lines are recognized by a Hough transform and are mapped into world measurements using a homography matrix. The prediction phase of the EKF is developed using an odometry model of the robot, and the updating makes use of the line parameters in Kalman equations without any intermediate stage for calculating the distance or the position. We show two experiments (indoor and outdoor) dealing with a real robot in order to validate the project.

Keywords: SLAM, Kalman filter, Hough transform, monocular vision.

1. Introduction

In SLAM problems, a mobile robot uses its sensors to explore surroundings, construct a suitable map, and then compute its relative positioning from this map. These maps can be represented in several manners, such as an occupancy grid or a feature map. We are interested in the latter form of representation. Further theoretical details on the SLAM technique can be found in the works of Durrant-Whyte and Bailey (2006a; 2006b) as well as Skrzypczyński (2009).

The literature makes use of the expression *visual SLAM* when referring to the process of a robot using cameras as the main sensor to solve SLAM problems. The main challenges are as follows:

(a) how to detect features in images;

(b) how to recognize that a detected feature is or is not the same as a previously detected one;

(c) how to decide if a new detected feature will or will not be adopted as a new mark;

(d) how to calculate the 3D position of the marks from 2D images; and

(e) how to estimate the uncertainty associated with the calculated values.

Generally speaking, all these aspects must be solved. However, in special cases, it is possible to develop specific strategies to overcome all these problems.

The introduced system presents a visual SLAM technique considering a plane environment and lines on the floor. This is not a very restricting condition, since a lot of environments, such as universities, malls, museums, hospitals, houses, and airports, have lines on the floor.

The algorithm used is based on an Extended Kalman Filter (EKF) in order to enable a robot to navigate along the environment fusing odometry information and monocular vision. The image processor identifies lines on the floor using a Hough Transform (HT), and their parameters are used as marks of the environment. The prediction phase of the filter is carried out using the kinematic model of the robot. Then, during the updating phase, the parameters of the straight-line returned by Hough are mapped to world measurements using a homography matrix, and these values are used in Kalman equations without intermediate stage for calculating the distance or the position.

Using the preexisting lines as marks, the overall complexity of the SLAM problem is reduced, since

(a) lines can be easily detected in images;

(b) lines on the floor are usually equally spaced well apart, so the possibility of confusion is reduced;

(c) as the number of lines in an image is not so big, every newly detected line can define a new mark;

(d) a flat floor is a 2D surface and then there is a constant and easily computable conversion matrix (a homography) between the image plane and the floor plane, without uncertainties concerning the 3D depth of points; and

(e) after processing, the number of pixels in the image belonging to the line is a good measure of confidence in the detected mark.

2. Theoretical background

As mentioned previously, our proposal is based on the extended Kalman filter, the use of a Hough transform to extract information from images, and the application of a homography to develop the world image map. Thus, the purpose of this section is to briefly show these three topics, as well as to make a summary of the proposed system.

2.1. Extended Kalman filter. The extended Kalman filter deals with a system modeled by Eqn. (1), whose variables are described in Table 1. ε_t and δ_t are supposed to be zero-mean Gaussian white noises. The prediction and update phases of the EKF are based on Eqns. (2) and (3), respectively:

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

$$\begin{cases} \bar{\mu}_t = p(\mu_{t-1}, \mathbf{u}_{t-1}, 0), \\ \bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{V}_t \mathbf{M}_t \mathbf{V}_t^T, \end{cases} \tag{2}$$

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

where

$$\mathbf{G}_t = \left. \frac{\partial p(\mathbf{s}, \mathbf{u}, \varepsilon)}{\partial \mathbf{s}} \right|_{\mathbf{s}=\mu_{t-1}, \mathbf{u}=\mathbf{u}_{t-1}, \varepsilon=0}, \tag{4}$$

$$\mathbf{V}_t = \left. \frac{\partial p(\mathbf{s}, \mathbf{u}, \varepsilon)}{\partial \varepsilon} \right|_{\mathbf{s}=\mu_{t-1}, \mathbf{u}=\mathbf{u}_{t-1}, \varepsilon=0}, \tag{5}$$

$$\mathbf{H}_t = \left. \frac{\partial h(\mathbf{s})}{\partial \mathbf{s}} \right|_{\mathbf{s}=\mu_{t-1}}. \tag{6}$$

Table 1. Symbols in Eqns. (1)–(3).

$\mathbf{s}_t$	state vector (order n) at instant t—robot pose (x, y, θ) and map (ρ^i,α^i, ..., ρ^n,α^n)
$p(\cdot)$	non-linear model of the system
$\mathbf{u}_{t-1}$	input signals (order l), instant $t-1$
ε_{t-1}	process noise (order q), instant $t-1$
$\mathbf{z}_t$	vector of measurements (order m) returned by the sensors
$h(\cdot)$	non-linear model of the sensors
δ_t	measurement noise
$\bar{\mu}_t$, μ_t	mean (order n) of the state vector $\mathbf{s}_t$, before and after the update phase
$\bar{\Sigma}_t$,Σ_t	covariance ($n \times n$) of the state vector $\mathbf{s}_t$
$\mathbf{G}_t$	Jacobian matrix ($n \times n$) that linearizes the system model $p(\cdot)$
$\mathbf{V}_t$	Jacobian matrix ($n \times q$) that linearizes the process noise ε_t
$\mathbf{M}_t$	covariance ($q \times q$) of the process noise ε_t
$\mathbf{K}_t$	gain of the Kalman filter ($n \times m$)
$\mathbf{H}_t$	Jacobian matrix ($m \times n$) that linearizes the model of the sensors $h(\cdot)$
$\mathbf{Q}_t$	covariance matrix ($m \times m$) of the measurement noise δ_t

2.2. Hough transform. Due to the choice of the floor lines as marks, the adopted technique to identify them was the Hough transform (Hough, 1962). The purpose of this technique is to identify imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a space of parameters from which candidate objects are obtained as local maxima in an accumulation matrix which is constructed by the algorithm for computing the Hough transform (Bradski and Kaehler, 2008).

When the classical Hough transform is used to identify lines, its return is two parameters (ρ and α) for each obtained straight line in the image (Gonzalez and Woodes, 2007). The interpretation of these parameters is as follows: ρ is the module and α is the angle of the shortest vector connecting the origin of the system of coordinates to the line (see Fig. 1). The relation between them is

$$\rho = x \cos(\alpha) + y \sin(\alpha). \tag{7}$$

From a practical point of view, Forsyth and Ponce (2002) relate the quantification of the accumulator matrix and the influence of the image noise as Hough problems. As a solution to these problems, they suggest the use of an edge detector to smooth the texture and increase the border contrast, as well as choosing the size of the accumulator matrix carefully in accordance with the application.

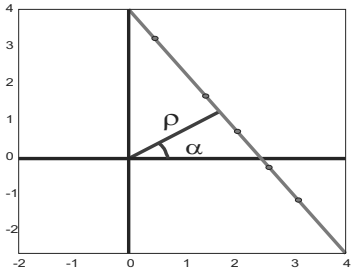

Fig. 1. Line parameters ρ and α.

2.3. Homography. In computer vision, planar homography is defined as a projective mapping from a planar surface to another. Thus, the mapping of points from a two-dimensional planar surface to the image plane of a camera is an example of planar homography.

It is possible to express this mapping as a matrix multiplication if we use homogeneous coordinates to express the world point Q, as well as the image point q which is mapped from Q. Then, one can express the action of the homography matrix $\mathbf{A}$ as

$$\tilde{q} = s \cdot \mathbf{A} \cdot Q, \tag{8}$$

regarding s as an arbitrary scale factor.

2.4. Proposed system. The system proposed in this work presents a suitable visual SLAM technique to plane environment with lines on the floor. The algorithm used is based on information about odometry and image processing. The lines are identified by the Hough transform, and the parameters of the normal representation of the straight-line returned by the Hough transform are mapped to the world using a homography matrix. The EKF prediction phase is developed using the odometry model of the robot, and the updating phase uses the line parameters directly in Kalman equations without any intermediate stage for calculating the distance or the position. Figure 2 shows this proposal.

It is important to note that the proposed approach has no intention to be generic, since a plane floor with lines is necessary. This is not such a restrictive condition, since that is the case in many buildings such as universities, shopping malls, museums, hospitals, homes, supermarkets and airports that have lines on the floor. This work shows that satisfactory results are verified if this approach can be applied.

3. Related work

Recent extensions to the general SLAM problem show it is possible to use a camera instead of a sonar or a laser. For example, Davison and Murray (2002), Jung (2004), Moreno *et al*. (2009) and Herath *et al*. (2007)

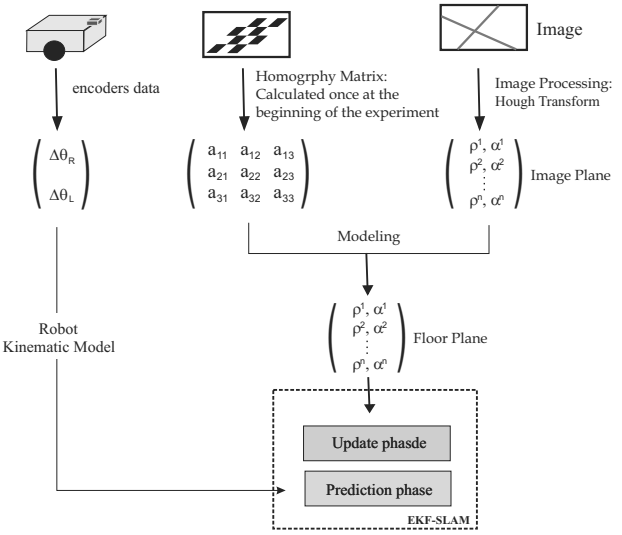

Fig. 2. Proposed system.

make use of stereo vision, while Davison *et al*. (2004) and Kwok *et al*. (2005) use only one camera.

Mansinghka (2004) shows visual SLAM for a dynamic environment using the Scale-Invariant Feature Transform (SIFT) and optical flow. Estrada *et al*. (2005) suggest a hierarchical mapping method that makes it possible to obtain precise metric maps of large environments in real time. The bottom level of the map is formed by a set of local maps which are statistically independent. The upper part of the map is a neighborhood graph whose arcs are marked with the relation between the localization of the local maps, and a relative estimation of these maps is kept at this level through a stochastic relation. The main advantage of this approach is its precision to the closed-loop problem. Williams *et al*. (2009) draw comparisons between monocular SLAM techniques and the closed-loop problem.

Chen and Samarabandu (2006) show a solution from geometric information about the environment. They explain that SLAM redundancy can increase the reliability and precision of the observed features. Thus the common geometric primitives in indoor environments (lines, and squares) are, for instance, incorporated in an EKF to increase the knowledge level of the observed feature.

Frintrop *et al*. (2006) introduce a new method that is a biologically inspired attention system to identify contrasting regions in an image. This approach makes regions more easily re-identified, and thus easier to be matched.

Dailey and Parnichkun (2005) use stereo vision to a visual SLAM based on particle filter. Choi *et al*. (2006) show an approach fusing information from a stereo vision sonar using an extended Kalman filter.

Automatic identification and register of objects as visual marks are proposed by Lee and Song (2007). In this case, the SIFT and a contour algorithm are used to distin-

guish objects from the background image. When objects are detected and considered suitable for robot navigation, they are stored and can be used to correct its pose.

Clemente *et al*. (2007) show that SLAM with only one camera transmitting input data can run on a large scale and in real time. They use inverse depth and hierarchical maps during the experiment. Another work on monocular SLAM for outdoors is by Wu *et al*. (2010).

Jing Wu and Zhang (2007) show a camera modeling for visual SLAM. It aims at discovering how to model optical sensor uncertainty and how to develop probabilistic compounds of the model. If we have the deterministic component of the camera calibration process, we can use intrinsic parameters for error re-projection. Then, the errors are found according to a bivariate Gaussian distribution, and the covariance measurement can be calculated considering different distances from the camera.

Recent studies on visual SLAM show how to obtain depth information from one camera (e.g., Civera *et al*., 2008; Marzorati *et al*., 2009; Li *et al*., 2008).

Hafez *et al*. (2008) also give a proposal to estimate profundity and to perform in real time based on the optimization of triangulation techniques. Calway and Cuevas (2008) show resources to identify and then incorporate a high-level structure, such as lines and surfaces, on visual SLAM maps, while Martinez-Carranza and Calway (2009) use planar information.

Lemaire and Lacroix (2007) suggest the use of 3D lines as marks. They analyze the advantages of using 3D lines: these primitives are numerous indoors and, unlike dispersed point maps, which are useful only for localization, a relevant segment map supplies information about the environment structure.

Fu *et al*. (2007) use vertical lines to show a work on combining laser information and a camera in an extended Kalman filter. In this case, the lines are extracted from an image using a Canny detector. Ahn *et al*. (2007) show a strategy for developing a hybrid map for SLAM using 3D points and lines.

The literature covers some older works (Dao *et al*., 2003; Smith *et al*., 2006; Kitanov *et al*., 2007) and recent ones (Kim and Oh, 2008; Wu *et al*., 2009) using lines.

Wongphati *et al*. (2009) develop fast SLAM using vertical straight lines identified by an omnidirectional view system. Fu and Yang (2009) also show an approach using straight lines for indoor SLAM. Amarasinghe *et al*. (2009) give a SLAM proposal based on extended Kalman filter integrating laser and camera so that the identified marks are wall outer points, tables and chairs. Eade and Drummond (2006) show that edges can be used as marks for monocular SLAM.

Unlike the last works presented in this section, our approach uses 2D lines from the environment as marks. The lines are extracted from an image and mapped to the robot plane through the elements of a homography matrix. The assumption that the straight lines are on the same plane provides wide and precise information about its characteristics and enables the use of these characteristics at the possible detection moment.

4. Modelling

4.1. Prediction phase: Process model. If we take into account a differential drive robot (see Fig. 3) so that $\Delta\theta_R$ and $\Delta\theta_L$ are right and left angular movements of the wheels, respectively, and the speed can be considered constant during the sampling period, it is possible to determine the kinematic geometric model of the robot movement through

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

in which

$$\begin{cases} \Delta L = \dfrac{1}{2}(\Delta\theta_R r_R + \Delta\theta_L r_L), \\ \Delta\theta = \dfrac{1}{b}(\Delta\theta_R r_R - \Delta\theta_L r_L). \end{cases} \tag{10}$$

ΔL and $\Delta\theta$ are the robot linear and angular movements; b is the distance between the wheels, and r_R and r_L are the radii of the right and left wheels, respectively. As $\Delta\theta \to 0$, another system obtained from the limit of Eqn. (9) must be used.

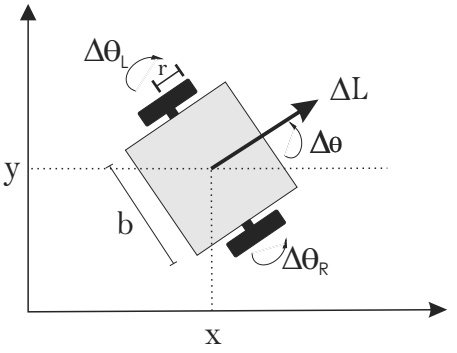

Fig. 3. Kinematic variable model.

The approach supported by Thrun *et al*. (2005), which takes into account the odometry information as input signals to be incorporated to the model of the robot instead of sensory measures, was adopted.

The difference between the real angular movement of the wheels ($\Delta\theta_R$ and $\Delta\theta_L$) and movements measured by encoders ($\Delta\tilde{\theta}_R$ and $\Delta\tilde{\theta}_L$) are molded by a Gaussian white noise: $\Delta\theta_R = \Delta\tilde{\theta}_R + \varepsilon_R$ and $\Delta\theta_L = \Delta\tilde{\theta}_L + \varepsilon_L$. The increments $\Delta\tilde{L}$ and $\Delta\tilde{\theta}$ are defined by switching ($\Delta\theta_R$ and $\Delta\theta_L$) for ($\Delta\tilde{\theta}_R$ and $\Delta\tilde{\theta}_L$) in Eqn. (10).

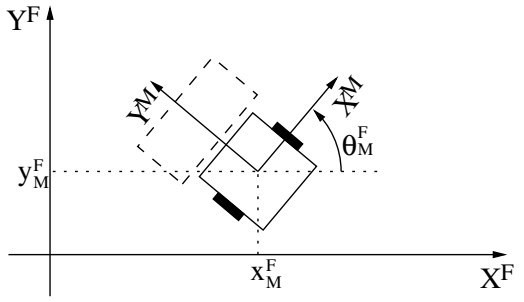

Fig. 4. Fixed and mobile coordinate systems.

Using Eqns. (9) and (10) to determine the state model $p(\cdot)$, it is possible to calculate the matrices **G** and **V** (Eqns. (4) and (5)) used in the prediction phase by model derivation.

It is known that odometry causes accumulative errors. Therefore, the standard deviation of noises ε_R and ε_L is proportional to the modulus of the angular movement of each wheel. This observation admits the specification of the matrix **M** as expressed by

$$\mathbf{M} = \begin{pmatrix} (M_R|\Delta\tilde{\theta}_R|)^2 & 0 \\ 0 & (M_L|\Delta\tilde{\theta}_L|)^2 \end{pmatrix}. \qquad (11)$$

4.2. Updating phase: Sensor model. This section presents a model for a direct mapping of the normal parameters of a straight line from an image to the world. If it is known that a camera is connected to a robot structure, it is necessary to express the parameters of the straight line in relation to the robot pose.

So, a Fixed (F) coordinate system and a Mobile (M) coordinate system, which are attached to the robot, were defined as illustrated in Fig. 4.

Using Eqn. (7) to describe a straight line in world coordinates and the relations between coordinates of the systems (M) and (F), it is possible to obtain Eqns. (12) and (13), which represent the mapping of the straight line parameters of the fixed coordinate system to the mobile coordinate system.[1] 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 one:

$$\begin{cases} \rho^M = \rho^F - x_M^F \cos(\alpha^F) - y_M^F \sin(\alpha^F), \\ \alpha^M = \alpha^F - \theta_M^F + \pi/2, \end{cases} \qquad (12)$$

$$\begin{cases} \rho^M = -\rho^F + x_M^F \cos(\alpha^F) + y_M^F \sin(\alpha^F), \\ \alpha^M = \alpha^F - \theta_M^F - \pi/2. \end{cases} \qquad (13)$$

The next step is to map the line parameters of the mobile coordinate system to the image plane. It is known that the mapping between planes can be modeled by a homography matrix **A**. Thus, a point $(x^M, y^M, 1)$ in the plane (M) can be mapped into another point $(x^I, y^I, 1)$ of the image plane (I) using the relation expressed in Eqn. (14):

$$s \cdot \begin{pmatrix} x^M \\ y^M \\ 1 \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} x^I \\ y^I \\ 1 \end{pmatrix}. \qquad (14)$$

Using the parametric equation of a straight line, a point $(x, y) \in$ *line* (ρ, α) can be defined by Eqn. (15):

$$\begin{cases} x = \rho \cdot \cos(\alpha) - \lambda \cdot \sin(\alpha), \\ y = \rho \cdot \sin(\alpha) + \lambda \cdot \cos(\alpha). \end{cases} \qquad (15)$$

Expanding Eqn. (14), using Eqn. (15) and manipulating these equations, we obtain Eqn. (16), which represents the mapping from an image to the world (mobile coordinate system) of the straight line parameters, where C_{ij} are the elements of the matrix of **A**-cofactors. For mathematical details, see Appendix.

Using Eqn.(6) to obtain the sensor model $h(\cdot)$, through model derivation it is possible to calculate the matrix **H** used in the updating phase of the filter. Its elements are fully described in Appendix.

4.3. Matching. A very important aspect of the SLAM algorithm is to establish a correspondence between the detected line in the image and one of the marks represented in the state vector. To choose the right mark, firstly, if $\tilde{\alpha}^M \geq 0$, the predicted values for (ρ^F, α^F), using the measured values of $(\tilde{\rho}^M, \tilde{\alpha}^M)$, and the model of Eqn. (16), using Eqn. (12), are calculated; if $\tilde{\alpha}^M < 0$, then Eqn. (16) is calculated using Eqn. (13).[2]

The predicted values are compared with each value $({}^i\rho^F, {}^i\alpha^F)$ in the state vector. If the difference between the predicted value and the best, $({}^i\rho^F, {}^i\alpha^F)$, is small enough, the correspondence is found. Otherwise, one assumes that a new mark was detected and the size of the state vector must be enlarged.

5. Image processing

5.1. Detection of lines. The adopted technique to identify the marks was a Hough transform, since it is a method that aimed to find imperfect instances of the objects (Gonzalez and Woodes, 2007). In this case, these objects are pre-existing lines in the environment. The images are captured in gray scale and converted to black-and-white through the Canny edge detector (Canny, 1986). Figure 5 shows an image of the floor from an indoor environment where one of the experiments was carried out, and Fig. 6 shows images of an outdoor environment.

[1] The two equations are needed because the same line identified by the robot for two different points of view (mirrored situation) gives the same result.

[2] We use a tilde (˜) above the variable to indicate the measured values instead of the calculated ones.

$$\begin{cases} \rho^I = \dfrac{-\rho^M C_{33} + \sin(\alpha^M)C_{32} + \cos(\alpha^M)C_{31}}{\sqrt{(\rho^M C_{23} - \sin(\alpha^M)C_{22} - \cos(\alpha^M)C_{21})^2 + (\rho^M C_{13} - \sin(\alpha^M)C_{12} - \cos(\alpha^M)C_{11})^2}}, \\ \alpha^I = \tan^{-1}\left(\dfrac{\rho^M C_{23} - \sin(\alpha^M)C_{22} - \cos(\alpha^M)C_{21}}{\rho^M C_{13} - \sin(\alpha^M)C_{12} - \cos(\alpha^M)C_{11}}\right). \end{cases} \tag{16}$$

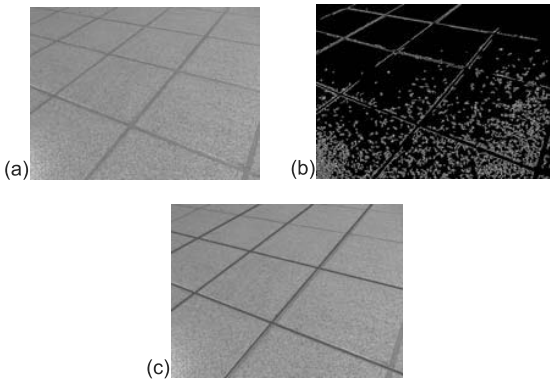

Fig. 5. Image processing: original (a), Canny (b), Hough (c).

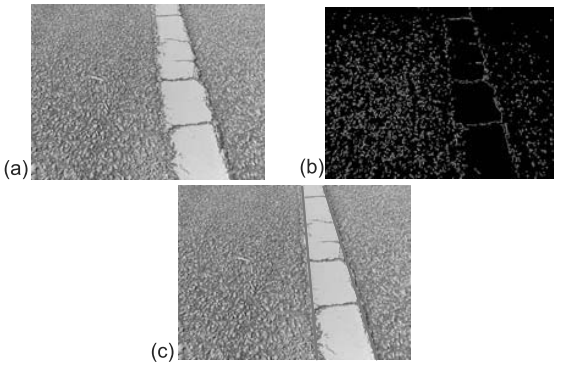

Fig. 6. Image processing: original (a), Canny (b), Hough (c).

5.2. From images to the world. Since the floor is a plane and the camera is fixed, there is a constant relation (a homography $\mathbf{A}$) between the points in the floor plane (x, y) and the points in the image plane (u, v) (see Eqn.(14)).

The scale factor s is determined for each point so that the value of the third element is always 1. The homography can be calculated off-line using a model with four or more notable points whose coordinates are known. After detecting the notable points, one has several correspondences between the coordinates of the points on the plane of the floor and of the image. Substituting these points in Eqn. (14), one finds a linear system in which one can determine the eight elements of the homography matrix.

In this case, the matrix $\mathbf{A}$ is calculated at the beginning of the experiment from an image in which the robot identifies four or more points through the intersection of straight lines detected by the Hough transform. The metric values of the points in the world are measured and provided.

Fig. 7. Identification of the points to calculate the matrix A.

5.3. Sensor noise. Once the sensor modeling is developed so that it allows the mapping of the Hough parameters directly from the image to the world, the noise model takes into consideration the two main components: the acquisition noise and the processing noise of the image. The first is modeled by a Gaussian (as suggested by Forsyth and Ponce (2002)), and the second aggregates the influence of the discretization of the accumulative matrix. To model this influence, the adopted standard deviation of the noise in this work is a measure that takes into account the quantity of received votes (Eqn. (17)) for each straight line in an individual way:

$$\rho(n_{(\rho,\alpha)}) = a \cdot \frac{n_{\max}}{n_{(\rho,\alpha)}} \cdot \psi_t^I. \tag{17}$$

In this equation, $n_{\max}$ is the maximum number of votes that a straight line can receive, $n_{(\rho,\alpha)}$ is the number of the votes received by the straight line, ψ_t^I is the acquisition noise of the image and a is a weighing factor. Through experimentation, the value 0.02 for a was found. The other variables are calculated for each image.

6. Results

6.1. Indoor experiment. The experiments were carried out using a robot whose wheels were powered by DC motors with differential action. Each motor had an optical encoder and a dedicated velocity controller. The robot system also had a color webcam, and a notebook to process the information (see Fig. 8).

During this experiment the robot navigated in an environment that had a floor made out of squared tiles with sides of length 25 cm (see Fig. 5). The intersections of these squared tiles created horizontal and vertical lines that were used as marks in the SLAM algorithm. Figure 9 shows the characteristics of this environment.

The robot performed a pretty much rectangular trajectory in this building, and during its movement 1962 images were processed. The camera used captures 640×480 images and each image is processed, on average, in 180 ms. Figure 10 shows the graphs of the acquisition time of the image, the processing time and the total time of the system including acquisition, processing and SLAM algorithm calculations.

In this graph it is clearly possible to observe two peaks. They occurred due to the change in the illumination in the environment when the robot came near the garden. The processing time increased due to the sensitivity of the Canny algorithm. The average acquisition time was 50 ms, the average processing time was 125 ms, and the average total time was 180 ms.

In relation to homography, Fig. 7 shows the image that was used at the beginning of the experiment to calculate it. The camera was set so that it was possible to have a vision field about twice as large as the robot size. It is important to know that from the camera position the image plane is not parallel to the floor plane (see Fig. 8).

Figure 11 shows the 2D trajectory in meters. The blue points correspond to the trajectory calculated by odometry only, and the red points correspond to the trajectory calculated by SLAM.

This experiment shows that the odometry model is flawed in rotational movements of the robot due to the effect of wheel slippage, and this effect did not have major implications in SLAM.

Considering the total movement carried out by the robot, the correspondence of detected lines was right in 95% of the cases and each line was observed in 15 consecutive images on the average. In 98% of the images, the system detected three lines in 61%, four lines in 26% and five lines in 11% of the images. The remainder of 2% covered errors of the image processing algorithm.

In spite of these 2% of errors, the vision system responded well to the illumination problems (Fig. 12) and the dirt on the floor (Fig. 13).

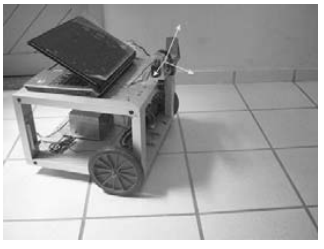

Fig. 8. Robotic system.

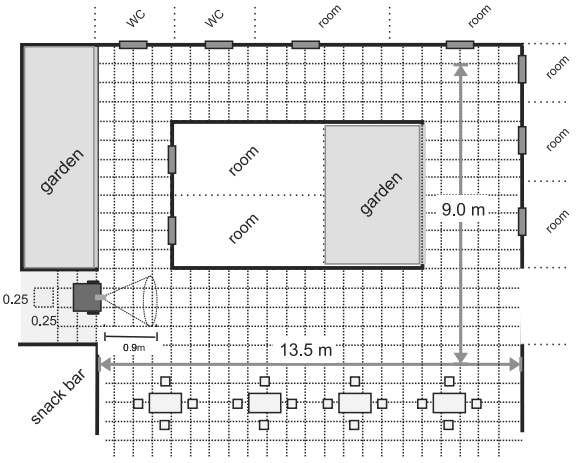

Fig. 9. Description of the environment. The real image can be obtained from online mapping systems using the coordinates $5^o50'29.50''S35^o1'49.48''W$.

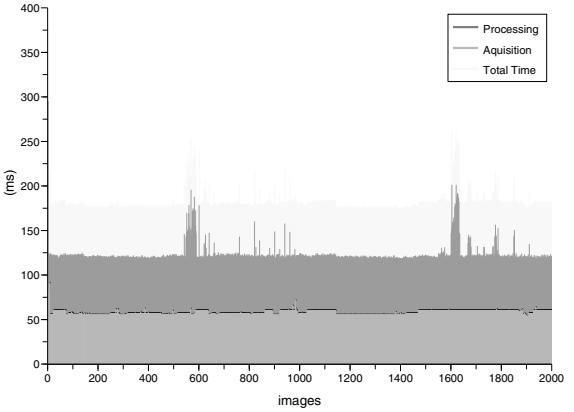

Fig. 10. Times of the system.

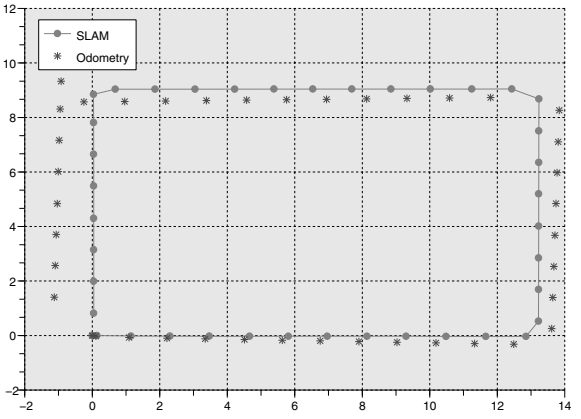

Fig. 11. Trajectory.

In spite of these 2% of errors, the vision system responded well to the illumination problems (Fig. 12) and the dirt on the floor (Fig. 13).

The distance between the initial and final position

calculated from an initial pose $(0, 0, 0^o)$ was 0.89 m using odometry only, and 0.03 m using SLAM. The final real distance measured *in loco* was very close to that calculated by SLAM.

6.2. Structured outdoor experiment. The second experiment was carried out in a supermarket parking lot. This environment (Fig. 14) has lines painted on the floor: white lines to delimit the parking place (see Fig. 6), and yellow lines to indicate the central area of the track where the cars can pass through (see Fig. 16).

In this experiment the robot developed a rectangular trajectory of size 35 m $\times$ 16 m. Figure 15 shows the trajectory (calculated) carried out by the robot using odometry and using the proposed algorithm.

During this movement (about 100 m long), 2566 images were processed and the average processing time of each image was 100 ms. In 78% of images lines were identified. In this percentage, one line was identified in 53%, two lines in 31%, three lines in 11% of the images and four lines in 5%. To illustrate this, Fig. 16 shows a situation in which the image processor should have identified two straight lines, but it did identify only one.

When there was dirt on the floor, the results of the second experiment were not as satisfactory as those in the first one. For instance, Fig. 17 shows a mark that was not identified by the robot.

The correspondence between detected lines was correct in 94% of the cases and each line was observed on the average in seven consecutive images. Despite the small number of observations of each straight line if compared with the indoor experiment, the rate of correspondence was higher. This happened because the straight lines in the environment were more spaced.

In relation to the closed-loop problem, even after the robot navigated for more than 100 m, the behavior of the algorithm was suitable. The error between the robot final position calculated by SLAM and its initial position was 88 cm. This value was measured in the environment

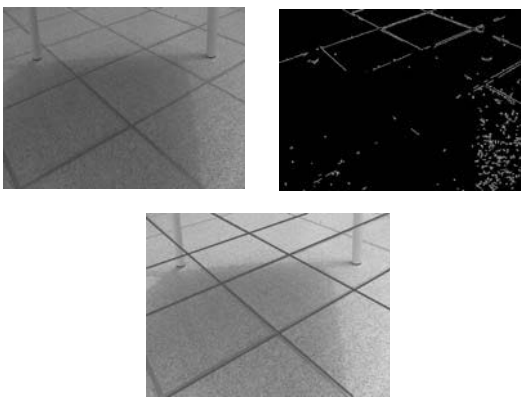

Fig. 12. Resistance to an illumination change.

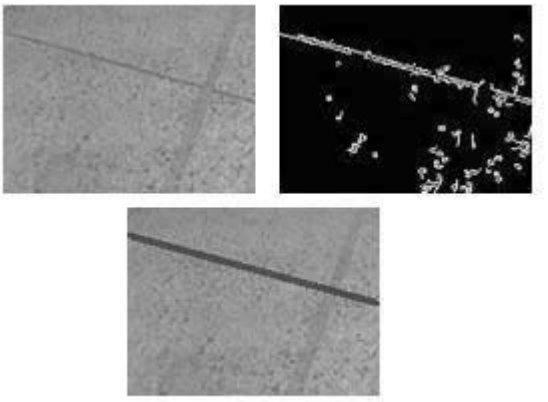

Fig. 13. Resistance to dirt on the floor.

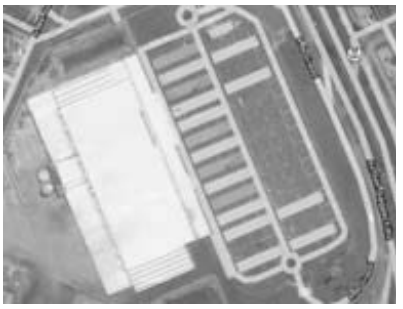

Fig. 14. Supermarket: structured outdoor experiment. The supermarket view obtained from an online mapping system using the coordinates $5^o45'31''S35^o14'50''W$.

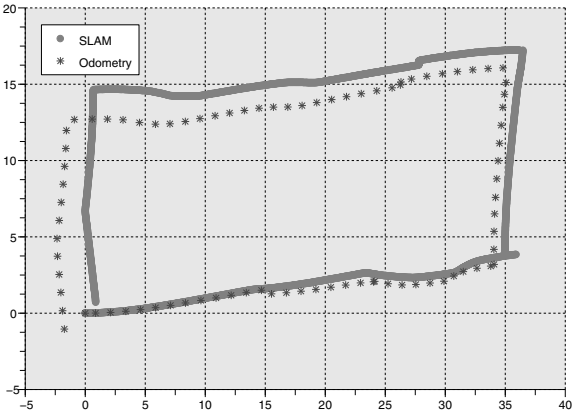

Fig. 15. Trajectory in the supermarket parking.

considering the starting and final position of the robot.

6.3. Comparative results. The classical technique of visual SLAM based on points was used for comparison. It is known that a big problem related to this technique is right initialization of the three coordinates of the detected mark. However, there are works that propose ways to solve this problem, as shown by Civera *et al*. (2008).

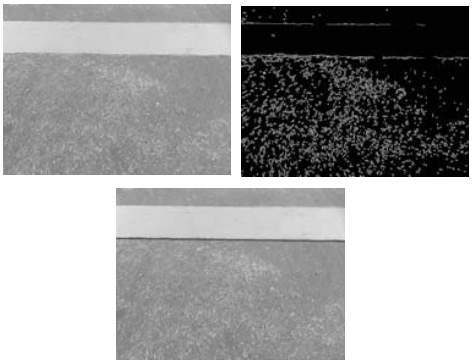

Fig. 16. Identification of only one straight line from the image.

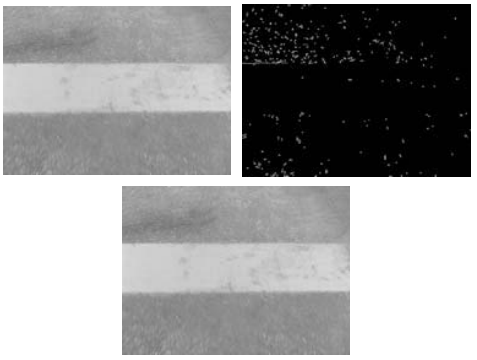

Fig. 17. Some lines not identified from the image due to dirt.

The implemented algorithm to compare our proposal uses as features the coordinates (x, y, z) of the intersection points between the straight lines on the floor. In order to make a fair comparison between the two algorithms, we provided the correct value of the third coordinate of the points, assuming that they all lied on the floor.

Figure 18 shows the trajectories calculated by the robot using the algorithm based on points and using the one based in straight lines. It is noticeable that the approach using straight lines exhibits a more satisfactory behavior than that using points.

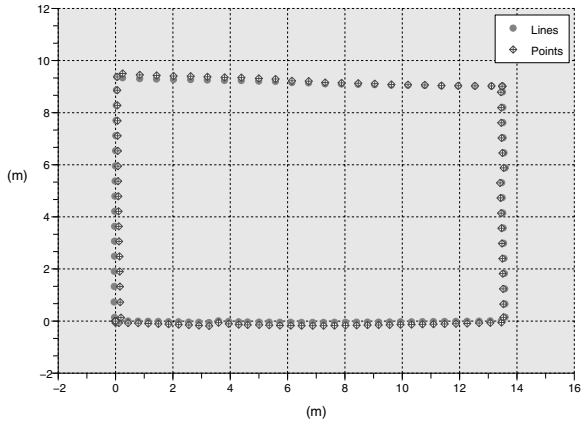

Fig. 18. Trajectory by SLAM using lines and points.

The final error was 0.03 m when using lines and 0.18 m when using points. The average number of observations of each mark was 15 times for the method using lines and 6 times for the method using points. Another comparative result is the quantity of the characteristics used in the SLAM algorithm during the total robot movement. For the approach using points, 1401 features referring to 467 marks were inserted in the state vector, while in the algorithm using straight lines there were 388 features and 194 marks inserted.

Finally, Figs. 19–21 show the variances of the components x, y, θ of the robot pose, respectively. These figures show that the variance behavior of the robot pose was smaller during a great majority of the runs.

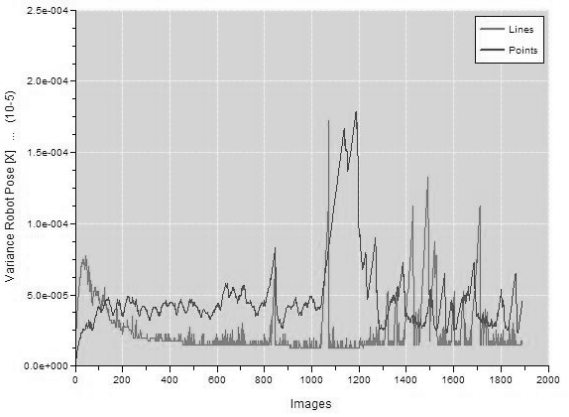

Fig. 19. Variance of the component x of the robot pose.

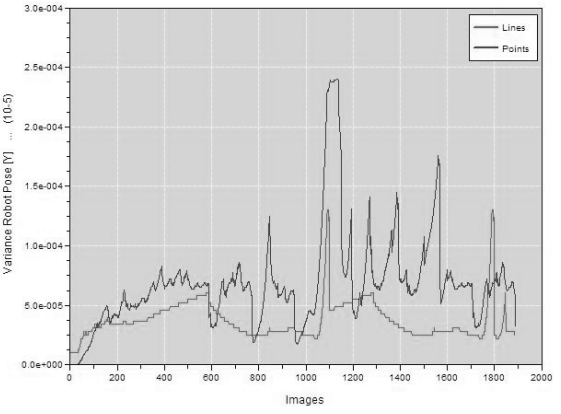

Fig. 20. Variance of the component y of the robot pose.

7. Conclusions and perspectives

The main contribution of this work is the modeling of the optical sensor in order to allow the use of the parameters of the straight line obtained from the processing image algorithm directly in the Kalman filter equations without any intermediate phases for calculating the position and distance.

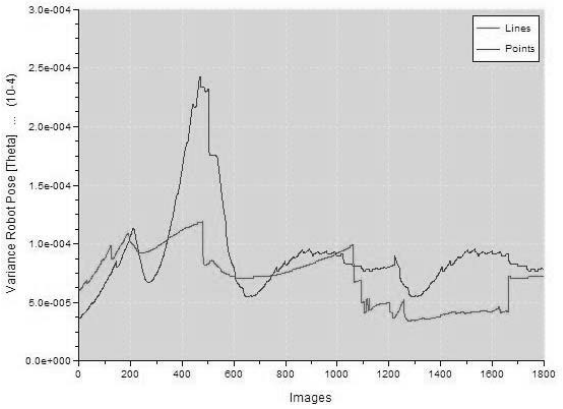

Fig. 21. Variances of the component θ of the robot pose.

Another important aspect of this approach is the use, besides odometry, of a system of monocular vision and no other auxiliary sensors such as lasers or sonars. Despite monocular vision, the straight lines can be used in the Kalman filter at the moment of detection, since their characteristics are mapped directly to world coordinates using the homography matrix.

The proposed approach has no intention to be generic, since a plane floor with lines is necessary. However, if this approach can be used, it is more efficient with regard to computational costs (due to the reduced number of elements to represent the environment) and accuracy (due to the small rate of correspondence errors and accuracy in determining 3D information of the characteristics) when compared with other classical approaches of visual SLAM.

Even with the loop-closed problem, as was shown in the results (Section 6), the system performed well in both indoor and outdoor environments, recognizing previously detected lines without special procedures to deal with this problem.

In future studies, it is necessary to improve the real time properties of the algorithm of image processing through adoption of some variants of the Hough transform, to deal with segments of straight line of finite length, and to verify our approach using other statistic filters: an unscented Kalman filter, an information filter and a particle filter.

References

Ahn, S., Chung, W.K. and Oh, S. (2007). Construction of hybrid visual map for indoor SLAM, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA,* pp. 1695–1701.

Amarasinghe, D., Mann, G. and Gosine, R. (2009). Landmark detection and localization for mobile robot applications: A multisensor approach, *Robotica Cambridge* **28**(5): 663–673.

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

Calway, A. and Cuevas, W. (2008). Discovering higher level structure in visual SLAM, *IEEE Transactions on Robotics* **24**(5): 980–990.

Canny, J. (1986). A computational approach to edge detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6): 679–698.

Chen, Z. and Samarabandu, J. (2006). A visual SLAM solution based on high level geometry knowledge and Kalman filtering, *Canadian Conference on Electrical and Computer Engineering (CCECE), Ottawa, Canada*, pp. 1283–1286.

Choi, J., Ahn, S., Choi, M. and Chung, W. (2006). Metric SLAM in home environment with visual objects and sonar features, *International Conference on Intelligent Robots and Systems (IROS), Beijing, China,* pp. 4048–4053.

Civera, J., Davison, A.J. and Montiel, J.M. (2008). Inverse depth parametrization for monocular SLAM, *IEEE Transactions on Robotics* **24**(5): 932–945.

Clemente, L., Davison, A., Reid, I., Neira, J. and Tardos, J. (2007). Mapping large loops with a single hand-held camera, *Robotics: Science and Systems, Atlanta, GA, USA*.

Dailey, M. and Parnichkun, M. (2005). Landmark based simultaneous localization and mapping with stereo vision, *Asian Conference on Industrial Automation and Robotics (ACIAR), Bangkok, Thailand,* pp. 108–113.

Dao, N., You, B., Oh, S. and Hwangbo, M. (2003). Visual self-localization for indoor mobile robots using natural lines, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA,* pp. 1252–1257.

Davison, A.J., Cid, Y.G. and Kita, N. (2004). Real-time 3D SLAM with wide-angle vision, *Symposium on Intelligent Autonomous Vehicles (IAV), Lisbon, Portugal*.

Davison, A.J. and Murray, D.W. (2002). Simultaneous localization and map-building using active vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(7): 865–880.

Durrant-Whyte, H. and Bailey, T. (2006a). Simultaneous localization and mapping: Part I, *IEEE Transactions on Robotics and Automation* **13**(2): 99–108.

Durrant-Whyte, H. and Bailey, T. (2006b). Simultaneous localization and mapping: Part II, *IEEE Transactions on Robotics and Automation* **13**(3): 109–117.

Eade, E. and Drummond, T. (2006). Edge landmarks in monocular SLAM, *British Machine Vision Conference (BMVC), Edinburgh, UK*, pp. 7–17.

Estrada, C., Neira, J. and Tards, J.D. (2005). Hierarchical SLAM: Real-time accurate mapping of large environments, *IEEE Transactions on Robotics* **21**(4): 588–596.

Forsyth, D. and Ponce, J. (2002). *Computer Vision: A Modern Approach*, Prentice Hall.

Frintrop, S., Jensfelt, P. and Christensen, H.I. (2006). Attentional landmark selection for visual SLAM, *International Conference on Intelligent Robots and Systems (IROS)*.

Fu, S., Liu, H., Gao, L. and Gai, Y. (2007). SLAM for mobile robots using laser range finder and monocular vision, *IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy,* pp. 91–96.

Fu, S. and Yang, G. (2009). Uncalibrated monocular based SLAM for indoor autonomous mobile robot navigation, *International Conference on Networking, Sensing and Control (ICNSC), Porto, Portugal,* pp. 663–668.

Gonzalez, R. and Woodes, R. (2007). *Digital Image Processing*, Prentice Hall, Upper Saddle River, NJ.

Hafez, A., Bhuvanagiri, S., Krishna, M. and Jawahar, C. (2008). On-line convex optimization based solution for mapping in VSLAM, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France,* pp. 4072–4077.

Herath, D.C., Kodagoda, K.R.S. and Dissanayake, G. (2007). Stereo vision based SLAM issues and solutions, *IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy*, pp. 1892–1897.

Hough, P. (1962). Method and means for recognizing complex patterns, *US Patent 3069654*.

Jing Wu, J. and Zhang, H. (2007). Camera sensor model for visual SLAM, *IEEE Canadian Conference on Computer and Robot Vision (CRV), Montreal, Canada,* pp. 149–156.

Jung, I. (2004). *Simultaneous Localization and Mapping in 3D Environments with Stereovision*, Ph.D. thesis, Institut National Polytechnique de Toulouse, Toulouse.

Kim, S. and Oh, S. (2008). SLAM in indoor environments using omni-directional vertical and horizontal line features, *Journal of Intelligent Robotic Systems* **51**(1): 31–43.

Kitanov, A., Bisevac, S. and Petrovic, I. (2007). Mobile robot self-localization in complex indoor environments using monocular vision and 3D model, *International Conference on Advanced Intelligent Mechatronics (ASME), Zurich, Switzerland,* pp. 1–6.

Kwok, N.M., Dissanayake, G. and Ha., Q.P. (2005). Bearing-only SLAM using a SPRT based Gaussian sum filter, *IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain*, pp. 1109–1114.

Lee, Y. and Song, J. (2007). Autonomous selection, registration, and recognition of objects for visual SLAM in indoor environments, *International Conference on Control, Automation and Systems (ICCAS), Seoul, Korea*, pp. 668–673.

Lemaire, T. and Lacroix, S. (2007). Monocular-vision based SLAM using line segments, *IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy*, pp. 2791–2796.

Li, C., Huang, Y., Kang, Y. and Yuan, J. (2008). Monocular SLAM using vertical straight lines with inverse-depth representation, *World Congress on Intelligent Control and Automation (WCICA), Chongqing, China*, pp. 3015–3020.

Mansinghka, V.K. (2004). Towards visual SLAM in dynamic environments, *Report*, MIT, Cambridge, MA.

Martinez-Carranza, J. and Calway, A. (2009). Efficiently increasing map density in visual SLAM using planar features with adaptive measurements, *British Machine Vision Conference (BMVC), London, UK*, pp. 1–11.

Marzorati, D., Matteucci, M., Migliori, D. and Sorrenti, D. (2009). On the use of inverse scaling in monocular SLAM, *IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, pp. 2030–2036.

Moreno, F., Blanco, J. and Gonzalez, J. (2009). Stereo vision specific models for particle filter-based SLAM, *Robotics and Autonomous Systems* **57**(9): 955–970.

Skrzypczyński, P. (2009). Simultaneous localization and mapping: A feature-based probabilistic approach, *International Journal of Applied Mathematics and Computer Science* **19**(4): 575–588, DOI: 10.2478/v10006-009-0045-z.

Smith, P., Reid, I. and Davison, A. (2006). Real-time monocular SLAM with straight lines, *British Machine Vision Conference (BMVC), Edinburgh, UK*, pp. 17–27.

Thrun, S., Burgard, W. and Dieter, F. (2005). *Probabilistic Robotics*, MIT Press, Cambridge, MA.

Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I. and Tardos, J. (2009). A comparison of loop closing techniques in monocular SLAM, *Robotics and Autonomous Systems* **57**(12): 1188–1197.

Wongphati, M., Niparnan, N. and Sudsang, A. (2009). Bearing only FastSLAM using vertical line information from an omnidirectional camera, *IEEE International Conference on Robotics and Biomimetics (ROBIO), Guilin, China*, pp. 1188–1193.

Wu, E., Zhao, L., Guo, Y., Zhou, W. and Wang, Q. (2010). Monocular vision SLAM based on key feature points selection, *IEEE International Conference on Robotics and Automation (ICRA), Povoa de Varzim, Portugal*, pp. 1741–1745.

Wu, M., Huang, F., Wang, L. and Sun, J. (2009). Cooperative multi-robot monocular-SLAM using salient landmarks, *International Asian Conference on Informatics in Control, Automation and Robotics (CAR), Bangkok, Thailand,* pp. 151–155.

André M. Santana, Ph.D. in computer engineering from the Federal University of Rio Grande do Norte (Universidade Federal do Rio Grande do Norte, UFRN), Natal, Brazil (2011), master's degree in electrical engineering from UFRN (2007), graduate in computer science from the Federal University of Piauí (Universidade Federal do Piauí, UFPI), Teresina, PI, Brazil (2004). He is currently an associate professor in the Department of Informatics and Statistics (DIE), UFPI. His scientific interests include mobile robotics, computer vision, stochastic filtering and artificial intelligence.

Adelardo A.D. Medeiros, Ph.D. in robotics from at LAAS-CNRS, Paul Sabatier University in Toulouse, France (1997), master's degree in electrical engineering and computer science from the Airfoce Technological Institute (Instituto Tecnolgico de Aeronutica, ITA), So Jos dos Campos, SP (1991), and graduate in electrical engineering from the Federal University of Rio Grande do Norte (Universidade Federal do Rio Grande do Norte, UFRN), Natal, RN (1988). He is currently an associate professor at UFRN, attached to the Department of Computer Engineering and Automation (DCA). His scientific interests include mobile robots, real-time systems, systems architecture and industrial automation.

Appendix

This appendix shows the modeling of the straight lines so that the parameters can be used directly in the extended Kalman filter equations (see Fig. 22). The mapping of the image plane to world coordinates is done using a homography matrix, calculated once at the beginning of the experiment, and the direct conversion (image to world) is presented below.

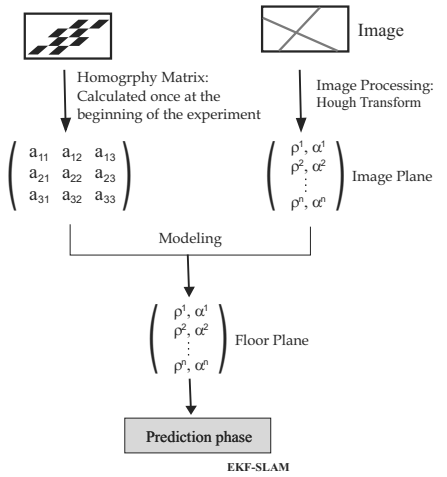

Fig. 22. Modeling scheme.

We should take into account a homography matrix $\mathbf{A}_{3\times3}$ and a matrix of $\mathbf{A}$-cofactors $\mathbf{C}_{3\times3}$, where $C_{ij} = (-1)^{i+j} \cdot \det(\mathbf{A}^*)$ and $\mathbf{A}^*$ can be obtained from $\mathbf{A}$ by eliminating line i and column j:

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix},$$
$$\mathbf{C} = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix}. \tag{18}$$

Using the parametric equation of the straight line (Eqn. (19)) and the homography relation (Eqn. (20)), any point in a plane (I) can be mapped to another plane (F) using the relation expressed in Eqn. (21).

$$\begin{cases} x = \rho \cdot \cos(\alpha) - \lambda \cdot \sin(\alpha), \\ y = \rho \cdot \sin(\alpha) + \lambda \cdot \cos(\alpha), \end{cases} \tag{19}$$

$$s \cdot \begin{pmatrix} x^I \\ y^I \\ 1 \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} x^F \\ y^F \\ 1 \end{pmatrix}, \tag{20}$$

$$x^I = \frac{K_x \cdot \lambda + C_x}{K_t \cdot C_t}, \quad y^I = \frac{K_y \cdot \lambda + C_y}{K_t \cdot C_t}, \tag{21}$$

so that

$$\begin{cases} K_x = a_{12} \cdot \cos(\alpha) - a_{11} \cdot \sin(\alpha), \\ K_y = a_{22} \cdot \cos(\alpha) - a_{21} \cdot \sin(\alpha), \\ K_t = a_{32} \cdot \cos(\alpha) - a_{31} \cdot \sin(\alpha), \\ C_x = a_{11} \cdot \rho \cdot \cos(\alpha) + a_{12} \cdot \rho \cdot \sin(\alpha) + a_{13}, \\ C_y = a_{21} \cdot \rho \cdot \cos(\alpha) + a_{22} \cdot \rho \cdot \sin(\alpha) + a_{23}, \\ C_t = a_{31} \cdot \rho \cdot \cos(\alpha) + a_{32} \cdot \rho \cdot \sin(\alpha) + a_{33}. \end{cases} \tag{22}$$

Using the normal representation of the straight line

$$\rho = x\cos(\alpha) + y\sin(\alpha) \tag{23}$$

and manipulating Eqn. (21), Eqn. (24), which shows the mapping between (I) and (F) planes, can be obtained.

It is necessary to find a relationship between the mobile coordinate system and the fixed coordinate system. Thus a fixed coordinate system (F) and a mobile coordinate system (M) were defined, see Fig. 23.

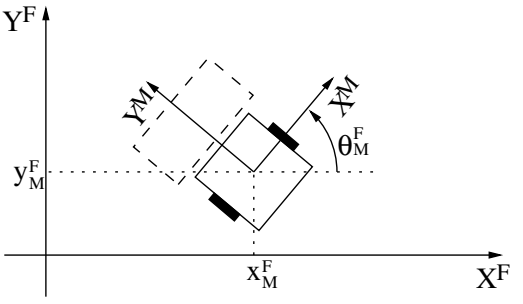

Fig. 23. Fixed and mobile coordinate systems.

The origin of the system has coordinates (x_M^F, y_M^F) in the fixed system and θ_M^F represents the rotation of the mobile system in relation to the fixed one. It is important to note that there is a close relation between these variables $(x_M^F, y_M^F, \theta_M^F)$ and the robot pose (x_t, y_t, θ_t), which is given by

$$x_t = x_M^F, \qquad y_t = y_M^F, \qquad \theta_t = \theta_M^F + \pi/2. \tag{25}$$

$$\begin{aligned}\rho^I &= \frac{-\rho^F C_{33} + \sin(\alpha^F)C_{32} + \cos(\alpha^F)C_{31}}{\sqrt{(\rho^F C_{23} - \sin(\alpha^F)C_{22} - \cos(\alpha^F)C_{21})^2 + (\rho^F C_{13} - \sin(\alpha^F)C_{12} - \cos(\alpha^F)C_{11})^2}},\\ \alpha^I &= \tan^{-1}\left(\frac{\rho^F C_{23} - \sin(\alpha^F)C_{22} - \cos(\alpha^F)C_{21}}{\rho^F C_{13} - \sin(\alpha^F)C_{12} - \cos(\alpha^F)C_{11}}\right),\end{aligned} \tag{24}$$

$$\begin{cases}\rho^I = \dfrac{-\rho^M C_{33} + \sin(\alpha^M)C_{32} + \cos(\alpha^M)C_{31}}{\sqrt{(\rho^M C_{23} - \sin(\alpha^M)C_{22} - \cos(\alpha^M)C_{21})^2 + (\rho^M C_{13} - \sin(\alpha^M)C_{12} - \cos(\alpha^M)C_{11})^2}},\\ \alpha^I = \tan^{-1}\left(\dfrac{\rho^M C_{23} - \sin(\alpha^M)C_{22} - \cos(\alpha^M)C_{21}}{\rho^M C_{13} - \sin(\alpha^M)C_{12} - \cos(\alpha^M)C_{11}}\right).\end{cases} \tag{31}$$

The relation between the coordinates of the systems (M) and (F) (Eqn. (26)) and Eqn. (23) in both the coordinate systems (Eqns. (27) and (28)) was used:

$$\begin{cases}x^F = x^M \cdot \cos(\theta_M^F) - y^M \cdot \sin(\theta_M^F) + x_M^F,\\ y^F = x^M \cdot \sin(\theta_M^F) + y^M \cdot \cos(\theta_M^F) + y_M^F,\end{cases} \tag{26}$$

$$\rho^F = x^F \cdot \cos(\alpha^F) + y^F \cdot \sin(\alpha^F), \tag{27}$$

$$\rho^M = x^M \cdot \cos(\alpha^M) + y^M \cdot \sin(\alpha^M), \tag{28}$$

To obtain Eqns. (29) and (30), it is necessary to substitute Eqn. (26) in Eqn. (27), exploit the equivalences with Eqn. (28), and substitute some variables using Eqn. (25):

$$\begin{cases}\rho^M = \rho^F - x_t \cdot \cos(\alpha^F) - y_t \cdot \sin(\alpha^F),\\ \alpha^M = \alpha^F - \theta_t + \pi/2,\end{cases} \tag{29}$$

$$\begin{cases}\rho^M = -\rho^F + x_t \cdot \cos(\alpha^F) + y_t \cdot \sin(\alpha^F),\\ \alpha^M = \alpha^F - \theta_t - \pi/2.\end{cases} \tag{30}$$

Finally, replacing Eqn. (29) or (30) in Eqn. (24), we obtain Eqn. (31).

The matrix $\mathbf{H}$ used in the Kalman filter is defined by

$$\mathbf{H}_t = \left.\frac{\partial h(\mathbf{s})}{\partial \mathbf{s}}\right|_{\mathbf{s}=\mu_{t-1}}. \tag{32}$$

Thus,

$$\begin{cases}\dfrac{\partial \alpha^I}{\partial x_t} = -\cos(\alpha^F) \cdot \dfrac{\partial \alpha^I}{\partial \rho^F},\\ \dfrac{\partial \rho^I}{\partial x_t} = -\cos(\alpha^F) \cdot \dfrac{\partial \rho^I}{\partial \rho^F},\\ \dfrac{\partial \alpha^I}{\partial y_t} = -\sin(\alpha^F) \cdot \dfrac{\partial \alpha^I}{\partial \rho^F},\end{cases} \tag{33}$$

$$\begin{cases}\dfrac{\partial \rho^I}{\partial y_t} = -\sin(\alpha^F) \cdot \dfrac{\partial \rho^I}{\partial \rho^F},\\ \dfrac{\partial \alpha^I}{\partial \theta_t} = -K \cdot \det(\mathbf{A}) \cdot f_1,\\ \dfrac{\partial \rho^I}{\partial \theta_t} = \sqrt{K^3} \cdot \det(\mathbf{A}) \cdot \Big[\rho \cdot \cos(\alpha^M) \cdot f_3 - \rho \cdot \sin(\alpha^M) \cdot f_4 + f_5\Big],\\ \dfrac{\partial \alpha^I}{\partial \rho^M} = -K \cdot \det(\mathbf{A}) \cdot f_2,\\ \dfrac{\partial \rho^I}{\partial \rho^M} = \sqrt{K^3} \cdot \det(\mathbf{A}) \cdot [\sin(\alpha^M) \cdot f_3 - \cos(\alpha^M) \cdot f_4],\\ \dfrac{\partial \alpha^I}{\partial \alpha^M} = y_t \cdot \dfrac{\partial \alpha^I}{\partial x_t} - x_t \cdot \dfrac{\partial \alpha^I}{\partial y_t} - \dfrac{\partial \alpha^I}{\partial \theta_t},\\ \dfrac{\partial \rho^I}{\partial \alpha^M} = y_t \cdot \dfrac{\partial \rho^I}{\partial x_t} - x_t \cdot \dfrac{\partial \rho^I}{\partial y_t} - \dfrac{\partial \rho^I}{\partial \theta_t},\end{cases} \tag{34}$$

and

$$\begin{aligned}K &= [(K_x C_t - K_t C_x)^2 + (K_t C_y - K_y C_t)^2]^{-1},\\ f_1 &= \rho^M \cdot a_{31} \cdot \cos(\alpha^M) + \rho^M \cdot a_{32} \cdot \sin(\alpha^M) - a_{33},\\ f_2 &= a_{31} \cdot \sin(\alpha^M) - a_{32} \cdot \cos(\alpha^M),\\ f_3 &= T_{13} \cdot \rho^M - T_{12} \cdot \sin(\alpha^M) - T_{11} \cdot \cos(\alpha^M),\\ f_4 &= T_{23} \cdot \rho^M - T_{22} \cdot \sin(\alpha^M) - T_{21} \cdot \cos(\alpha^M),\\ f_5 &= T_{33} \cdot \rho^M - T_{32} \cdot \sin(\alpha^M) - T_{31} \cdot \cos(\alpha^M).\end{aligned}$$

The elements a_{ij} are from the homography matrix $\mathbf{A}$ and the elements C_{ij} are from cofactor $\mathbf{A}$ while t_{ij} are elements of the matrix $\mathbf{T}$:

$$\mathbf{T} = \begin{pmatrix}h_{11} & h_{21}\\ h_{12} & h_{22}\\ h_{13} & h_{23}\end{pmatrix} \cdot \begin{pmatrix}C_{21} & C_{22} & C_{23}\\ C_{11} & C_{12} & C_{13}\end{pmatrix}. \tag{35}$$

Received: 5 January 2011
Revised: 23 August 2011