Mobile Robot Localization and Map Building using Monocular Vision

José Neira, María Isabel Ribeiro, Juan Domingo Tardós

Universidad de Zaragoza
Centro Politécnico Superior (CPS)
Departamento de Informática e Ingeniería de Sistemas
c/María de Luna, 3,
E-50015 Zaragoza, SPAIN
email: neira@prometeo.cps.unizar.es, jdtardos@posta.unizar.es

Instituto Superior Técnico (IST)
Instituto de Sistemas e Robótica (ISR)
Torre Norte - Piso 8
Av.Rovisco Pais, 1
P 1096 Lisboa CODEX PORTUGAL
email: mir@isr.ist.utl.pt

Abstract. In this paper we analyze the fundamental aspects of the construction of indoor maps suitable for their use in navigation tasks by a mobile robot. We highlight the limitations inherent to the use of *a priori* maps. We show the fundamental problems related to the simultaneous construction of the map and the localization of the robot within this map. We describe an experiment with the use of a monocular vision system to build a vertical-edge-based environmental map for robot localization purposes and discuss its results.

1 Introduction

Many robotic tasks, such as robot self-localization, require the use of a precise description of the robot environment. For this purpose, the robot is usually provided with an *a priori* map, obtained from the architectural designs of the building, or even by hand [LDW92, LA93, HS95]. The self-localization process compares scene information obtained by some sensor or sensor combination, with the scene description given by the *a priori* map, and uses any discrepancy to correct errors inherent to dead-reckoning. This solution, although theoretically valid, proves unsatisfactory in some cases for the following reasons:

1. *Maps are incomplete.*
   A map with sufficient detail may prove to be too expensive to obtain. This may result in the map not containing scene features that are detectable with sensors and thus potentially useful for tasks such as self-localization.
2. *Maps are incorrect.*
   If *a priori* maps may lack information relevant to many tasks, in many cases they also include information that proves to be incorrect. This is very frequent in maps obtained from architectural designs, because they reflect the intention of the builders, but they may not precisely reflect the final result. If this is the case, sensor information that correctly reflects the scene features has to be discarded as spurious when compared with the incorrect map. In general this can be a big source of errors if, as a result, sensorial information is incorrectly interpreted. It is also frequent that initially correct maps become out-of-date as the scene changes due to many types of possible modifications (wall additions and eliminations, inclusion of large objects, etc.).
3. *Maps are imprecise.*
   It is difficult and costly to obtain a precise *a priori* map of the environment. In most cases, the precision with which the sensor is able to perceive the environment is higher than that of the map. This has two negative consequences: *first*, some potentially useful information may be considered spurious because the correspondence process is not able to match it with any feature of the map; *second*, even if some sensorial information is adequately matched in the map, the imprecision in the location of the corresponding features in the map will induce a systematic error in the estimation process. This makes the correspondence process discard correct sensorial information, and also causes the estimation process to be less precise than it would be possible.

Given that sensors may perceive the robot environment more accurately, the evident alternative is to have the sensor *build* the map, rendering a precise and always up-to-date description of the scene. In this work, we investigate the limitations of *a priori* maps for mobile robot self-localization, and the fundamental issues related to automatic map building using as example a monocular vision system for environment perception. In section 2 we analyze the main problems of building an environment using a monocular vision system mounted on a mobile robot. In section 3 we describe the mathematical tools that we use to represent and integrate uncertain sensor information. Section 4 contains the results we have obtained experimenting in map building using monocular vision. Finally, in section 5 we draw the main conclusions of this work.

2 Map Building using Monocular Vision

2.1 Stochastic Maps

Some robot tasks require the system to precisely determine *both* the location of environment features *and* the robot location using sensor measurements of the environment. This problem is a simplified version of the *structure and motion problem*; in this case we must determine the structure and *refine* the motion of which we have an estimation (odometry). However, we are confronted with a rather difficult problem: our estimation mechanism, the Extended Kalman Filter, requires all integrated measurements to be *statistically independent*. In this case, we wish to use a sensor measurement to estimate both robot and feature location. Thus, these two estimations will become *statistically correlated*. This has two fundamental consequences: *first*, each sensor observation must be integrated only once, and thus, the estimation mechanism must integrate the observation to estimate the robot and the feature location at the same time, and *second* information regarding these correlations must be maintained.
An environment map where correlations between all estimated features are maintained is usually referred to as a *stochastic map* [SSC88]. If the amount of features is large, maintaining correlations is a computationally expensive process in time and space[LDWC92]. From a theoretical point of view, statistical correlation will exist. Thus, although costly, it is necessary to build and maintain the full stochastic map [CTS97, HBBC96]. We make use of the SPmodel, a probabilistic model to represent and integrate uncertain geometric information that is specially well suited to build and update stochastic maps. This model has been used for map building using laser [CTS97], and in this work we show that it can also be used for map building using monocular vision, and thus also in a combination of these and other sensors. In section 3 we describe the 2D version of the SPmodel and its application to stochastic map building. Further details related to the SPmodel can be found in [Tar92].

2.2 The Correspondence Problem

Another fundamental problem in map-based localization is the *correspondence problem*. In general, this problem consists in matching sensorial information with environment model features that constitute the map. In the case of 2D monocular vision, we must determine the identity of the vertical edges observed in an image with respect to prior observations, as well as with respect to the built map. When there is an *a priori* map available, this problem is simpler, given that we always have the map as reference. But when we are building the map, specially in the initial steps, there is no reference to compare with, and this makes the correspondence problem error-prone.
The solution to this problem depends largely on the type of sensor being considered and also on the type of environment features that compose the map. Some sensors, such as laser, and environment features, such as walls, make correspondence a simpler problem to solve. In the case of laser, the mathematical tools used in determining if a laser point belongs to some wall are rather robust because indoor environments are structured enough to have very little confusion on candidate walls for a laser point.
This is not the case in the problem we study. We extract vertical edges for monocular vision images and try to establish a correspondence between the obtained projection rays and the features that constitute the map, wall corners and door frames. In this case, in indoor environments, for example corridors, there may be several corners visible and potentially compatible with a projection ray of a single image. Instead of determining each matching separately, which limits coherence to a *local* match, we can decide some matching process that tries to maximize *global* coherence. That is, decide the matchings for *all* the projection rays of an image, so that they are *globally* coherent. This may be done in an iterative process by first accepting the most reliable pairings (projection rays with only one candidate corner), integrating the pairings to reduce the robot orientation error, systematic to all projection rays of one image, and then deciding the rest of the matchings. This alternative can be implemented successfully [NHTS96], although it ignores the fact that the images are taken at small distances, and thus information related to the pairings obtained for the previous image may help decide the pairings for the next image. For this reason, in this work we *track* the vertical edges in the sequence of images that the robot takes as it moves [DF90, CSSP92], which gives additional pairing information that helps to reduce the complexity of the correspondence problem.

In the next section we describe the mathematical tools that allow us to represent the imprecise location of a mobile robot, map features and sensor observations, which are used to build and maintain a stochastic map of the robot's environment.

3 Representing and Integrating Uncertain Locations

We use a uncertainty representation and integration model, the SPmodel, in which a reference E is associated to the location of any type of geometric feature. The location of this reference with respect to a base reference is given by a transformation t_{WE} composed by two cartesian coordinates and an angle:

$$\mathbf{x}_{WE} = (x,\ y,\ \phi)^T \quad \text{where:} \quad t_{WE} = \text{Trans}(x,\ y) \cdot \text{Rot}(z,\ \phi)$$

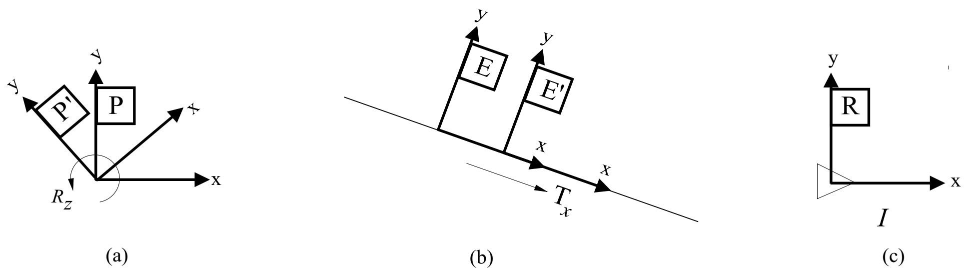
The composition of two location vectors is denoted by $\oplus$, and the composition with the inverse is abbreviated by $\ominus$. Thus, given $\mathbf{x}_{AB} = (x_1,\ y_1,\ \phi_1)^T$ and $\mathbf{x}_{BC} = (x_2,\ y_2,\ \phi_2)^T$, their composition is calculated as follows:

$$\begin{aligned} \mathbf{x}_{AC} &= \mathbf{x}_{AB} \oplus \mathbf{x}_{BC} \\ &= (x_1 + x_2 \cos\phi_1 - y_2 \sin\phi_1, y_1 + x_2 \sin\phi_1 + y_2 \cos\phi_1, \phi_1 + \phi_2)^T \end{aligned} \tag{1}$$

Similarly,

$$\ominus\mathbf{x}_{AB} = (-x_1 \cos\phi_1 - y_1 \sin\phi_1, x_1 \sin\phi_1 - y_1 \cos\phi_1, -\phi_1)^T \tag{2}$$

Different geometric features have different d.o.f. associated to their location. For example, the location of a robot in 2D is determined by three d.o.f., while the location of a point only by two. The d.o.f. that determine the location of a geometric entity are related to its *symmetries of continuous motion.* The symmetries of a geometric entity E are defined as the set $\mathcal{S}_E$ of transformations that preserve the element. For example, the symmetries of an infinite edge are the set of continuous translations (T_x) along the edge (fig. 1). We represent the set of symmetries using a *row selection matrix* B_E, denominated *binding matrix of the feature.* In fig. 1, the binding matrix for different types of geometric entities are given.



$$\text{(a) } B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix};\ \text{(b) } B = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix};\ \text{(c) } B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 1.: Representation of (a) a 2D point (map corner); (b) a 2D infinite edge (monocular projection ray); (c) a mobile robot.

Since sensors give imprecise information, it is only possible to obtain an estimate of the location of a given geometric element. The SPmodel is a probabilistic model in which the estimate of the location of a given entity E is denoted by $\hat{\mathbf{x}}_{WE}$, and the error associated to this estimate is expressed using a *differential location vector* $\mathbf{d}_E$, relative to the reference associated to the element, so that the true location of E is given by:

$$\mathbf{x}_{WE} = \hat{\mathbf{x}}_{WE} \oplus \mathbf{d}_E \tag{3}$$

Since the d.o.f. of $\mathbf{d}_E$ corresponding to the symmetries of continuous motion contain no location information, we assign 0 to their corresponding values. We call *perturbation vector* a vector $\mathbf{p}_E$ formed by the non-null elements of $\mathbf{d}_E$. These two vectors are related by the binding matrix B_E:

$$\mathbf{d}_E = B_E^T \mathbf{p}_E \quad ; \quad \mathbf{p}_E = B_E \mathbf{d}_E$$

The information associated to the estimated location of a geometric element E is represented by a quadruple $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E, B_E)$, where:

$$\mathbf{x}_{WE} = \hat{\mathbf{x}}_{WE} \oplus B_E^T \mathbf{p}_E \quad ; \quad \hat{\mathbf{p}}_E = E[\mathbf{p}_E] \quad ; \quad C_E = Cov(\mathbf{p}_E)$$

We denominate this quadruple *uncertain location vector*. Note that the error associated to a location is expressed relative to the feature reference E and not to the base reference W. In this way the value of the covariance is not magnified by the distance of the feature to the base reference. This guarantees that the covariance values have a clear interpretation. The use of the binding matrix also makes the representation non-overparameterized.

3.1 Stochastic Map Building and Updating

The estimation of the location of an object or feature from a set of geometric observations is nonlinear, due to the existence of orientation terms, and can be solved using the *extended Kalman filter* (EKF) [Gel74, Boz83]. In this case we use the 2D version of the SPmodel, along with the specialized versions of the EKF, to estimate the location of a mobile robot and a set of environmental features from a set of partial and imprecise observations of these features.

EKF state vector Let $\mathbf{L}_{WR} = (\hat{\mathbf{x}}_{WR}, \hat{\mathbf{d}}_R, C_R, I_3)$ be the estimated location of a mobile robot. Let $\mathbf{L}_{WM_i} = (\hat{\mathbf{x}}_{WM_i}, \hat{\mathbf{p}}_{M_i}, C_{M_i}, B_{M_i})$ represent the estimated location of a vertical edge in the scene (corner, door or window frame,...), where $i \in \{1, \ldots, m\}$. The state to be estimated is composed of the perturbation vectors of the robot and the set of vertical edges, which constitute the map features:

$$\mathbf{x} = [\mathbf{d}_R \; \mathbf{p}_1 \ldots \mathbf{p}_m]^T \quad ; \quad P = [P_{i,j}]_{i,j=\{0,\cdots,m\}} = \begin{bmatrix} C_R & C_{0,1} & \cdots & C_{0,m} \\ C_{1,0} & C_{1,1} & \cdots & C_{1,m} \\ \vdots & \vdots & \vdots & \\ C_{m,0} & C_{m,1} & \cdots & C_{M_m} \end{bmatrix}$$

where $P_{0,0} = C_R$ is the uncertainty in the robot location, $P_{i,i} = C_{M_i}$ is the uncertainty in the location of a feature, and $P_{i,j}$ expresses the correlation between elements i and j of $\mathbf{x}$. This scheme is usually referred to as *stochastic map* [HBBC96, CTS97].

EKF Prediction Phase Suppose that the robot moves from location R to location S, with respect to a global reference W. This motion can be represented using an uncertain location as:

$$\mathbf{L}_{RS} = (\hat{\mathbf{x}}_{RS}, \hat{\mathbf{d}}_S, C_S, I_3)$$

where $\hat{\mathbf{x}}_{RS}$ represents the new location of the robot with respect to its prior location, $\hat{\mathbf{d}}_S$ represents the estimated error in $\mathbf{x}_{RS}$, and C_S its covariance. The error associated with this motion depends on the odometric system of the robot. The predicted location of S with respect to W (the new robot location) is obtained as follows: location vector $\mathbf{x}_{WS}$ will be given by:

$$\begin{aligned} \mathbf{x}_{WS} &= \mathbf{x}_{WR} \oplus \mathbf{x}_{RS} \\ &= \hat{\mathbf{x}}_{WR} \oplus \mathbf{d}_R \oplus \hat{\mathbf{x}}_{RS} \oplus \mathbf{d}_S \\ &= (\hat{\mathbf{x}}_{WR} \oplus \hat{\mathbf{x}}_{RS}) \oplus (J_{SR}\, \mathbf{d}_R \oplus \mathbf{d}_S) \\ &\simeq (\hat{\mathbf{x}}_{WR} \oplus \hat{\mathbf{x}}_{RS}) \oplus (J_{SR}\, \mathbf{d}_R + \mathbf{d}_S) \end{aligned}$$

where J_{SR} is the Jacobian corresponding to transformation $\mathbf{x}_{SR} = (x,\ y,\ \phi)^T$:

$$J_{SR} = \begin{bmatrix} \cos\phi & -\sin\phi & y \\ \sin\phi & \cos\phi & -x \\ 0 & 0 & 1 \end{bmatrix}$$

The uncertain location of S with respect to W is given by $\mathbf{L}_{WS} = (\hat{\mathbf{x}}_{WS}, \hat{\mathbf{d}}_S^W, C_S^W, B_S)$, where:

$$\hat{\mathbf{x}}_{WS} = \hat{\mathbf{x}}_{WR} \oplus \hat{\mathbf{x}}_{RS} \quad ; \quad \mathbf{d}_S^W = J_{SR}\, \mathbf{d}_R + \mathbf{d}_s \quad ; \quad C_S^W = J_{SR} C_R J_{SR}^T + C_S$$

The predicted state of the stochastic map after the robot moves from location k of the trajectory to $k+1$ is:

$$\mathbf{x}_{(k+1)} = F\, \mathbf{x}_{(k)} + D\, \mathbf{d}_{(k+1)}$$

where:

$$F = \begin{bmatrix} J_{k+1,k} & 0_{3\times 2m} \\ 0_{2m\times 3} & I_{2m\times 2m} \end{bmatrix} \quad ; \quad D = \begin{bmatrix} I_{3\times 1} \\ 0_{2m\times 1} \end{bmatrix} \quad ; \quad P_{(k+1)} = F P_{(k)} F^T + D C_{(k+1)} D^T$$

EKF Estimation Phase Let $\mathbf{L}_{RE} = (\hat{\mathbf{x}}_{RE}, \hat{\mathbf{p}}_E, C_E, B_E)$ be a monocular 2D edge, a *partial* observation of corner $M_k = M$ according to a monocular vision system (fig. 2). Assuming that observation E corresponds to model feature M, we may use it to improve the estimation of the robot location, as well as the location of the feature itself. In this case, the state to be estimated is represented by the perturbation vectors of R and M, and the measurement by the perturbation vector of E:

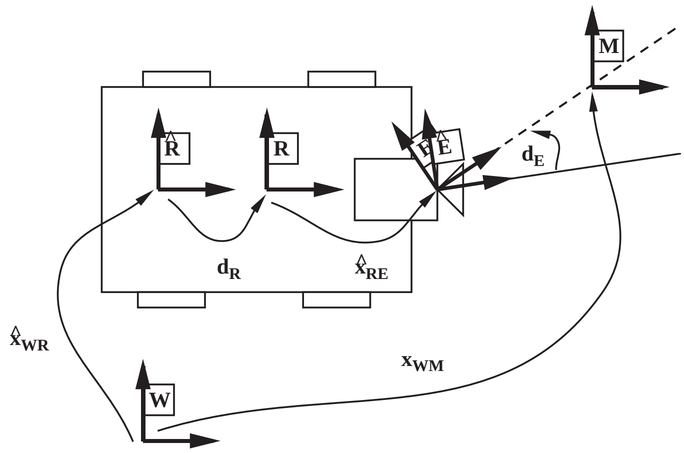


Fig. 2.: References involved in the integration of a monocular edge to the estimation of the robot location.

$$\mathbf{x} = [\mathbf{d}_R \ \mathbf{p}_M]^T \quad ; \quad \mathbf{y}_k = \mathbf{p}_E$$

The EKF measurement equation, which relates the state and the measurement, expresses the coincidence between E and M:

$$\begin{aligned}
\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) &= B_{EM}\mathbf{x}_{EM} \\
&= B_{EM}(\ominus\mathbf{x}_{RE} \ominus \mathbf{x}_{WR} \oplus \mathbf{x}_{WM}) \\
&= B_{EM}(\ominus(\hat{\mathbf{x}}_{RE} \oplus \mathbf{d}_E) \ominus (\hat{\mathbf{x}}_{WR} \oplus \mathbf{d}_R) \oplus \hat{\mathbf{x}}_{WM} \oplus \mathbf{d}_M) \\
&= B_{EM}(\ominus\mathbf{d}_E \ominus \hat{\mathbf{x}}_{RE} \ominus \mathbf{d}_R \ominus \hat{\mathbf{x}}_{WR} \oplus \hat{\mathbf{x}}_{WM} \oplus \mathbf{d}_M) \\
&= B_{EM}(\ominus\mathbf{d}_E \ominus \hat{\mathbf{x}}_{RE} \ominus \mathbf{d}_R \oplus \hat{\mathbf{x}}_{RM} \oplus \mathbf{d}_M) \\
&= B_{EM}(\ominus\mathbf{d}_E \ominus \hat{\mathbf{x}}_{RE} \oplus \hat{\mathbf{x}}_{RM} \ominus J_{MR}\mathbf{d}_R \oplus \mathbf{d}_M) \\
&= B_{EM}(\ominus\mathbf{d}_E \oplus \hat{\mathbf{x}}_{EM} \ominus J_{MR}\mathbf{d}_R \oplus \mathbf{d}_M) \\
&= B_{EM}(\ominus B_E^T\mathbf{p}_E \oplus \hat{\mathbf{x}}_{EM} \ominus J_{MR}\mathbf{d}_R \oplus B_M^T\mathbf{p}_M) \\
&= 0
\end{aligned}$$

Intuitively, this implicit function states that if E is an observation of M, then their location must coincide, up to symmetries. Thus, the components of their relative location vector $\mathbf{x}_{EM}$ not corresponding to their symmetries must be equal to zero. Since $\mathbf{f}_k$ is non-linear, we use a first order approximation of $\mathbf{f}_k$:

$$\mathbf{f}_k(\mathbf{x}, \mathbf{y}_k) \simeq \mathbf{h}_k + H_k(\mathbf{x} - \hat{\mathbf{x}}) + G_k(\mathbf{y}_k - \hat{\mathbf{y}}_k)$$

where:

$$\mathbf{h}_k = \mathbf{f}_k(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k) \ ; \ H_k = \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k)} \ ; \ G_k = \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{y}}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k)} \tag{4}$$

If we consider both estimations centered, we have:

$$\begin{aligned}
\mathbf{h}_k &= B_{EM}\hat{\mathbf{x}}_{EM} \\
H_R &= \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{d}_R}\right|_{(\hat{\mathbf{x}}, \hat{\mathbf{y}}_k)} = -B_{EM}J_{2\oplus}\{\hat{\mathbf{x}}_{EM}, 0\}J_{MR}
\end{aligned}$$

$$
\begin{aligned}
H_M &= \left.\frac{\partial \mathbf{f}_k}{\partial \mathbf{p}_M}\right|_{(\hat{\mathbf{x}},\hat{\mathbf{y}}_k)} = B_{EM} J_{2\oplus}\{\hat{\mathbf{x}}_{EM}, 0\} B_M^T \\
G_k &= -B_{EM} J_{1\oplus}\{0, \hat{\mathbf{x}}_{EM}\} B_E^T
\end{aligned} \tag{5}
$$

where $J_{1\oplus}$ and $J_{2\oplus}$ are the Jacobians of the composition of two location vectors:

$$
J_{1\oplus}\{\mathbf{x}_1, \mathbf{x}_2\} = \left.\frac{\partial(\mathbf{y} \oplus \mathbf{z})}{\partial \mathbf{y}}\right|_{\mathbf{y}=\mathbf{x}_1,\ \mathbf{z}=\mathbf{x}_2} = \begin{bmatrix} 1 & 0 & -x_2 \sin\phi_1 - y_2 \cos\phi_1 \\ 0 & 1 & x_2 \cos\phi_1 - y_2 \sin\phi_1 \\ 0 & 0 & 1 \end{bmatrix} \quad ; \quad J_{1\oplus}\{0, \mathbf{x}\} = \begin{bmatrix} 1 & 0 & -y \\ 0 & 1 & x \\ 0 & 0 & 1 \end{bmatrix}
$$

$$
J_{2\oplus}\{\mathbf{x}_1, \mathbf{x}_2\} = \left.\frac{\partial(\mathbf{y} \oplus \mathbf{z})}{\partial \mathbf{z}}\right|_{\mathbf{y}=\mathbf{x}_1,\ \mathbf{z}=\mathbf{x}_2} = \begin{bmatrix} \cos\phi_1 & -\sin\phi_1 & 0 \\ \sin\phi_1 & \cos\phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad ; \quad J_{2\oplus}\{\mathbf{x}, 0\} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

In this case we have:

$$
B_M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad ; \quad B_E = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad ; \quad B_{EM} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}
$$

The estimation equations follow the general EKF formulation[BSF88]: let $\hat{\mathbf{x}}^{(k-1)}$ and $P^{(k-1)}$ be the estimate of the state vector and its covariance after integrating the first $k-1$ observations. The estimate $\hat{\mathbf{x}}^{(k)}$ of the state vector and its covariance $P^{(k)}$ after integrating measurement $\hat{\mathbf{y}}_k$ are:

$$
\hat{\mathbf{x}}^{(k)} = \hat{\mathbf{x}}_i^{(k-1)} - k_i \mathbf{h}_k \quad ; \quad k = \{0, \cdots, m\} \quad ; \quad P_{i,j}^{(k)} = P_{i,j}^{(k-1)} - k_i (H_R\, P_{0,j}^{(k-1)} + H_M\, P_{M,j}^{(k-1)}) \tag{6}
$$

where:

$$
\begin{aligned}
k_i &= \left(P_{i,0}^{(k-1)}\, H_R^T + P_{i,M}^{(k-1)}\, H_M^T\right) R \\
R &= \left(H_R\, P_{0,0}^{(k-1)} H_R^T + H_M\, P_{M,R}^{(k-1)} H_R^T + H_M\, P_{M,M}^{(k-1)} H_M^T + H_R\, P_{R,M}^{(k-1)} H_M^T + G_k S_k G_k^T\right)^{-1}
\end{aligned}
$$

3.2 Tracking

In this subsection we describe the mathematical details of the tracking mechanism used to solve the correspondence problem. Tracking is used to estimate the motion of vertical edges in the image sequence [DF90, GM95] (e.g. the variation of location parameters and edge length). This estimation is used to predict where each edge should be located in the next image, and thus solve the correspondence problem by choosing the edge nearest to the predicted location in each case. In this case the Kalman filter is stated as follows: let $(\mathbf{x}, P)$ be the state to be estimated and its covariance; let $(\mathbf{z}, R)$ be a measurement of $\mathbf{x}$; the *plant* and *measurement* models of the system follow the equations:

$$
\begin{aligned}
\mathbf{x}_{k+1} &= F\,\mathbf{x}_k + \mathbf{u} \\
\mathbf{z} &= H\,\mathbf{x} + \mathbf{v}
\end{aligned}
$$

Thus, given the *estimation* of the system state $(\hat{\mathbf{x}}_e, P_e)$ at some instant, the *prediction* $(\hat{\mathbf{x}}_p, P_p)$ at the next instant is calculated as:

$$
\begin{aligned}
\hat{\mathbf{x}}_p &= F\,\hat{\mathbf{x}}_e \\
P_p &= F\,P_e F^T + Q
\end{aligned} \tag{7}
$$

where $Q = cov(\mathbf{u})$. Given a new measurement, the new *estimation* of the system is calculated as follows:

$$
\begin{aligned}
S &= H P_p H^T + R \\
K &= P_p H^T S^{-1} \\
\mathbf{r} &= \mathbf{z} - H\hat{\mathbf{x}}_p \\
\hat{\mathbf{x}}_e &= \hat{\mathbf{x}}_p + K\mathbf{r} \\
P_e &= (I - KH) P_p
\end{aligned} \tag{8}
$$

Vector $\mathbf{r}$ is denominated *prediction error* or *innovation*; S is its covariance. To determine whether measurement $\mathbf{z}$ can be considered compatible with the prediction, a statistical measurement, the *Mahalanobis* distance, is used:

$$d = \mathbf{r}^T S^{-1} \mathbf{r}$$

In the Gaussian case, the Mahalanobis distance follows a χ^2_n distribution, where n denotes the degrees of freedom (n is the number of parameters measured). Given a measurement $(\mathbf{z}, R)$, whose innovation for a given estimation is $(\mathbf{r}, S)$, and given an acceptance level α, the measurement can be considered compatible with the state if:

$$d \leq \chi^2_n(\alpha)$$

We consider three parameters in the identification of each edge: the coordinates (x, y) of the middle point and the edge length l. Edge orientation is not used since only vertical edges are considered. Given that we intend to determine the variation of these parameters in the image sequence, in this case the state vector of the system is:

$$\mathbf{x} = \left(x,\, \dot{x},\, y,\, \dot{y},\, l,\, \dot{l}\right)^T$$

Assuming that each parameter p is decoupled from the rest, and assuming also that its variation is constant, the state equation for each pair $(p, \dot{p})^T$ is defined in the following way:

$$\begin{aligned} p_p &= p_e + \dot{p}\Delta t \\ \dot{p}_p &= \dot{p}_e \end{aligned}$$

where Δt is the elapsed time between the acquisition of an image and the acquisition of the next. From the prediction equation (7), we conclude that:

$$F_p = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad ; \quad F = \begin{bmatrix} F_x & 0 & 0 \\ 0 & F_y & 0 \\ 0 & 0 & F_l \end{bmatrix}$$

Given that in each image we can observe each parameter p, but not its derivate $\dot{p}$, from equation (8) we have that the state and the measurement are related by:

$$H_p = \begin{bmatrix} 1 \; 0 \end{bmatrix} \quad ; \quad H = \begin{bmatrix} H_x & 0 & 0 \\ 0 & H_y & 0 \\ 0 & 0 & H_l \end{bmatrix}$$

To model the covariance of the state noise Q, we use the following discretization model [BSF88]:

$$Q_p = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & \Delta t \end{bmatrix} \dot{q}_p \quad ; \quad Q = \begin{bmatrix} Q_x & 0 & 0 \\ 0 & Q_y & 0 \\ 0 & 0 & Q_l \end{bmatrix}$$

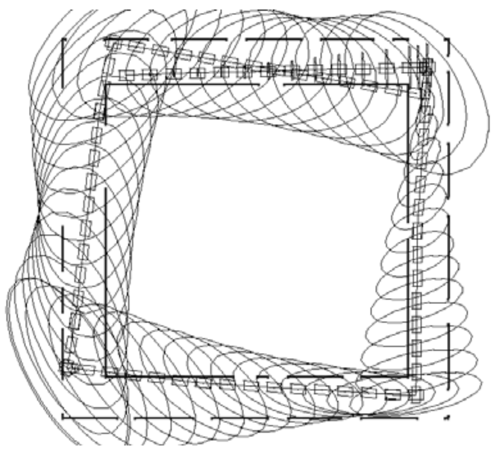


Fig. 3.: Robot trajectory according to odometry, starting in the upper left corner, and ending in the upper right corner after full traversal. Ellipsoids represent the region where the uncertainty model of the robot expects the robot to be located with a 95% confidence level.

Fig. 4.: *Top left and right:* first and last image of the sequence of images of the initial part of the trajectory; *middle:* Image location of a vertical edge (corresponding to a scene corner) tracked over the image sequence; *bottom:* estimated location of the corner after integrating all projection rays.

4 Experiments

In our experimental setup, a Labmate robot is programmed to follow a path around a square corridor. The nominal position is given by the measurements obtained from the odometric system, as shown in fig. 3, together with its associated uncertainty. An approximate *a priori* map is shown in the figure to give an idea of the scene, but it is not used during the process. It is well known that odometry errors grow unbounded, so when the robot returns to its initial position (we have programmed it to visit the initial section of the square path twice), error has accumulated so much that the measurement is useless.

At each step of the robot trajectory, a monocular image of the scene is obtained, and vertical edges, corresponding to scene corners and door frames, are extracted using Burns' procedure[BHR86]. A stochastic 2D map of the scene, consisting of 2D points, is built and updated using these observations. Correspondences between vertical edges in the image are established by tracking them as described above. Fig. 4, left, show the results of tracking one edge over the first ten images of the trajectory.

The sets of corresponding edges obtained by the tracking process are used to build and update the stochastic map. The first two projection rays are used to analytically determine the map feature location, and subsequent projection rays are used to refine it. The resulting estimation of the feature location for the set of edges is shown in fig. 4, right. Repeating this process for all the observations obtained by the monocular vision system along the trajectory, the resulting estimation of the robot location at each point of the trajectory, as well as its uncertainty, is obtained (fig.

5).

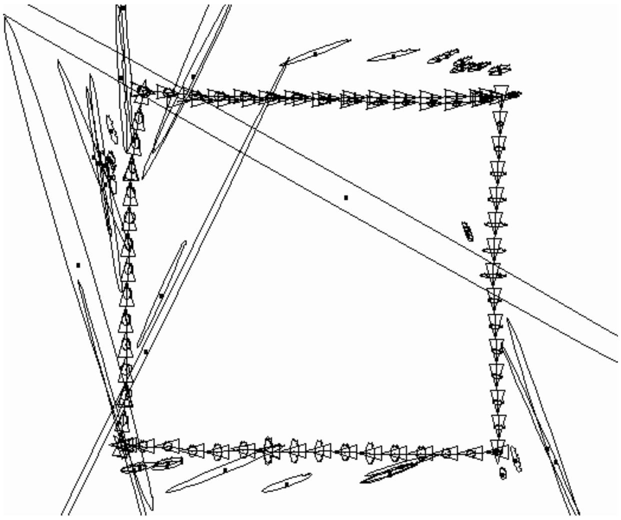


Fig. 5.: Estimated robot trajectory and stochastic map after building and updating a vertical-edges-based map. Map features with high uncertainty have been observed few times and/or from nearby locations.

In general, tracking is possible during the straight part of the trajectory. But when the robot makes a right turn, none of the features previously observed are visible any more. If turning motions are carried out in a smoother way, this problem will disappear. The system obtains a more precise estimation of a feature if it has seen it several times from significantly different locations (map features with high uncertainty are seen only twice and from nearby locations). Nevertheless, the system is capable of making use of the available information to produce an estimated trajectory highly coherent with the environment geometry. The fundalmental reason for this is that the main source of errors in odometric systems are orientation errors. Orientation errors in one point of the trajectory become position errors in subsequent points (position and orientation become coupled). The information that can be obtained with a monocular vision system is angular, and thus it allows to limit the effect of these orientation errors.

5 Conclusions

In this work we discuss the most relevant aspects of the map building process for mobile robot navigation. We describe an experiment where simultaneous localization and map building using a monocular vision system is carried out. Results obtained are highly satisfactory for localization purposes: the quality of the obtained map is low for other purposes such as path planning, but the camera acts as a simple and precise robot localization instrument. Improvements could be obtained using a wide angle lens that would allow to observe more features from a given location. Additional constraints, such as visibility (the estimated location of the vertical edges should always be within the range covered by the camera) may also improve the correspondence process. The location of features in the stochastic map far from the global reference will always be less precise than the location of those near the global reference. For localization purposes, we can build the map relative to the robot reference instead of relative to a global reference.

Building a simple vertical-edge-based map is sufficient for localization purposes. However, if a more detailed and complete map is needed for other tasks, the stochastic map should be built using a combination of sensors, such as monocular or stereo vision and laser. This is possible using the SPmodel, and it would make the system more robust and precise. This will be the subject of future work.

Acknowledgments

This work has been financed by the HCM/ERNET programme, contract No. ERBCHRXCT 930381, of the European Commission. The authors wish to thank José Santos-Victor, of ISR, for collaborating in obtaining experimental data, and Luis Mario Cascante, of CPS, for his help in programming.

References

[BHR86] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(4):425–455, 1986.

[Boz83] C.A. Bozzo. *Le Filtrage Optimal et ses Applications aux Problèmes de Poursuite*. Technique et Documentation, Montrouge, France, 1983.

[BSF88] T. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press In., 1988.

[CSSP92] J.L. Crowley, P. Stelmaszyk, T. Skordas, and P. Puget. Measurement and integration of 3-D structures by tracking edge lines. *International Journal of Computer Vision*, 8(1):29–59, 1992.

[CTS97] J. Castellanos, J.D. Tardós, and G. Schmidt. Building a global map of the environment of a mobile robot: The importance of correlations. In *IEEE Intl. Conference on Robotics and Automation (to appear)*, Alburqueque, New Mexico, 1997.

[DF90] R. Deriche and O. Faugeras. Tracking line segments. In *First European Conference on Computer Vision*, pages 259–268, Antibes, France, 1990.

[Gel74] A. Gelb, editor. *Applied Optimal Estimation*. The M.I.T. Press, Cambridge, Mass., 1974.

[GM95] J.J. Guerrero and J.M. Martínez. Determination of corresponding segments by tracking both geometrical and brightness information. In *International Conference on Advanced Robotics*, pages 703–709, Barcelona, September 1995.

[HBBC96] P. Herbért, S. Betgé-Brezetz, and R. Chatila. Probabilistic map learning: Necessity and difficulties. In Leo Dorst, Michiel van Lambalgen, and Frans Voorbraak, editors, *Reasoning with Uncertainty in Robotics*, pages 307–321. Springer, 1996.

[HS95] J. Horn and G. Schmidt. Continuous Localization of a Mobile Robot Based on 3D–Laser–Range–Data, Predicted Sensor Images, and Dead–Reckoning. *Robotics and Autonomous Systems, Special Issue "Research on Autonomous Mobile Robots"*, 14:99–118, 1995.

[LA93] X. Lebègue and J.K. Aggarwal. Significant line segments for an indoor mobile robot. *IEEE Trans. on Robotics and Automation*, 9(6):801–815, 1993.

[LDW92] J.J. Leonard and H.F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers, London, 1992.

[LDWC92] J.J. Leonard, H.F. Durrant-Whyte, and I.J. Cox. Dynamic map building for an autonomous mobile robot. *IEEE Journal of Robotics Research*, 11(4), 1992.

[NHTS96] J. Neira, J. Horn, J.D. Tardós, and G. Schmidt. Multisensor mobile robot localization. In *IEEE International Conference on Robotics and Automation*, Minneapolis, USA, May 1996.

[SSC88] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In J.F. Lemmer and L.N. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 435–461. Elsevier Science Pub., 1988.

[Tar92] J.D. Tardós. Representing partial and uncertain sensorial information using the theory of symmetries. In *IEEE International Conference on Robotics and Automation*, pages 1799–1804, Nice, France, May 1992.

This article was processed using the TEX macro package with SIRS97 style