CONSTRAINT-BASED MOBILE ROBOT LOCALIZATION *

José A. Castellanos   Juan D. Tardós   José Neira

Dpto. de Informática e Ingeniería de Sistemas
Centro Politécnico Superior, Universidad de Zaragoza
María de Luna 3, E-50015 Zaragoza, SPAIN
Email: joseac@prometeo.cps.unizar.es

Abstract. *Mobile Robot Localization is a fundamental problem to solve when navigating in an indoor structured environment. This problem might be stated as a matching problem between sensor observations and model features of an a priori map of the environment. Geometric constraints may be used to reduce the complexity of the matching process. Appropiate modelling of the geometric information is required to deal with such constraints. In this paper we estimate the mobile robot localization by means of a matching between segments obtained by a laser rangefinder mounted on the robot and model segments of an a priori map of the environment. A probabilistic method is used to represent the uncertainty and partiallity of the geometric information involved. We give some experimental results in which we compare two alternative matching schemes.*

Keywords: *Mobile Robot Localization, Geometric Constraints, Matching Algorithms, Probabilistic Methods*

1 Introduction

Mobile robot localization is an important problem for navigation in an indoor structured environment. It has given rise to a great number of solutions using different types of external sensors mounted on the mobile robot. We focus on feature-based methods, in which a set of features are extracted from the sensed data (such as line segments, corners, etc.) and then matched with the corresponding features in a model. In general, matching problems are of exponential complexity. In this case, reduction of this complexity can be achieved by application of two fundamental ideas [6]: the use of *validation mechanisms* that allow the system to discard entire subspaces of the solution space from further consideration, and the use of *strategies for the generation and verification of hypothesis*, that can help the system in searching the solution space more efficiently to obtain more plausible hypothesis promptly. The mobile robot localization problem implies carrying out two tasks: determining the observation-model pairings (i.e. identification), and computing its location in the environment. Identification is a search problem, while computing robot location is an estimation problem. This twofold goal has given rise to two fundamental matching schemes: The *Identifying Before Locating* scheme [5], based on separating the processes of pairings identification, and of determining the robot location in the environment, and the *Identifying While Locating* scheme [4], in which identification and localization are carried out simultaneously.

Most of the previous works have considered the use of the identifying before locating scheme. Thus, Drumheller, [3] estimates the localization of a mobile robot by application of this scheme to match the observations obtained by a sonar rangefinder and the model features of an priori map of the room. To reduce the complexity of the process he uses local constraints. In [9] Talluri et al. present a technique for estimating the location of a mobile robot in an structured outdoor environment, consisting in polyhedral buildings. They establish a correspondence between the lines that constitute the rooftops of the buildings and their images obtained by a CCD camera.

In this paper we compare the two alternative matching schemes, in order to decide which is the most appropiate for the problem of mobile robot localization. We use a segment-based method combined with a probabilistic model to represent the uncertainty and partiallity of the geometric information involved. We use geometric constraints to reduce the exponential complexity of the mobile robot localization, that is, the validation that geometric relations between model features are satisfied in the observations we are trying to match with them.

In section 2.1 we briefly describe the uncertainty representation model we use. Section 2.2 describes the geometric constraints used througout the work. Complete algorithms of the two matching schemes are also provided in section 3.1 and 3.2. Finally, we present some experimental results comparing the performance of both algorithms.

2 Uncertain Geometric Constraints

Geometric constraints are greatly influenced by the uncertainty model used. In this section we present a probabilistic model to represent uncertain geometric information, which is based both on the theory of symme-

*This work was partially supported by CICYT, project TAP94-0390 and by a research grant FPI PN94-29099535.

tries and probability theory. We also describe the geometric constraints used to prune the interpretation tree in the search-for-pairings process.

2.1 Symmetries and Perturbations Model

Sensors obtain uncertain geometric information from the environment of the mobile robot. There are two fundamental aspects of geometric uncertainty:

- *Partiallity*, which refers to the degrees of freedom associated to different geometric entities, and how they determine the location of other entities related to them.

- *Imprecision*, which refers to the accuracy in the estimation of the location of geometric entities.

The Symmetries and Perturbations Model (SPmodel) [10] combines the use of probability theory to represent the imprecision in the location of a geometric element, and the theory of symmetries to represent the partiallity due to characteristics of each type of geometric element. A reference E is associated to every geometric element $\mathcal{E}$. Its location is given by a *location vector* $\mathbf{x}_{WE} = (x,\ y,\ \phi)^T$, respect to a base reference, W, composed of two Cartesian coordinates and an angle (considering 2D). The estimation of the location of an element is denoted by $\hat{\mathbf{x}}_{WE}$, and the estimation error is represented locally by a *differential location vector* $\mathbf{d}_E$ relative to the reference attached to the element. Thus, the true location of the element is:

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

where $\oplus$ represents the composition of location vectors (the inversion is represented with $\ominus$).

To account for the symmetries of the geometric element, we assign in $\mathbf{d}_E$ a null value to the degrees of freedom corresponding to them, because they do not represent an effective location error. We call *perturbation vector* the vector $\mathbf{p}_E$ formed by the non null elements of $\mathbf{d}_E$. Both vectors can be related by a row selection matrix B_E that we call *self-binding matrix* of the geometric element:

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

Based on these ideas, the SPmodel represents the information about the location of a geometric element $\mathcal{E}$ by a quadruple $\mathbf{L}_{WE} = (\hat{\mathbf{x}}_{WE}, \hat{\mathbf{p}}_E, C_E, B_E)$, where:

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

Transformation $\hat{\mathbf{x}}_{WE}$ is an estimation taken as base for perturbations, $\hat{\mathbf{p}}_E$ is the estimated value of the perturbation vector, and C_E its covariance. When $\hat{\mathbf{p}}_E = 0$, we say that the estimation is *centered*.

2.1.1 Uncertain Location of a Laser Point

The most elementary geometric feature we deal with in this work is a *laser point* which is obtained directly by a laser rangefinder. A reference P_k is attached to each laser point with the X-axis aligned with the laser beam (figure 1). Thus, a laser point is represented by an uncertain location, $\mathbf{L}_{LP_k} = (\hat{\mathbf{x}}_{LP_k}, \hat{\mathbf{p}}_{P_k}, C_{P_k}, B_{P_k})$, with respect to the laser rangefinder, L, where:

$$\begin{aligned} \hat{\mathbf{x}}_{LP_k} &= (\rho_k\ cos\phi_k, \rho_k\ sin\phi_k, \phi_k)^T \\ \hat{\mathbf{p}}_{P_k} &= (\hat{d}_x,\ \hat{d}_y)^T \end{aligned}$$

$$C_{P_k} = \begin{pmatrix} \sigma_\rho^2 & 0 \\ 0 & \rho_k^2 \sigma_\phi^2 \end{pmatrix} \ ; \ B_{P_k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

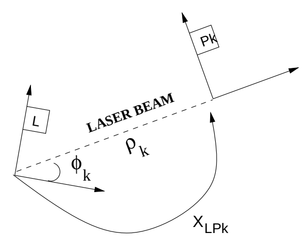

Figure 1: Uncertain Location of a Laser Point.

2.1.2 Uncertain Location of a Laser Segment

Laser segments are obtained by application of the segmentation method presented in [2] where segmentation is achieved by dividing the laser data in regions formed by a unique polygonal line, which are subsequently divided into segments by an iterative method.

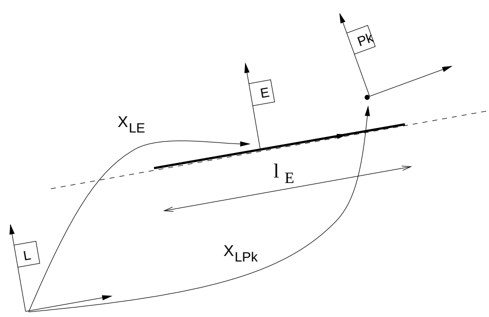

Figure 2: Uncertain Location of the edge associated with a Laser Segment.

We attach a reference E to each of the supporting edges of the segments, placed in the middle point of the segment and with the X-axis aligned with the edge (figure 2). Thus, a laser segment is represented by an uncertain location:

$$\mathbf{L}_{LE} = (\hat{\mathbf{x}}_{LE}, \hat{\mathbf{p}}_E, C_E, B_E)$$

and its observed length:

$$\mathbf{l}_E = \{\hat{l}_E, \sigma^2_{l_E}\}$$

where:

$$\begin{aligned} \hat{\mathbf{x}}_{LE} &= (\hat{x}_{LE}, \hat{y}_{LE}, \hat{\phi}_{LE})^T \\ \hat{\mathbf{p}}_E &= (\hat{d}_y,\ \hat{d}_\phi)^T \end{aligned}$$

$$C_E = \begin{pmatrix} \sigma_y^2 & 0 \\ 0 & \sigma_\phi^2 \end{pmatrix} \ ;\ B_E = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.2 Geometric Constraints

One of the fundamental ideas to reduce the exponential complexity of the mobile robot localization, stated as a matching problem, is the use of geometric constraints: the validation that geometric relations between model features are satisfied in the observations we are trying to match with them. Geometric constraints are a set of parameters that derive from the geometry of each feature, and from the relative location between features [6]. We can classify geometric constraints into two categories:

- *Location Independent Constraints*, which can be validated without having an estimation of the location of the robot. They include *unary* constraints, and *binary* constraints.
- *Location Dependent Constraints*, based on the availability of the robot location. The fundamental constraint of this type is *rigidity*: the estimation of the robot location in the environment determines the location of the model features with respect to the robot. We also present the *extension* constraint which considers the real dimensions of the involved geometric entities.

2.2.1 Location Independent Constraints

In this section we study *unary* geometric relations: which depend on a single geometric feature, such as lenght; and *binary* geometric relations: which depend on the relative location (distances and angles) between geometric entities.

Unary Constraints. Unary Constraints refer to those that apply to a single pairing of a data feature and a model feature. Such constraints involve geometric measurements such as the length of an edge, or the angle of a corner, or the area of a surface patch, etc. Unary constraints could also involve other sensory measurements, for example, the color associated with a feature, or the texture of a feature, or surface reflectance properties of a feature. Clearly, each such constraint can reduce the size of the search space.

Let a given segment E, be represented by an uncertain location $\mathbf{L}_{RE} = (\hat{\mathbf{x}}_{RE}, \hat{\mathbf{p}}_E, C_E, B_E)$, respect to the robot

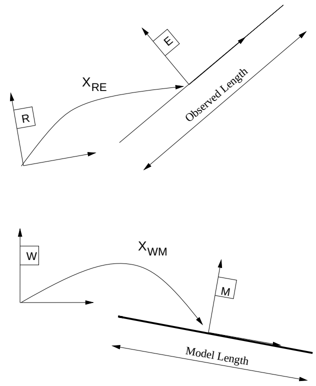

Figure 3: Validating Unary Constraint between an observed laser segment, E, and a model segment, M.

R, and by its observed length, $\mathbf{L}_E = \{\hat{l}_E, \sigma^2_{l_E}\}$. Let M be the model segment paired with E, which length is given by l_M. The length constraint is satisfied when $\hat{l}_E < l_M$, because usually we obtain only partial observations of the model features. Otherwise, a hypothesis test based on the χ^2 distribution is applied to decide if they are compatible using the Mahalanobis distance [1] calculated as:

$$D^2 = \frac{(\hat{l}_E\ -\ l_M)^2}{\sigma^2_{l_E}}$$

Under the gaussianity hypothesis, D^2 follows a chi-square distribution. For a given significance level, α, the unary constraint is satisfied if:

$$D^2 \leq D^2_{m,\alpha}$$

where $D^2_{m,\alpha}$ is a threshold value, obtained from the χ^2_m distribution, such that the probability of rejecting a good matching is α with $m = 1$ degree of freedom.

Binary Constraints. Binary Constraints refer to those that apply to two pairings of data features and model features. In general, binary geometric relations are nonlinear functions of the relative location of the involved geometric features. For this reason, *aligning transformations*, which belong to the set of symmetries of the geometric entities involved, are applied to the original uncertain locations of the features to obtained new uncertain locations which allow to estimate binary relations as linear functions of their relative location vector [8].

Let A and B represent attached references to two observed edges, which estimated location vectors are given by $\hat{\mathbf{x}}_{RA}$ and $\hat{\mathbf{x}}_{RB}$ with respect to the robot. Then, references $\bar{A}$ and $\bar{B}$ (figure 4)are characterized by the estimated location vectors:

$$\hat{\mathbf{x}}_{R\bar{A}} = \hat{\mathbf{x}}_{RA} \oplus \mathbf{x}_{A\bar{A}}\ \ ;\ \hat{\mathbf{x}}_{R\bar{B}} = \hat{\mathbf{x}}_{RB} \oplus \mathbf{x}_{B\bar{B}}$$

where the aligning transformations are:

$$\mathbf{x}_{A\bar{A}} = (x_a,\ 0,\ 0)^T\ \ ;\ \mathbf{x}_{B\bar{B}} = (x_b,\ 0,\ 0)^T$$

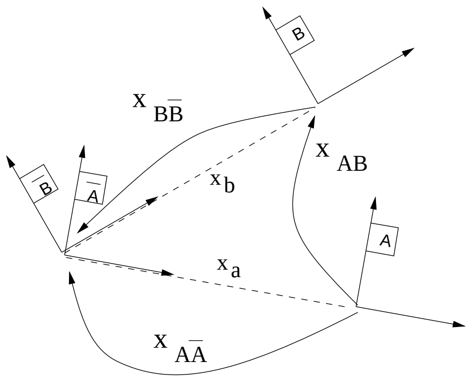

Figure 4: Reference alignment for two edges.

with x_a and x_b depending on the relative orientation of the features. Let $\hat{\mathbf{x}}_{AB} = (\hat{x}_{AB},\ \hat{y}_{AB},\ \hat{\phi}_{AB})^T$ represents the estimated relative location between features A and B, and let $\hat{\mathbf{x}}_{\bar{A}\bar{B}} = (\hat{x}_{\bar{A}\bar{B}},\ \hat{y}_{\bar{A}\bar{B}},\ \hat{\phi}_{\bar{A}\bar{B}})^T$ represents the estimated relative location between features $\bar{A}$ and $\bar{B}$, then there are two different cases:

1. When the observed features are parallel, that is $\hat{\phi}_{\bar{A}\bar{B}} \simeq 0$ or $\hat{\phi}_{\bar{A}\bar{B}} \simeq \pi$, we have:

$$x_a = 0\ ;\ x_b = -\hat{x}_{AB}$$

$$\hat{\mathbf{x}}_{\bar{A}\bar{B}} = (0, \hat{y}_{\bar{A}\bar{B}},\ 0)^T$$

Then we calculate the Mahalanobis distance by:

$$D^2 = \frac{(\hat{y}_{\bar{A}\bar{B}}\ -\ y_{M_A M_B})^2}{\sigma^2_{y_{\bar{A}\bar{B}}}}$$

2. Otherwise, we have:

$$x_a = \hat{x}_{AB} - \frac{\hat{y}_{AB}}{tan\hat{\phi}_{AB}}\ ;\ x_b = -\frac{\hat{y}_{AB}}{sin\hat{\phi}_{AB}}$$

$$\hat{\mathbf{x}}_{\bar{A}\bar{B}} = (0,\ 0,\ \hat{\phi}_{\bar{A}\bar{B}})^T$$

and the Mahalanobis distance is calculated as:

$$D^2 = \frac{(\hat{\phi}_{\bar{A}\bar{B}}\ -\ \phi_{M_A M_B})^2}{\sigma^2_{\phi_{\bar{A}\bar{B}}}}$$

For a given significance level, α, the binary constraint is satisfied if:

$$D^2 \leq D^2_{m,\alpha}$$

with $m = 1$ degree of freedom.

2.2.2 Location Dependent Constraints

The availability of an estimation of the location of the object gives us the possibility of applying other validation mechanisms on the observations. In this section we present the *rigidity* constraint and the *extension* constraint.

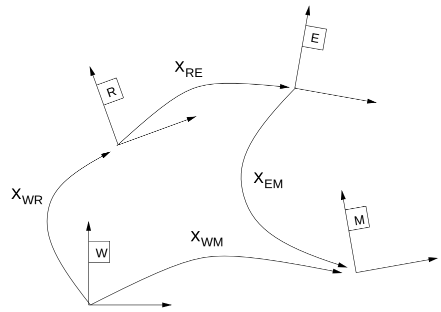

Figure 5: Validating Rigidity Constraint between an observed laser segment, E, and a model segment, M.

Rigidity Constraint. The fundamental location dependent constraint is denominated rigidity. Intuitively, rigidity states that the location of the robot in the environment determines the location of the model features respect to the robot.

Given an estimation $\mathbf{L}_{WR} = (\hat{\mathbf{x}}_{WR}, \hat{\mathbf{d}}_R, C_R, B_R)$ of the location of the robot, and given the location of the feature in the model, $\mathbf{x}_{WM}$, we can estimate the location of the feature with respect to the robot as follows:

$$\mathbf{L}_{RM} = \ominus \mathbf{L}_{WR} \oplus \mathbf{x}_{WM}$$

$$\mathbf{x}_{RM} = \ominus \mathbf{d}_R \ominus \hat{\mathbf{x}}_{WR} \oplus \mathbf{x}_{WM}$$

thus [1],

$$\begin{aligned} \hat{\mathbf{x}}_{RM} &= \ominus \hat{\mathbf{x}}_{WR} \oplus \mathbf{x}_{WM} \\ \hat{\mathbf{p}}^R_M &= -B_M J_{MR} \mathbf{d}_R \\ C^R_M &= (B_M J_{MR}) C_R (B_M J_{MR})^T \end{aligned}$$

An observed edge E can be considered compatible with a model edge M if their relative angle $\hat{\phi}_{EM}$ and their perpendicular distance $\hat{y}_{EM}$ are equal to zero. These conditions are expressed by:

$$B_E \mathbf{x}_{EM} = 0$$

where $\mathbf{x}_{EM}$ is the relative location between the edges. We can measure the discrepancy between the model edge and the observed edge using the Mahalanobis distance:

$$D^2 = (B_E \hat{\mathbf{x}}_{EM})^T (B_E Cov(\mathbf{x}_{EM}) B_E^T)^{-1} (B_E \hat{\mathbf{x}}_{EM})$$

For a given significance level, α, the rigidity constraint is satisfied if:

$$D^2 \leq D^2_{m,\alpha}$$

with $m = dim(\mathbf{p}_E) = rank(B_E)$ degrees of freedom. The estimated value of $\mathbf{x}_{EM}$ and its covariance (figure 5) can

[1] The expressions for transforming differential locations between references are: $\mathbf{d}_A \oplus \mathbf{x}_{AB} = \mathbf{x}_{AB} \oplus \mathbf{d}_B$, with $\mathbf{d}_B = J^{-1}_{AB} \mathbf{d}_A = J_{BA} \mathbf{d}_A$.

be obtained by:

$$\begin{aligned}
\hat{\mathbf{x}}_{EM} &= \ominus\hat{\mathbf{x}}_{RE} \oplus \mathbf{x}_{RM} \\
Cov(\mathbf{x}_{EM}) &= J_{1\oplus}\{0, \hat{\mathbf{x}}_{EM}\} B_E^T C_E B_E J_{1\oplus}^T \{0, \hat{\mathbf{x}}_{EM}\} \\
&\quad + J_{2\oplus}\{\hat{\mathbf{x}}_{EM}, 0\} B_M^T C_M^R B_M J_{2\oplus}^T \{\hat{\mathbf{x}}_{EM}, 0\}
\end{aligned}$$

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

$$J_{1\oplus}\{\mathbf{x}_1, \mathbf{x}_2\} = \left.\frac{\partial(\mathbf{y}\oplus\mathbf{z})}{\partial \mathbf{y}}\right|_{\mathbf{y}=\mathbf{x}_1,\ \mathbf{z}=\mathbf{x}_2}$$

$$J_{2\oplus}\{\mathbf{x}_1, \mathbf{x}_2\} = \left.\frac{\partial(\mathbf{y}\oplus\mathbf{z})}{\partial \mathbf{z}}\right|_{\mathbf{y}=\mathbf{x}_1,\ \mathbf{z}=\mathbf{x}_2}$$

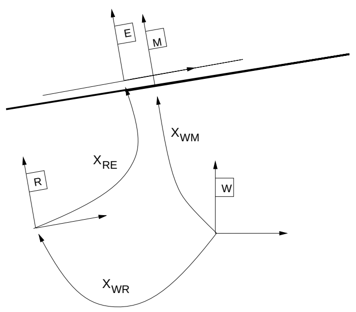

Figure 6: Validating Extension Constraint between an observed laser segment, E, and a model segment, M.

Extension Constraint. The rigidity constraint does not validate whether the observed feature is actually located within the region occupied by the model feature.

The validation that the observed feature is actually contained in the region occupied by the corresponding model feature is denominated the *extension constraint*. Unlike the other constraint validation methods, the extension constraint is particular for each type of geometric element. Considering laser segments the extension constraint consists in determining whether the most extreme points of the observed segment are contained within the extension occupied by its corresponding model segment. Let $\hat{\mathbf{x}}_{ME} = (\hat{x}_{ME}, \hat{y}_{ME}, \hat{\phi}_{ME})^T$ represent the estimated relative location between the model segment, M, and the observed segment, E. Let $\mathbf{l}_E = \{\hat{l}_E, \sigma^2_{l_E}\}$ represents the estimated length of the segment E, and let l_M be the length of the model segment. Then, E and M can be considered compatible if the endpoints of the segments are located, respect to the reference M, at a distance lower than the semilength of the model edge, that is:

$$\hat{x}_{ME} + \frac{\hat{l}_E}{2} < \frac{l_M}{2}$$

and

$$\hat{x}_{ME} - \frac{\hat{l}_E}{2} < \frac{l_M}{2}$$

Otherwise, we apply a hypothesis test:

$$D_a^2 = \frac{(\hat{x}_{ME} + \frac{\hat{l}_E}{2} - \frac{l_M}{2})^2}{\sigma^2_{x_{ME}} + \sigma^2_{l_E}} \leq D^2_{m,\alpha}$$

and

$$D_b^2 = \frac{(\hat{x}_{ME} - \frac{\hat{l}_E}{2} - \frac{l_M}{2})^2}{\sigma^2_{x_{ME}} + \sigma^2_{l_E}} \leq D^2_{m,\alpha}$$

with $m = 1$ degree of freedom.

3 Mobile Robot Self-Localization as a Matching Problem

Determining the location of a robot is an important problem for an autonomous vehicle navigating in a structured indoor environment. Mobile robot localization can be stated as a matching problem in which sensed features are obtained by the mobile robot sensors from its surrounding environment and matched with model features stored in a database which represents the structure of this environment. Two different matching schemes can be used to match the set of observations with the set of model features: the *Identifying Before Locating* scheme [5], and the *Identifying While Locating* scheme [4]. Both schemes use the geometric constraints to reduce the complexity of the matching process. There are some basic aspects of the matching problem that are common to both schemes, and some others are substantially different. The goal of the matching process is to generate an interpretation which relates each observation e_j with a model feature m_k by means of a pairing $p_i = (e_j, m_k)$. An interpretation is a set of robot-location hypotheses $\mathcal{H} = \{h_1, \ldots, h_h\}$ where each hypothesis has the form:

$$h_i = \{\mathbf{L}_{R_{h_i}}, \mathcal{S}_{h_i}\}$$

where $\mathbf{L}_{R_{h_i}}$ is the hypothesized robot location, and the set $\mathcal{S}_{h_i}$ is the set of pairings which support the hypothesis.

3.1 Identifying Before Locating

The *identifying before locating* scheme [5] is based on separating the processes of pairing identification, and of determining the robot location in the environment. In algorithm 3.1, we give a basic and simple implementation of this approach.

3.1.1 Searching for Pairings

This scheme uses very simple and fast validation mechanism to determine whether a given hypothesis is consistent with the set of observations. Such validations can be made by the geometric constraints given previously. The hypothesis generation process is based on traversing the interpretation tree in search for consistent interpretations. In algorithm 3.1, this process is written as a recursive procedure in which, at each step of the recursion, all consistent pairings between an observation e and the model features in $\mathcal{M}$ are obtained. It is important to highlight that the number of binary constraints to verify for a given hypothesis grows polinomially ($O(n^2)$)with the number of paired observations, what may lead to a great amount of computation. The `select_observation` function selects the most suitable observation because it is the one that generates as few pairings as possible.

3.1.2 Locating and Validating

Applying location independent constraints we only assure local consistency. Therefore, to assure global consistency it is necessary to estimate the robot location in order to determine whether the location of each observation and that of its corresponding model feature coincide, taking into account the imprecision in the localization of the observations. Robot localization is usually carried out using some estimation method that finds a transformation such that the error between each transformed model feature and its corresponding observed feature is minimal in some sense [7]. Once we know the robot location we apply location dependent constraints, that is, rigidity and extension constraints.

3.2 Identifying While Locating

The fundamental idea behind the *identifying while locating* scheme [4] is that an estimation of the location of the robot is a very important source of information for the identification process. Thus, the complexity of the recognition process can be reduced if identification and localization are performed simultaneously. In algorithm 3.2, we give a basic and simple implementation of this approach.

3.2.1 Hypothesis Generation

This process deals with the selection of the smallest set of observations that allow to estimate the location of the robot. As presented in algorithm 3.2 we choose two independent observations with the lowest number of possible pairings, so that the number of alternative hypothesis be small.

3.2.2 Hypothesis Verification

Verification of hypothesis is carried out in a data-driven fashion (algorithm 3.2). An observation is selected, and it is determined whether it can be paired with one of the model features. If an acceptable pairing is found, the function recurs with a robot location estimation refined by the new pairing. Note that only one validation of the rigidity and extension constraints is necessary for each potential pairing. Thus, the number of validations for a given hypothesis grows linearly with the number of pairings.

4 Experimental Results

Considering the localization of a mobile robot in an indoor structured environment, we present an example of the application of the previously described matching schemes. Laser segments are obtained by application of the segmentation algorithm presented in [2].

			Unsorted	*Sorted*
Unary	*False*	*Computed*	41	48
		Table	2256	169
	True	*Computed*	89	90
		Table	2036	1091
Binary	*False*	*Computed*	144	167
		Table	1777	970
	True	*Computed*	97	71
		Table	1111	224
Rigidity	*False*	*Computed*	50	1
	True	*Computed*	191	52
Extension	*False*	*Computed*	48	22
	True	*Computed*	143	30

Table 1: Geometric Constraints Validated by the Identifying Before Locating scheme. The example considers sixteen model features and nine observed features.

4.1 Implementation Details

We have considered two alternative strategies to select the observations: first we have selected observations in the order given by the sensor then we have sorted out the observations using segment length as the sorting criteria, thus, choosing in first place the larger segments. To reduce computation time we have adopted two solutions:

1. Results from the validation of location independent constraints are stored in a table, thus we only go once through the computation of the constraints. When a value previously calculated is required we obtain it from the table. In tables 1 and 2 we show the number of constraints which have been calculated and those which have been obtained from the tables.

2. Recursion is stopped when at least 60 percent of the available observations have been paired with model features. Thus, a complete traversing of the interpretation-tree is not required.

4.2 Algorithms Performance

Tables 1 and 2 present the number of geometric constraints validated by each scheme. Note that, in this case, sorting observations in the identifying while locating scheme does not improve performance; this is due to the fact that using the first two observations provided by the sensor, the system is able to estimate a robot location and then it applies location dependent constraints, which

			Unsorted	*Sorted*
Unary	*False*	*Computed*	30	48
		Table	0	50
	True	*Computed*	56	90
		Table	0	234
Binary	*False*	*Computed*	2	81
		Table	0	153
	True	*Computed*	1	27
		Table	0	29
Rigidity	*False*	*Computed*	48	54
	True	*Computed*	6	48
Extension	*False*	*Computed*	1	22
	True	*Computed*	5	26

Table 2: Geometric Constraints Validated by the Identifying While Locating scheme. The example considers sixteen model features and nine observed features.

are tighter. Previous works have dealt with the application of location independent constraints (i.e. unary and binary) to the reduction of the complexity of the matching problem. We have observed that using only location independent constraints, the identifying before locating algoritm obtains false hypotheses, that is, they satisfy local consistency but do not satisfy global consistency. In figure 7 we show an example of false hypothesis. The number of different supporting sets, for the example considered, which satisfy local consistency are 98 in the case of unsorted observations and 23 in the sorted case. Note that their correspondent localization might coincide in some cases. Applying local dependent constraints (i.e. rigidity and extension) the number of supporting sets is reduced to 1 and 5 respectively. Global consistent hypotheses might have an incorrect estimation of the robot location due to the fact that we stop the recursive process when at least 60 percent of the observations have been paired. These hypotheses can be eliminated by exploring further the interpretation tree, in order to validated consistency by pairing the observations which have not yet been used. The real robot location obtained is given in figure 8 where we explicitly present the supporting set of the hypothesized location. As tables 1 and 2 show, an estimation of the robot location constitutes a very important source of information for the following reasons:

1. There is a reduction in the number of constraints to verify.

2. Location dependent constraints are tighter, they assure global consistency and not just local consistency.

5 Conclusions

The problem of mobile robot localization has been stated as a matching problem, in which observations have been obtained by an exteroceptive sensor mounted on the mobile robot and model features had been stored in a database. In a general sense, matching problems have exponential complexity, therefore a mechanism to reduce this handicap is necessary. We have used geometric constraints, both location independent and location dependent, to prune the solution-space. We have compared two different matching schemes using the number of geometric constraints validated by them. As experimental results show, an estimation of the robot location constitutes a very important source of information during the matching process. Future work will consider larger indoor structured environment, which require new constraints, such as the visibility constraint, that only considers matching of the model features which are visible from the estimated robot location. This estimated location might be obtained by the dead-reckoning system of the mobile robot.

References

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

[2] J.A. Castellanos and J.D. Tardós. Laser-based segmentation and localization for a mobile robot. In F. Pin, M. Jamshidi, and P. Dauchez, editors, *Robotics and Manufacturing vol. 6 - Procs. of the 6th Int. Symposium (ISRAM)*, pages 101–108. ASME Press, New York, NY, 1996.

[3] M. Drumheller. Mobile robot localization using sonar. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(2):325–332, 1987.

[4] O. D. Faugeras and M. Hebert. The representation recognition, and locating of 3D objects. *IEEE Journal of Robotics Research*, 5(3):27–52, 1986.

[5] W.E.L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* The MIT Press, Massachusetts, 1990.

[6] J. Neira. *Geometric Object Recognition in Multisensor Systems.* PhD thesis, Dpto. de Ingeniería Eléctrica e Informática, University of Zaragoza, Spain, Agosto 1993.

[7] J. Neira, J. Horn, J.D. Tardós, and G. Schmidt. Multisensor mobile robot localization. In *Proc. 1996 IEEE Int. Conf. on Robotics and Automation*, pages 673–679, Minneapolis, USA, 1996.

[8] J. Neira, L. Montano, and J.D. Tardós. Constraint-based object recognition in multisensor systems. In *Proc. 1993 IEEE Int. Conf. on Robotics and Automation*, pages 135–142, Atlanta, Georgia, 1993.

[9] R. Talluri and J.K. Aggarwal. Mobile robot self-location using constrained search. In *Proc. 1991 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 903–908, Osaka, Japan, 1991.

[10] J.D. Tardós. Representing partial and uncertain sensorial information using the theory of symmetries. In *Proc. 1992 IEEE Int. Conf. on Robotics and Automation*, pages 1799–1804, Nice, France, 1992.

```
FUNCTION identify_before_locating (E, M)
; E: set of sensorial observations
; M: set of model features

    H_g := search_for_pairings (∅, E, M);
    H_v := locate_and_validate (H_g);
RETURN H_v;
END;

FUNCTION search_for_pairings (S_h, E, M)
; S_h: current set of pairings
; E: remaining observations to be paired
; M: set of candidate model features

    H := ∅;
    IF E = ∅ THEN
        H := H ∪ {S_h};
    ELSE
        e := select_observation (E);
        FOR m ∈ M DO
            p := (e, m);
            IF satisfy_unary_constraints (p) THEN
                binary := TRUE;
                FOR p_p = (e_p, m_p) ∈ S_h WHILE binary DO
                    binary := satisfy_binary_constraints (p_p, p);
                OD;
                IF binary THEN
                H := H ∪ search_for_pairings (S_h ∪ {p} , E \ {e} , M);
                FI;
            FI;
        OD;
        H := H ∪ search_for_pairings (S_h, E \ {e} , M);
    FI;
RETURN H;
END;

FUNCTION locate_and_validate (H)
; H: set of hypotheses h whose robot location has not been
; estimated and contain only the support pairings S_h

    H_v := ∅;
    FOR S_h ∈ H DO
        L_h := estimate__robot_location (S_h);
        valid := TRUE;
        FOR p ∈ S_h WHILE valid DO
        valid := satisfy_rigidity_and_extension_constraints (L_h, p);
        OD;
        IF valid THEN
            H_v := H_v ∪ h;
        FI;
    OD;
RETURN H_v;
END;
```

Algorithm 3.1: *Identifying before Locating*

```
FUNCTION identify_while_locating (E, M)
; E: set of available observations
; M: set of model features

    REPEAT
        e_f := select_first_observation (E);
        e_s := select_second_observation (E \ {e_f} , e_f);
        H_g := search_for_pairings (∅, {e_f, e_s} , M);
        H_v := ∅;
        FOR h ∈ H_g DO
            L_h := estimate_robot_location (S_h);
            H_v := H_v ∪ verify_hypothesis (h, E, M);
        OD;
        E := E \ {e_f, e_s};
    UNTIL H_v ≠ ∅;
RETURN H_v;
END;

FUNCTION verify_hypothesis (h, E, M)
; h : robot-location hypothesis to verify
; E: set of available observations
; M: set of model features

    H := ∅;
    IF E = ∅ THEN
        H := H ∪ {h};
    ELSE
        e := select_observation (E);
        FOR m ∈ M DO
            p := (e, m);
            IF satisfy_rigidity_and_extension (L_h, p) THEN
                L_{h_p} := refine_robot_location (L_h, p);
                S_{h_p} := S_h ∪ {p};
                H := H ∪ verify_hypothesis (h_p, E \ {e} , M);
            FI;
        OD;
        H := H ∪ verify_hypothesis (h, E \ {e} , M);
    FI;
RETURN H;
END;
```

Algorithm 3.2: *Identifying while Locating*

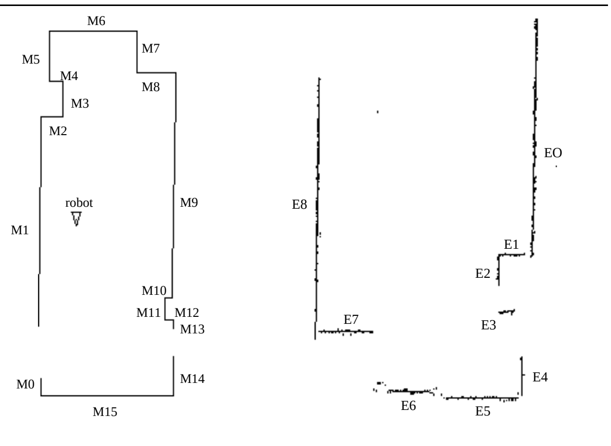

Figure 7: False Location obtained by the Identifying Before Locating matching scheme. Left figure shows the robot location in the model map. Right figures shows the set of observations available after segmentation of the laser data. The support set of the hypothesis is $\mathcal{S}_h = \{E0-M9,\ E4-M11,\ E5-M15,\ E6-M15,\ E8-M1\}$.

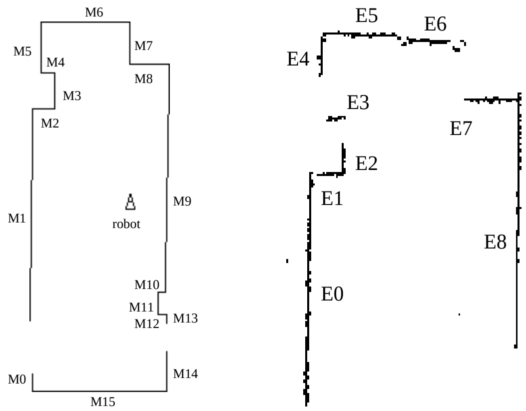

Figure 8: Mobile Robot Localization obtained by both matching schemes. Left figure shows the robot location in the model map. Right figures shows the set of observations available after segmentation of the laser data. The support set of the hypothesis is $\mathcal{S}_h = \{E0 - M1,\ E2 - M3,\ E4 - M5,\ E5 - M6,\ E7 - M8,\ E8 - M9\}$