Hough Localization for Mobile Robots in Polygonal Environments

Luca Iocchi, Daniele Nardi

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198, Roma, Italy
{iocchi,nardi}@dis.uniroma1.it

Abstract

Knowing the position and orientation of a mobile robot situated in an environment is a critical element for effectively accomplishing complex tasks requiring autonomous navigation and many techniques for robot self-localization have been extensively studied in the past.

In this paper we present a self-localization method that is based on the Hough Transform for matching a geometric reference map with a representation of range information acquired by the robot's sensors. The technique is adequate for indoor office-like environments, and specifically for those environments that can be suitably represented by a set of segments.

Many experiments are described to evaluate the effectiveness of the proposed method. Moreover we have successfully tested this method in some dynamic environments populated with unknown and moving obstacles (e.g. persons or other robots moving around): office environments as well as the RoboCup environment.

Key words: Mobile Robots, Self-Localization, Map matching, Hough Transform

1 Introduction

A general problem in mobile robotics is knowing the robot's pose (position and orientation) in the environment in which it is operating. This is a crucial feature for autonomous robots performing complex tasks over long periods of time, because failures in localization often lead to failures in task execution.

In the past years several techniques for robot self-localization have been developed (see [2] for a survey). Such techniques are based on design choices concerning the following three aspects:

(1) the type of sensors used for the localization task,
(2) the precision required by the localization module,
(3) the assumptions made on the environment.

The sensors used for localization determine the type of localization method that can be applied to the robot. We can distinguish between *relative* and *absolute* positioning methods: the former are based on determining the current pose of the robot from the previous pose and information about the robot's motion over a given interval of time, while the latter compute an absolute position of the robot in the environment (or position hypotheses in case of ambiguities), by means of global positioning systems (e.g. [31]), landmark recognition (e.g. [1,29,33,22]), or model matching (e.g. [17,5,32,4,9,3,14,25,15]). Relative methods are accurate only over short runs of the robot since they accumulate errors over time. On the other hand, global positioning systems and artificial landmark-based methods are effective as long as the environment can be appropriately structured. In addition, all these methods must deal with the difficulties in acquiring information from the environment due not only to sensor internal noise, but also to varying conditions in the environment (e.g. people or other robots moving around).

The second design choice that is important for localization is the precision required to the localization process, depending on the task the robot is performing. Metric localization aims at computing the correct coordinates of the robot within a global reference system in the environment, while topological localization aims at determining the robot's position with respect to a graph-like topological map (see for example [29,34]).

Localization methods presented in the literature often rely on some assumptions on the environment or on the robot capabilities. When these assumption are not valid during actual operation of the robot in the environment, these methods may either not work or lose precision and robustness. A typical assumption is to rely at every time on a good estimation of the robot's pose. This case is also known as *position tracking* problem. Another assumption is that the robot operates in a static environment, in fact some localization methods may lose precision and/or robustness in dynamic environments. A third assumption is the availability in the environment of features on which the localization method is based.

Among all the localization methods, we are mostly concerned with the ones computing metric localization based on map matching for determining the absolute pose of the robot in the environment. These methods can be divided into two groups depending on the representation of the reference map: 1) set of points (raw map); 2) set of geometric features (geometric map).

The first group (e.g. [17,32,4,25,9]) includes algorithms performing map matching by using either the points captured by the sensor device or occupancy grids, without any geometrical assumption on these data. A common feature for these approaches is that they can be robust to sensor noise, ambiguous situations, partial model descriptions. However, map matching tends to be computationally expensive, and in some cases these methods require heavy optimizations to implement an effective real-time localization on a mobile robot. In addition, in a dynamic environment, map matching based on raw data must also be robust to noise caused by unknown objects captured by range sensors, that may substantially change the local sensor data to be matched against the reference map. Some specific filtering techniques for dynamic environments are described in [9].

The second group of methods (e.g. [5,3,14,15]) make use of geometric features instead of raw points: therefore they require a preprocessing step in order to extract features (or natural landmarks) from the sensor data. Most of these methods deal with lines, segments, corners and the reference map is thus represented as a set of these features. These methods are usually more robust to noise given by unknown objects, as long as these objects do not affect feature recognition and reconstruction. However, their main drawback is that they rely on the availability of features in the environment and on the ability of the robot's perception system to detect them.

In this article we present a self-localization method, called Hough Localization, that provides for metric localization obtained by the integration of map matching with a relative method based on odometry. Hough Localization makes use of a geometric representation of the reference map (lines and circles) and it performs map matching in the domain given by the Hough Transform.

Hough Localization was introduced in [20,21] as the localization module of the soccer robots of the Azzurra Robot Team [28] competing in the RoboCup competitions, while the probabilistic formalization of the approach has been given in [19]. In this article we present the overall localization method and we show several experimental results for evaluating the effectiveness of the method in different environments.

While Hough Transform has been used in previous works on localization [32,12,18], the approach proposed in this article is called Hough Localization because it is based on a map matching in the Hough domain and not in the Cartesian space of the robot sensor reference system. This is the basic novel feature that makes our approach computationally efficient compared to other similar approaches. Following our approach, Marques and Lima [26] describe the application of Hough Localization with the use of an omnidirectional vision system able to see at every time all the environment (the RoboCup game field) in which the robot acts. The availability of a 360 degree view of the RoboCup soccer field allows for determining the position of the robot in the environment without any assumption on its initial pose.

Hough Localization can be applied to any robot equipped with a relative positioning system, which provides the displacement of the robot from the previous pose, and a range sensor, which returns a set of 2D points, in the local coordinates of the robot, corresponding to the visible surfaces of objects close to it. This is in fact a typical configuration for a mobile robot: motor encoders are usually used for integrating the robot's position and many range sensors can be used, such as ultrasonic sonars, laser range finders, or stereo vision systems. Observe that, in general, range sensors do not allow for simple implementation of object recognition techniques and thus they often retrieve range data from objects in the map (e.g. walls in the environment) as well as from unpredicted obstacles (such as persons moving in the world).

The method presented in this article is focussed on the position tracking problem in polygonal environments, i.e. environments that can be represented as a set of geometric features (like lines, circles, etc.). The approach is thus feasible in any office-like environment in which the availability of straight lines is usually guaranteed. The assumptions under which the method can be applied are thus: 1) the robot has at every time a good estimation of its pose in the environment; 2) the environment can be represented as a set of segments, thus suitable for polygonal office-like environments.

Hough Localization is based on matching the sensor data acquired by the range sensor with a model of the environment (a map), and to integrate this matching with odometry information. Hough Localization returns thus a probability distribution of the absolute pose of the robot in the map and the integration with odometry information is performed by means of an Extended Kalman Filter.

The main features of the proposed method in comparison with other localization approaches are:

(1) Hough Localization is effective in dynamic environments, since mobile objects partially occluding lines in the environment do not affect the map matching process. This is due to the properties of the Hough Transform that is able to detect an occluded line without any bias and is very robust to noise given by isolated points (i.e. sensor data coming from objects that are not modelled in the map).
(2) Position tracking assumption is easy to be achieved by any odometric system of a mobile robot. The largest position error that is acceptable for the method to work depends on the map of the environment and may typically be up to 45 degrees and 1 meter.
(3) Complexity analysis shows that the method is linear in the size of sensor data and in the number of reference lines in the map, thus making it

feasible for real-time execution.

We have done several experiments to validate the approach, both with the use of a simulator and by using real robots acting in an office environment, tracked by a global vision system in order to measure the true position of the robot. Moreover, we have successfully applied this method in the truly dynamic environment of the RoboCup middle-size league within the Azzurra Robot Team [28]. The technique turned out to be very fast and sufficiently accurate for all the tasks in which the robots were involved in playing soccer, even with the use of a vision-based range sensor, whose accuracy is considerably lower with respect to other range sensors (e.g. laser range finders).

The article is organized as follows. Section 2 describes the Hough Transform on which the method is based. Section 3 describes the probabilistic formalization of the Hough Localization method, the map matching process in the Hough domain and the integration with odometry. Section 4 describes experiments performed for evaluation of the effectiveness of the method. Related work is described in Section 5 and conclusions are drawn in Section 6.

2 Hough Transform

Hough Localization method is based on the matching between a known map of the environment and a local map built by the robot's sensors. Map matching is not performed in the Cartesian plane of the environment reference system, but in a parameter space obtained by the Hough Transform. In this section we introduce the Hough Transform (HT) and its properties that are useful for developing our localization method.

2.1 Hough Transform for lines

The Hough Transform is a robust and effective method for finding lines fitting a set of 2D points [6]. It is based on a transformation from the (x, y) plane (a Cartesian plane) to the (θ, ρ) plane (the Hough domain).

The transformation from (x, y) to (θ, ρ) is achieved by associating every point $P(x, y)$ in the Cartesian plane with the following curve in the Hough domain

$$\rho = x\, cos\theta + y\, sin\theta \tag{1}$$

At the same time, a point in the Hough domain corresponds to a line in (x, y). Notice that this is a *unique and complete* representation for lines in (x, y) as long as $0 \leq \theta < \pi$.

Given a set of sensor data $S = \{(x_i, y_i) \mid i = 1, .., n\}$, let us define the following functions:

$$h_i^S(\theta, \rho) = \begin{cases} 1 & \text{if } \rho = x_i cos\theta + y_i sin\theta \\ 0 & \text{otherwise} \end{cases}$$

$$HT_c^S(\theta, \rho) = \sum_{i=1}^{n} h_i^S(\theta, \rho)$$

The function $HT_c^S(\theta, \rho)$ will be called the *Hough Transform of the sensor data* S. In the following sections, however, we will make use of a discrete representation of this function that we denote with $HT^S(\theta, \rho)$ and that is obtained by generating a discrete grid of the (θ, ρ) plane (let $\delta\theta$ and $\delta\rho$ be the step units) and by defining $HT^S(\theta, \rho)$ as the number of points (x, y) whose corresponding curve (1) lies within the interval $[\theta, \theta + \delta\theta] \times [\rho, \rho + \delta\rho]$.

Observe that it is possible to consider the discrete Hough Transform of S as a voting space for points in (x, y), in which every point in (x, y) "votes" for a set of lines (represented as points in (θ, ρ)), that are all the lines passing through that point. Notice also that, in the case of a set of aligned points in (x, y), the point in the Hough domain that "receives" the highest number of votes is the one corresponding to the line passing through all these points.

The Hough Transform has a number of notable properties:

(1) Given a set of input points, a local maximum of $HT(\theta, \rho)$ corresponds to the best fitting line of these points. Given a set of input points originally belonging to several lines, local maxima of $HT(\theta, \rho)$ correspond to the best fitting lines for each subset of points relative to a single line.
(2) With respect to other techniques for extracting segments from a set of points, the Hough Transform is very robust to noise produced by isolated points (since their votes do not affect the local maxima) and to occlusions of the lines (since point distances are not relevant). Moreover, in presence of points belonging to multiple lines no clustering is needed.
(3) Measuring displacement of lines in the Cartesian plane corresponds to measuring distance of points in the Hough domain. Indeed, the distance between parallel lines and the angular difference between lines is given respectively by a $\Delta\rho$ and a $\Delta\theta$ between the corresponding points in the Hough domain.

An interesting property, that will be useful in the following sections, is in the relation between the transformations of the sensor readings when the robot moves.

Property 1. Given the Hough Transform of a set of sensor readings S, $HT^S(\theta, \rho)$, and a rotation/translation (T_x, T_y, θ_R) of the robot, the Hough Transform of S with respect to the new pose of the robot will be $HT^S(\theta', \rho')$ such that [1]:

if $0 \leq \theta + \theta_R < \pi$ then

$$\theta' = \theta + \theta_R$$
$$\rho' = \rho + T_x\, cos(\theta + \theta_R) + T_y\, sin(\theta + \theta_R)$$

if $\theta + \theta_R \geq \pi$ then

$$\theta' = \theta + \theta_R - \pi$$
$$\rho' = -(\rho + T_x\, cos(\theta + \theta_R) + T_y\, sin(\theta + \theta_R))$$

if $\theta + \theta_R < 0$ then

$$\theta' = \theta + \theta_R + \pi$$
$$\rho' = -(\rho + T_x\, cos(\theta + \theta_R) + T_y\, sin(\theta + \theta_R))$$

It is important to notice here that if $\theta_R = 0$ (i.e. the robot does not rotate) then $\theta' = \theta$ (i.e. θ does not change), and conversely if $T_x = T_y = 0$ (i.e. the robot does not translate) then $|\rho'| = |\rho|$ (i.e. the absolute value of ρ does not change). In other words, robot's alignment can be divided in two separate steps: first determining the orientation with a null translation, and then determining the translation with a null rotation.

2.2 Hough Transform for circles

The Hough Transform can be extended for detecting circles from a set of points by using the following parametric curve:

$$(x - \alpha)^2 + (y - \beta)^2 = r^2$$

If we assume that r is known (and thus constant), we have to determine only two parameters α and β corresponding to the center of the circle. The Circle

[1] We assume here that $| \theta_R | \leq \pi$

Hough Transform for the sensor data S will be denoted with $CHT^S(\alpha, \beta)$.

A property similar to the one for lines can be considered.

Property 2. Given the Circle Hough Transform of a set of sensor readings S, $CHT^S(\alpha, \beta)$, and a rotation/translation (T_x, T_y, θ_R) of the robot, the Circle Hough Transform of S with respect to the new pose of the robot will be $CHT^S(\alpha', \beta')$ such that:

$$\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} = \begin{pmatrix} cos\,\theta_R & -sin\,\theta_R \\ sin\,\theta_R & cos\,\theta_R \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + \begin{pmatrix} T_x \\ T_y \end{pmatrix}$$

3 Hough Localization

Hough Localization is based on map matching between the Hough representation of two maps: a map of the environment and a local map built by the robot's sensors. The displacement between the two maps is used for evaluating the displacement between the real pose of the robot and the estimated one, and thus to localize the robot in the environment.

In many recent works [4,13,9,30], localization is considered as the problem of evaluating the most likely pose of the robot given all the information on the environment coming from the sensor devices, and thus it is expressed as the task of evaluating the probability that the robot is at a certain location, given all the sensor readings.

We assume that at every time-step t the following data are available to the robot: data from the relative positioning system A^t and data from the range sensor S^t. We also denote with $p(l)$ the probability distribution of the robot's pose, with $l = (x, y, \theta) \in \Re^2 \times [0, 2\pi)$.

Localization can be expressed as the task of evaluating the probability distribution $p(l \mid A^t, S^t)$ from the previous distribution $p(l \mid A^{t-1}, S^{t-1})$, the current sensor readings A^t and S^t, and a reference map $\mathcal{M}$.

This task is usually performed in two steps:

(1) *Prediction.* Predicting the new pose of the robot by dead reckoning from the previous position (that is computing $p(l \mid A^t, S^{t-1})$ from $p(l \mid A^{t-1}, S^{t-1})$ and A^t).
(2) *Update.* Updating the robot pose with the results of a map matching process between S^t and $\mathcal{M}$ (that is computing $p(l \mid A^t, S^t)$ from $p(l \mid A^t, S^{t-1})$, S^t, and $\mathcal{M}$).

Therefore the map matching problem can be defined as follows: given a set of points acquired by the robot's range sensors S^t and a model of the environment $\mathcal{M}$, calculate the displacement between the estimated and the current pose of the robot.

Hough Localization is based on map matching between sensor data and a reference map that, under the assumption that the environment can be represented by a set of segments, is performed in the Hough domain. In this way, the model of the environment will be represented by a set of points in the Hough domain and the range data points acquired through the sensors are transformed in the Hough plane as described in the previous section. Local maxima of the Hough Transform correspond to the best fitting lines of the sensor points and they will be matched against the map reference points for computing the displacement needed for a correct re-positioning of the robot.

Finally, the result of map matching must be integrated with odometric information in order to determine the "best" estimated pose of the robot.

Summarizing, the Hough Localization method consists in the following steps:

(1) extracting range information from the environment in the form of a set of point S in the (x, y) plane,
(2) generating the discrete Hough Transform $HT^S(\theta, \rho)$ of such points,
(3) determining the local maxima of $HT^S(\theta, \rho)$ (for instance by a threshold),
(4) finding correspondences between local maxima and reference points,
(5) measuring the displacement between local maxima and the corresponding reference points in the Hough domain (that corresponds to the displacement between the estimated and the current pose of the robot),
(6) integrating map displacement with odometric information.

The critical step of this procedure is the fourth one, that is finding the correct correspondence between local maxima and reference points. Indeed errors in assigning correspondences usually lead to large positioning errors that are then difficult to recover.

In this article we focus the attention on the position tracking problem, in which an initial guess of the position of the robot is available before performing the map matching process. Position tracking relies on a *small error assumption*, i.e. the actual pose of the robot is close to the estimated one. In the case of Hough Localization, this assumption depends on the map of the environment, and in particular, it depends on the distance of map reference points in the Hough domain. More specifically, if $\Delta\theta^{\mathcal{M}}$ is the minimum distance on the θ axis between two map reference points in the Hough domain, and $\Delta\rho^{\mathcal{M}}$ is the minimum distance on the ρ axis between two map reference points in the Hough domain, then the position tracking assumption is that the position error of the robot must be at every time less than $\Delta\theta^{\mathcal{M}}/2$ in orientation and

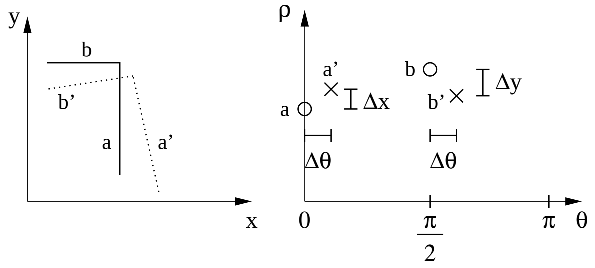

Fig. 1. Map matching in the Hough domain

$\Delta\rho^{\mathcal{M}}/2$ in translation. For example if we consider an orthogonal environment, in which all the lines are perpendicular among them and therefore there are only 2 possible directions of segments in the map, with a minimum distance of 1m between two lines, we can accept positioning errors up to 45 degrees in orientation and 50 cm in translation. If the position error is greater than this limit, it is not guaranteed that the robot can localize itself correctly. However, this limit can be easily achieved by any standard odometric system mounted on a mobile robot.

Position tracking is usually addressed by representing the probability distribution of the robot's pose $p(l)$ as a Gaussian, whose mean l_k is the most likely position of the robot and the covariance matrix P_k represents its variance.

Under the small error assumption, the correspondence problem can be easily addressed by adopting a closest matching approach. Indeed, under this assumption, it is possible to easily match a local maximum of HT^S, (θ^S, ρ^S), with the closest reference point $(\theta^{\mathcal{M}}, \rho^{\mathcal{M}})$.

In other words, it is possible to define a set of intervals in the Hough domain around the reference points $(\theta^{\mathcal{M}}, \rho^{\mathcal{M}})$ and consider only the local maxima of HT^S belonging to these intervals. Furthermore, for computational efficiency, we actually compute HT^S only in these intervals.

Consider the example shown in Fig. 1, where the robot faces a corner. The solid segments *a, b* represent the map model and the set of points *a', b'* represent data coming from sensor device. All (x, y) data here (and in the following) are expressed in a robot's centered reference system [2]. The four segments are also displayed in the Hough domain: *a, b* (indicated by a circle) are the reference points, while *a', b'* (indicated by a cross) represent the local maxima of the Hough Transform applied to the set of input points.

Under the small error assumption, the correspondence problem is solved in this case by assigning a' to a and b' to b. Therefore, the displacement between the estimated and the actual pose of the robot $(\Delta x, \Delta y, \Delta\theta)$ is easily computed

[2] Observe that it is usually more efficient to convert map data in the robot's reference system than sensor points in a global reference system.

in the Hough domain, as shown in the Figure.

According to the Property 1 described in Section 2, we can first compute the orientation $\Delta\theta$ (with ρ constant) and then Δx, Δy (with θ constant). In the example, $\Delta\theta$ is the difference $a'_\theta - a_\theta$ or $b'_\theta - b_\theta$. In ideal conditions these differences should be the same; if not, an average between these values allows for a good approximation. After the correction $\Delta\theta$ is applied to the robot's representation of the map, it is possible to calculate the other two elements $\Delta x = a'_\rho - a_\rho$ and $\Delta y = b'_\rho - b_\rho$, that are actually used for re-positioning the robot.

In case only a set of parallel lines are visible from the sensor, it is not possible to compute all the values Δx, Δy, and $\Delta\theta$, but it is still possible to compute at least $\Delta\theta$, applying only an angular correction.

3.1 Sensor data integration with Kalman filter

The map matching method described above provides for a correction of the estimated position of the robot that must be integrated with odometric information. A standard technique for this integration (that is suitable when the probability distribution of the pose of the robot is represented by a Gaussian) is using an Extended Kalman Filter (e.g. [24]).

We can describe the dynamics of the robot, with internal state $l_k = (x_k, y_k, \theta_k)^T$ (corresponding to the estimation of the robot pose), input $u_k = (\delta_k, \alpha_k)^T$ (corresponding to odometric measurement of translation and angular difference with respect to the previous pose), and output $z_k = (\hat{x}_k, \hat{y}_k, \hat{\theta}_k)^T$ (corresponding to a measurement of the robot pose), as follows

$$
l_{k+1} = \begin{pmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{pmatrix} = \begin{pmatrix} x_k + \delta_k \, cos\,\theta_k + w_\delta \, cos\,\theta_k \\ y_k + \delta_k \, sin\,\theta_k + w_\delta \, sin\,\theta_k \\ \theta_k + \alpha_k + w_\alpha \end{pmatrix} = l_k + B_k u_k + W_k w
$$

$$
z_k = \begin{pmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \end{pmatrix} = \begin{pmatrix} x_k + v_x \\ y_k + v_y \\ \theta_k + v_\theta \end{pmatrix} = l_k + v
$$

where

$$B_k = W_k = \begin{pmatrix} cos\,\theta_k & 0 \\ sin\,\theta_k & 0 \\ 0 & 1 \end{pmatrix}$$

The vectors $w = (w_\delta, w_\alpha)^T$ and $v = (v_x, v_y, v_\theta)^T$ are random variables representing respectively noise in odometric data and noise in the map matching process. For these random variables we assume a Gaussian white noise with zero mean and covariance matrices Q_k and R_k.

Extended Kalman filtering is performed in two steps:

1. Prediction. An estimated pose l_{k+1}^- of the robot is computed from the previous pose and odometry and the covariance matrix is updated.

$$\begin{aligned} l_{k+1}^- &= l_k + B_k u_k \\ P_{k+1}^- &= A_k P_k {A_k}^T + W_k Q_k {W_k}^T \end{aligned}$$

where A_k and W_k are the Jacobians of the robot dynamics with respect to l and w respectively:

$$A_k = \begin{pmatrix} 1 & 0 & -\delta_k\, sin\,\theta_k \\ 0 & 1 & \delta_k\, cos\,\theta_k \\ 0 & 0 & 1 \end{pmatrix} \qquad W_k = \begin{pmatrix} cos\,\theta_k & 0 \\ sin\,\theta_k & 0 \\ 0 & 1 \end{pmatrix}$$

2. Correction. The pose of the robot is corrected by the result of the map matching process. Indeed z_{k+1} represent the new pose of the robot according to map matching.

$$\begin{aligned} K &= P_{k+1}^-(P_{k+1}^- + R_k)^{-1} \\ l_{k+1} &= l_{k+1}^- + K(z_{k+1} - l_{k+1}^-) \\ P_{k+1} &= (I - K)P_{k+1}^- \end{aligned}$$

The effective performance of this filter depends on the definition of the matrices Q_k and R_k, whose values should be determined according to the characteristics of the robot's sensor devices. Currently in our implementation the values for Q_k and R_k have been determined by executing several experiments on our robots. Specifically, we have recorded several runs of the robot in the environment (storing the values for u_k and z_k over time) and we have processed these data with different parameters of the filter in order to obtain "good" values for Q_k and R_k. It is possible to make a more detailed analysis on this issue, but this is beyond the aims of this article.

3.2 Complexity analysis

Hough Localization is very efficient, since its computational complexity is $O(|\mathcal{S}||\mathcal{M}|)$, thus linear in the size of sensor data and of the number of map reference lines. In fact, we apply the Hough Transform to all the $|\mathcal{S}|$ points acquired by the range sensor and for each of these points we compute equation (1) for the intervals around the $|\mathcal{M}|$ map reference points.

The performance of the system is adequate for real-time execution with a low-cost color camera and a conventional Pentium based PC, that is on board the robot. In fact, in our case, most of the computation time (currently around 30 ms.) is taken by the image processing procedure for the extraction of points belonging to lines and walls of the environment, while the Hough method itself takes only a few milliseconds of computation time on a 400MHz Pentium CPU.

4 Experiments

Several experiments have been performed for evaluating the precision and the accuracy of the Hough Localization method presented here. In this section we first describe some experiments that make use of a simulator, in which we have evaluated precision and robustness of the approach under controlled conditions; then experiments with real robots are presented in order to test the effectiveness of the method under real operation conditions.

4.1 Experiments with a simulator

The accuracy of a localization method usually depends on the precision of the range sensor. In this section we present some experiments obtained by a simulation of a range sensor for a mobile robot. These experiments are useful for evaluating the method under controlled noise of the range sensor.

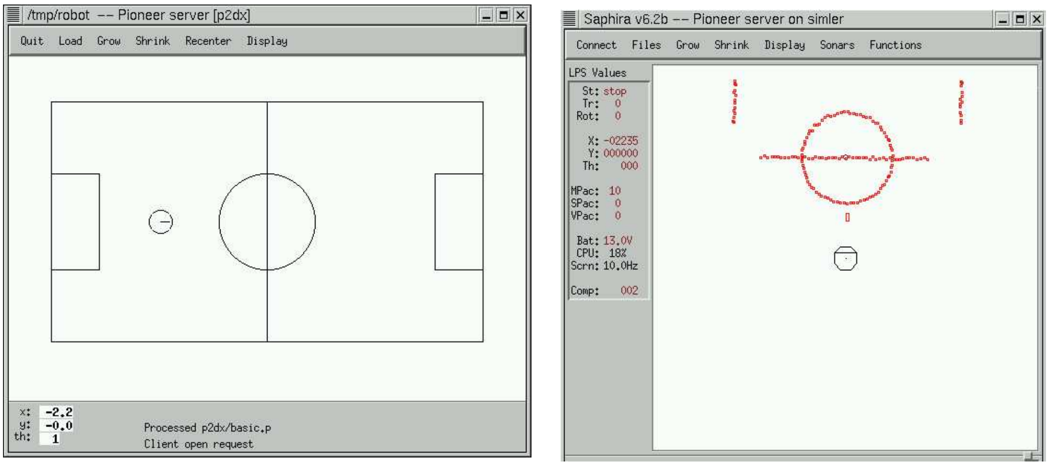

Fig. 2. Vision range simulator

We have developed a simulator of a range sensor for a mobile robot [3] in order to evaluate Hough Localization under different controlled conditions. The simulator includes a mathematical model of several different types of typical noises that occur during robot navigation and sensor perception; in this way, we can model different range sensors (like vision-based, laser range finders, ultrasonic sonars). In particular, we have modeled a vision-based range sensor that is able to extract points both from the walls and from lines drawn on the floor of an environment (a vision-based range sensor has actually been used in the RoboCup game field [21]). In Fig. 2 a snapshot of the simulator and the corresponding sensor data extracted by the simulated vision range sensor are shown.

The following kinds of noise have been implemented in the simulator:

- *odometric noise*: the position of the robot is affected by a random noise, such that the position error generally increases over time;
- *sensor noise*: sensor data are affected by a random noise that increases with the speed of the robot (this is used for taking into account vibrations of the camera when a robot moves, that introduces additional errors in the extracted data;
- *systematic error*: sensor data are affected by a systematic error that corresponds to errors in the calibration of the camera.
- *robot bumps*: random movements of the robot in the environment;
- *false positives*: addition of points that do not belong to any line;
- *occlusions*: removal of points that belong to a line, as occluded by other objects in the environment.

The models of the environments considered in our experiments are formed by

[3] We are grateful to Kurt Konolige for his permission to extend the Pioneer simulator.

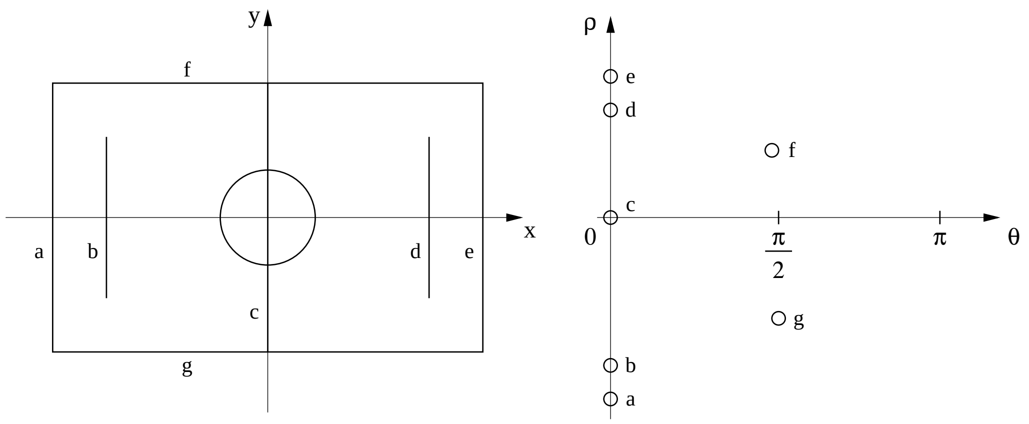

Fig. 3. The RoboCup world model

sets of segments. For example, in the RoboCup environment (see Fig. 3), we have four segments representing the boards (a,e,f,g), three segments (b,c,d) and one circle drawn in the field. In an office environment segments would represent walls of corridors and circular columns could be represented as circles.

With the use of a simulator it is possible to know exactly the *true pose* of the robot at every instant and thus evaluate the position error $\epsilon(t)$ as the distance between the true pose $r(t)$ and the estimated one $l(t)$:

$$\epsilon(t) = ||\, l(t) - r(t)\, ||$$

The graphs in Fig. 4 describe the position error $\epsilon(t)$ (in mm) for a typical navigation of the robot in this environment. The three lines for each graph represent different errors during the same experiment: 1) the solid light line is the odometry error (without any localization method applied), 2) the dashed line is the error of Hough Localization without Kalman filtering, 3) the bold line is the error of Hough Localization with Kalman filtering. In the second experiment (Fig. 4b) the robot was randomly moved to a different position (at time 100, 200, and 300) that was about 20 cm away form the current position and with an error of about 20 degrees in orientation.

Since we are within the limits of the position tracking assumption for Hough Localization in this environment, the robot can recover from this positioning error.

4.2 Experiments with real robots

Evaluation of localization methods with real robots is difficult, because the approach used with simulation cannot easily be extended. In fact, the real pose

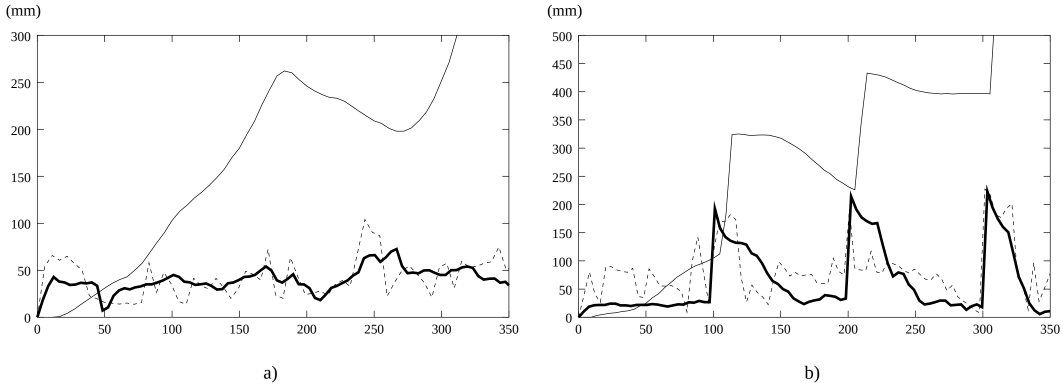

Fig. 4. Position error (in mm): odometry vs. Hough Localization

of the robots $r(t)$ is not known and it is difficult to measure, unless a setting with precise sensors is available. Moreover, in several robotic applications, environments cannot be modified and a setting for measuring robot positions cannot be set up. Therefore, in these cases, we have only an approximate measure of the real pose of the robot $r'(t)$, and thus we can compute the position error for a real robot with

$$\epsilon'(t) = ||\, r'(t) - l(t)\, ||$$

However, notice that in some cases the measurement error in determining the true robot's pose is comparable with the localization error, thus affecting the evaluation of the method. Moreover, since it is generally difficult to measure $r'(t)$ when the robot is moving, the position error $\epsilon'(t)$ is often manually computed only at some instants of time (when the robot is stopped) [13], but this procedure can again affect the results of evaluation, given that, with some kinds of sensors (like vision-based ones), data are more noisy when the robot is moving fast.

4.2.1 Experimental setting

The experimental setting we describe in this section has been implemented with the goal of evaluating the performance of the method, considering in particular its robustness in dynamic environments under real operating conditions of the robots.

Specifically, we have implemented a global vision system, that makes use of a fixed camera positioned outside the game field for measuring the actual pose of the robot. This setting does not introduce any modification in the environment, since vision system is a passive sensor. The images contain a global view of the field and they are analyzed for recognizing a special marker put on the robot and for determining its pose in the field.

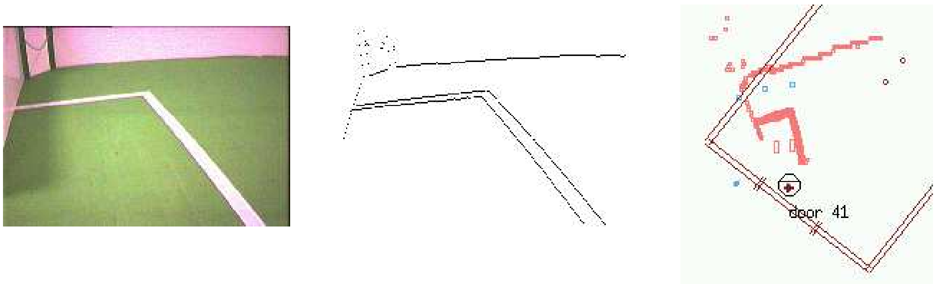

Fig. 5. Vision based range data extraction (RoboCup environment)

Fig. 6. Vision based range data extraction (office environment)

With this setting, as in the experiments with the simulator, we are able to estimate the robot's pose $r'(t)$ and, consequently, monitor the robot's position error $\epsilon'(t)$ during navigation and evaluate the robustness of the self-localization under real conditions. In other words, our experimental setting allows for determining an upper bound to the position error of the robot during its normal activities.

4.2.2 Experimental results

Experimental evaluation of Hough Localization depends on the accuracy of the range sensor used. For the experiments described here we have used a vision based range sensors formed by a single CCD color camera put on board the robot. The reasons for this choice have been not only the reduced cost with respect to other sensors like laser range finders, but also the intention to exploit our capabilities of using vision to detect both the walls and the lines painted on the ground. Obviously, the precision of such a vision based range sensor is not optimal with respect to other range sensors, and it requires an additional effort for the calibration of the system, it is however adequate for our goal to verify the robustness of Hough Localization.

In Fig. 5 and Fig. 6 two examples of data extracted by the vision sensor are given: the first set of data are from the RoboCup context, the second set is from a typical office-like environment. Observe that the vision based sensor is used to detect not only walls in the environment, but also lines drawn on the ground, that can be effectively used for localization. A detailed description of the realization of the vision based range sensor is given in [21].

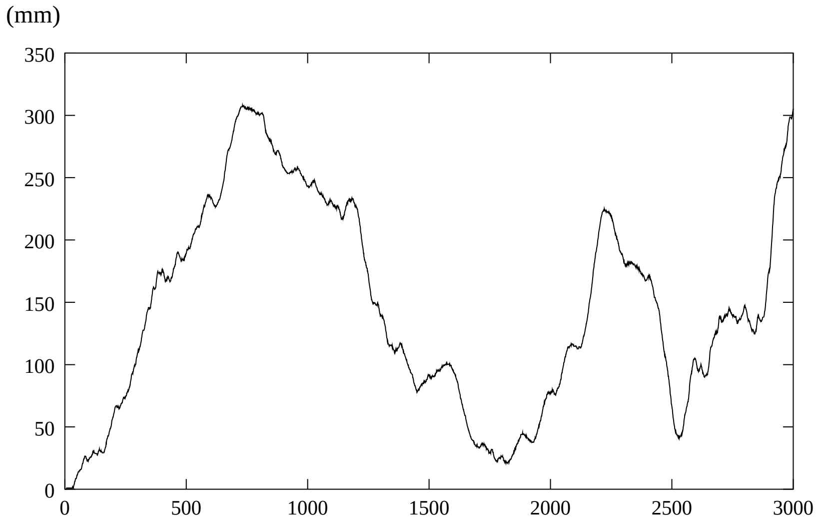

Fig. 7. Localization error (in mm) over time with a real robot

In order to evaluate the performance of our localization method we have done experiments with real robots monitoring their position during typical tasks in the RoboCup environment. The robot position has been taken both as a result of Hough Localization method $l(t)$ and from an external vision system able to track the robot $r'(t)$. Given these two trajectories, we have compared them by computing the distance of the two robot positions over time $\epsilon'(t)$.

In Fig. 7 we show a section of the function $\epsilon'(t)$, representing the estimated position error. In the global experiment (that was run for about 25 minutes) we had a maximum error of 35 cm and an average error of 15 cm. These numeric values are affected by the fact that the global vision system can determine the real robot position only with a precision of about 15 cm, that is thus comparable with the average error determined.

The error in measuring the real robot position makes this setting not suitable for a precise evaluation of the accuracy of the localization method, but instead it has been very useful for evaluating the robustness of localization. In fact, in this way, we have determined an upper bound for the precision of the method under real operating conditions: the position error is always limited within a threshold that is in the worst case 50 cm (35 cm from localization + 15 cm from the global vision system).

Experiments in the corridors of our laboratory have been performed with similar results [27]. In fact, the precision of the method is bound by the accuracy of the vision based sensor (the same for these two environments). However, the ability of the vision system to detect walls as well as reference lines drawn on the ground is exploited to increase the number of features available for localization thus improving the robustness of the method.

Because of the differences in the sensors used and in the aim of the experiments, the numerical values given here cannot be directly compared with other approaches (e.g. [9,13]).

In addition to these experiments performed in our laboratory, the Hough Localization method has been implemented on the robots of the Azzurra Robot Team, and effectively used in the 1999 and 2000 RoboCup competitions within the F2000 League. In this league [23], (i) each soccer player is equipped only with on-board acting and sensing devices, while global positioning systems are not allowed; (ii) the environment is highly dynamic (there are many robots and the ball moving in the field); (iii) the task must be performed continuously for a “long” time (the length of each period is 10 minutes); (iv) the environment cannot be modified; (v) collisions between robots are possible. The characteristics of the F2000 RoboCup setting give rise to a difficult scenario for localization methods and thus it has been a good benchmark for proving robustness of our localization method.

The results of our experiments have proven that the Hough Localization method is very robust in real dynamic environments even by using a low-accuracy range sensor based on a vision system. Simulated experiments have also shown that the precision of the Hough Localization method depends on the accuracy of the range sensor and thus, by using more precise range sensors (like for example a laser range finder) the precision of localization can be significantly improved.

5 Related work

Localization for mobile robots has been extensively studied in the past years and several different approaches have been proposed. Among all, the most common methods are based on map matching for determining the metric absolute pose of the robot in the environment and on the integration of this matching with relative positioning information. The Hough Localization presented here is based on a map matching between a geometric representation of the environment and of the robot sensor data, that is performed in the Hough parameter space, and thus it can be directly compared to some recent works.

The localization method presented in [32] is based on two certainty grids: a global certainty grid, containing data for the reference map, and a local certainty grid containing data acquired by a range sensor. The Hough Transform is used for detecting segments from the two grids and the association problem for determining pairs of segments is addressed by comparing each pair of segments in order to detect similarities. The computational time for this method is $O(|\mathcal{S}||\mathcal{H}_\theta| + |\mathcal{H}_{max}||\mathcal{M}|)$, where $\mathcal{S}$ is the set of sensor data, $\mathcal{H}_\theta$ is the discretization of the Hough space over the θ coordinate, $\mathcal{H}_{max}$ is the set of local maxima extracted from the Hough space, and $\mathcal{M}$ is the number of reference segments in the map. The Hough Localization method presented here differs from the above referenced approach mainly in the fact that matching is performed directly in the Hough space so that, under the position tracking assumption, the computational time is given by $O(|\mathcal{S}||\mathcal{M}|)$. Therefore we obtain an increase in efficiency considering that usually the number of segments in a reference map $|\mathcal{M}|$ is significantly less than the discretization of the Hough space $|\mathcal{H}_\theta|$.

Other feature-based map matching approaches within the Markov Localization framework are described in [33,22]: they make use of natural landmarks extracted from sensor data for matching with reference landmarks. In these works the environment in which the robot acts is divided into regions characterized by the presence of specific landmarks and Markov Localization is applied for determining a coarse estimation of the robot position (topological localization). The approach presented in this paper makes use of geometric features in the environment, but it is designed for metric localization, thus allowing for a more precise estimation of the robot's pose.

Among the methods that do not consider geometric features during the map matching process the most common is Markov Localization described in [9,11]. The method is based on an explicit representation of the probability distribution of the robot's pose in the environment that is updated according to the sensor readings during robot operation. It is effective in any (not only polygonal) environment and it is adequate for dynamic environments. However, the computation time of the update process is linear in the three dimensions of the robot space (x, y, θ) and thus the method tends to be computationally expensive and it requires some heuristics to be applied in order to improve performance [9,10]. Another recent interesting work is Monte Carlo Localization [8], that is a slightly different version of Markov Localization. While Markov Localization requires a representation of the whole environment, Monte Carlo Localization is based on a set of samples representing the possible robot's poses in the environment. In other words, only the small portion of the environment in which the robot may be is represented. These samples are generated during robot motion using odometry information and are updated according to sensor measurements. Monte Carlo Localization is more efficient than Markov Localization since computation time is linear in the number of samples that are only a small part of the whole environment. In contrast with the above approaches, in which computation time is linear in the size of the environment or in the number of possible poses of the robot in the environment, in our approach computation time depends only on the number of reference elements in the map. Therefore, although our approach is limited to position tracking and in polygonal environments, in these situations it performs more efficiently than Markov-based approaches.

Other recent works on localization are based on scan matching or line matching. In [25] a local search in the robot's pose space is performed in order to find the best overlap between the current scan and the reference map, without extracting geometric features from the sensor data. This method can be used in any environment, but it is more sensible to noise given by unknown objects, that usually affects the results of map matching. Our method instead exploits the properties of the Hough Transform in order to discard noisy points (from unknown objects) not belonging to any line: in this way the result of matching in our case is not affected by this noise and thus the method performs well in dynamic environments. In [15] a LineMatch algorithm is proposed for matching lines extracted from sensor data points. This method must deal with the clustering problem (i.e. associating every point to the line to which it belongs in case of multiple lines) as well as with noisy points not belonging to any reference line. The main difference with our approach is that in their case line extraction and matching is performed in the Cartesian space, while in our case we exploit the features of the Hough Transform for clustering points associated to multiple lines and for discarding noisy points, that allow for a more efficient implementation.

Following the approach proposed in this paper, Hough Localization has been also used in [26] for mobile robots equipped with an omnidirectional vision system able to see at every time all the environment (the RoboCup field) in which the robot acts, and thus to extract lines from the environment to be matched with a reference map. A 360 degrees view of the RoboCup soccer field usually allows for determining the position of the robot in the environment without the need of knowing an estimate of the initial robot pose and thus without the need of integrating a relative positioning system. Although the method is quite effective in the RoboCup environment, it cannot be easily extended to general environments, in which it is usually not possible to see at every time all the features needed for a global localization.

Among successful applications of localization methods in the RoboCup environment, besides the work in [15], that we have already mentioned, there are some interesting approaches that makes use of vision for detecting features from the environment. One of these methods is described in [16]. In this work 2D projections of curves extracted from images are matched against reference curves characterizing the environment. This matching is performed by minimizing the distance between the curves and the integration of model matching and odometry is performed by using a technique similar to the Extended Kalman Filter. This approach is similar to ours, except for the fact that matching is performed in the Cartesian space and, as other similar methods performing matching in the Cartesian space, the need of solving the data association problem (that is associating every point to the corresponding curve to which it comes from) may, in general, affect the efficiency of the method. Another method presented in [7] describes the application of Monte Carlo Localization in the RoboCup environment, based on the extraction of particular features (or natural landmarks) available in the environment, such as goal posts, corners, etc. As a difference with their approach, we extract from the environment a set of points belonging to lines and walls and then transform these points into lines or circles that are used as features for map matching. The properties of the Hough Transform allow to detect a line or a circle even if it is partially occluded: in this way, the situations in which features are not available for map matching are very limited. As in our case, the accuracy of the method depends from the precision of the vision system and typical localization errors reported in their work are comparable with ours. However, localization methods based on natural landmarks (such as [7]) may fail in detecting features when, in dynamic environments (like RoboCup), these landmarks are frequently occluded for a long time, and these situations may affect the robustness of these methods in dynamic environments.

6 Conclusion

Knowing the position of a mobile robot in an environment (and specifically in the RoboCup environment) is a critical element for effectively accomplishing complex tasks requiring autonomous navigation. The localization problem has thus been investigated from many different perspectives.

In this paper we have presented a self-localization technique for mobile robots that is suitable with any kind of sensors able to provide range information about objects in the world. We exploit the robustness properties of the Hough Transform for defining an effective and robust self-localization method for dynamic environments. Hough Localization method is adequate for indoor office-like polygonal environments, and specifically for those environments that can be represented by a set of segments. Since the availability of geometric features is critical for the effectiveness of the method, we have experimented a vision based sensor, that is able to detect both walls and lines drawn on the ground, so that it is possible to reduce situations in which features are not detected for a long time, thus increasing the robustness of the method.

Robustness in dynamic environments is guaranteed by the ability of the Hough Transform to detect lines without any bias in presence of partially occluded lines and noise given by isolated points representing (mobile) objects that are not modelled in the map. Finally, the method is linear in the number of sensor data and in the number of reference lines in the map, therefore its performance is adequate for real-time execution. An extensive set of experiments have been performed both by using a simulator and with real robots. Effectiveness of the method has been also proven by its use on the robots of the Azzurra Robot Team during official RoboCup matches in 1999 and 2000.

Future work includes the extension of the method for releasing the position tracking assumption and thus attaching the global localization problem. Global localization can again be determined in the Hough domain, but without the small error assumption the straightforward application of the technique proposed in this paper will substantially increase the computational time of the process, because a search in the Hough space is required to associate local maxima to reference points. We are currently investigating suitable techniques to solve this problem.

7 Acknowledgements

This work has been supported by Facoltà di Ingegneria, University of Rome "La Sapienza", "Progetto Robot Calciatori" funded by Italian CNR, "Progetto Robot Calciatori Sapienza 2001", funded by University of Rome "La Sapienza". We are particularly grateful to Giorgio Grisetti, Domenico Mastrantuono, and Daniele Baldassari for their help in the implementation and for the experimental work. We also greatly appreciate the suggestions of the anonymous reviewers that helped us in improving the quality of this article.

References

[1] M. Betke and L. Gurvits. Mobile robot localization using landmarks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, pages 135–142, 1994.

[2] J. Borenstein, H. R. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques.* A. K. Peters, Ltd., 1996.

[3] S. Borthwick and H. Durrant-White. Simultaneous localization and map building for autonomous guided vehicles. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 761–768, 1994.

[4] W. Burgard, D. Fox, D. Hennig, and T. Schnidt. Estimating the absolute position of a mobile robot using position probabilities grid. In *Proc. of 14th National Conference on Artificial Intelligence (AAAI'96)*, pages 896–901, 1996.

[5] I. J. Cox. Blanche - an experiment in guidance and navigation of an autonomous mobile robot. *IEEE Trans. on Robotics and Automation*, 7(2):193–204, 1991.

[6] R. Duda and P. Hart. Use of the hough transformation to detect lines and curves in the pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[7] S. Enderle, M. Ritter, D. Fox, S. Sablatnog, G. Kraetzschmar, and G. Palm. Vision-based localization in robocup environments. In *RoboCup-2000: Robot Soccer World Cup IV*, 2000.

[8] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proc. of the 16th National Conference on Artificial Intelligence (AAAI99)*, 1999.

[9] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.

[10] D. Fox, S. Thrun, F. Dellaert, and W. Burgard. Particle filters for mobile robot localization. In A. Doucet, N. De Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2000.

[11] Dieter Fox. *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation.* PhD thesis, Institute of Computer Science III, University of Bonn, Germany, 1998.

[12] A. Grossmann and R. Poli. Robust mobile robot localisation from sparse and noisy proximity readings. In *Proc. of Workshop on Reasoning with Uncertainty in Robot Navigation*, 1999.

[13] J. S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.

[14] J. S. Gutmann and C. Schlegel. AMOS: Comparison of scan matching approaches for self-localization in indoor environments. In *1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT)*, 1996.

[15] J.-S. Gutmann, T. Weigel, and B. Nebel. Fast, accurate, and robust self-localization in the robocup environment. In *RoboCup-99: Robot Soccer World Cup III*, pages 304–317, 1999.

[16] R. Hanek, T. Schmitt, M. Klupsch, and S. Buck. From multiple images to a consistent view. In *RoboCup-2000: Robot Soccer World Cup IV*, 2000.

[17] R. Hinkel and T. Knieriemen. Environment perception with a laser radar in a fast moving robot. In *Proc. of Symposium on Robot Control (SYROCO'88)*, pages 68.1–68.7, 1988.

[18] J. Howell and B. R. Donald. Practical mobile robot self-localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2000)*, 2000.

[19] L. Iocchi, D. Mastrantuono, and D. Nardi. A probabilistic approach to Hough Localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2001)*, pages 4250–4255, 2001.

[20] L. Iocchi and D. Nardi. Hough trasform-based localization for mobile robots. In Nikos Mastorakis, editor, *Advances in Intelligent Systems and Computer Science*, pages 359–364. World Scientific Engineering Society, 1999.

[21] L. Iocchi and D. Nardi. Self-localization in the RoboCup environment. In *RoboCup-99: Robot Soccer World Cup III*, pages 318–330, 1999.

[22] L. P. Kaelbling, A. R. Cassandra, and J. A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'96)*, 1996.

[23] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara. Robocup: A challenge problem for ai and robotics. In *Lecture Note in Artificial Intelligence*, volume 1395, pages 1–19, 1998.

[24] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. Robotics and Automation*, 7(3):376–382, 1991.

[25] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18:249–275, 1997.

[26] C. F. Marques and P. U. Lima. A localization method for a soccer robot using a vision-based omni-directional sensor. In *RoboCup-2000: Robot Soccer World Cup IV*, 2000.

[27] Domenico Mastrantuono. Autolocalizzazione per robot mobili basata sulla trasformata di Hough. Master's thesis, University "La Sapienza" Rome, 2000.

[28] D. Nardi, G. Adorni, A. Bonarini, A. Chella, G. Clemente, E. Pagello, and M. Piaggio. ART-99: Azzurra Robot Team. In *RoboCup-99: Robot Soccer World Cup III.* Springer-Verlag, 1999.

[29] I. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2):53–60, 1995.

[30] Clark F. Olson. Probabilistic self-localization for mobile robots. *IEEE Transactions on Robotics and Automation*, 16(1):55–66, 2000.

[31] K. S. Premi and C. B. Besant. A review of various vehicle guidance techniques that can be used by mobile robots or AGVS. In *Proc. of 2nd International Conference on Automated Guided Vehicle Systems*, 1983.

[32] B. Schiele and J. Crowley. A comparison of position estimation techniques using occupancy grids. *Robotics and Autonomous Systems*, 12:163–171, 1994.

[33] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95)*, 1995.

[34] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2000)*, pages 1023–1029, 2000.