Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea • May 21-26, 2001

A probabilistic approach to Hough Localization

Luca Iocchi, Domenico Mastrantuono, Daniele Nardi
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza", Italy
{iocchi,mastrant,nardi}@dis.uniroma1.it

Abstract

Autonomous navigation for mobile robots performing complex tasks over long periods of time requires effective and robust self-localization techniques. In this paper we describe a probabilistic approach to self-localization that integrates Kalman filtering with map matching based on the Hough Transform. Several systematic experiments for evaluating the approach have been performed both on a simulator and on soccer robots embedded in the RoboCup environment.

1 Introduction

Self-localization is a crucial feature for autonomous navigation of mobile robots performing complex tasks over long periods of time. Indeed several practical application domains require mobile robots to know their position within the environment, in order to effectively and reliably accomplish their tasks.

The use of a particular kind of sensor usually affects the design choices for the localization method. A typical configuration for a mobile robot is having a *relative positioning system* (e.g. motor encoders), which provides an estimate of the displacement of the robot from the previous pose, and a *range sensor* (e.g. ultrasonic sonars, laser range finders, vision systems), 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 configuration is suitable for applying localization methods that are based on model matching or map matching (see [2] for a survey).

Among the several existing methods for robot self-localization, map matching has been extensively studied in the past years and the proposed approaches 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 includes algorithms performing map matching by using all the points captured by the sensor device, without any geometrical assumption on these data. A common feature for these approaches is that matching tends to be computationally hard, and in some cases the proposed methods require heavy optimization to implement an effective real-time localization task on a mobile robot. In [5] 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. In the Markov Localization described in [4] there is an explicit representation of the probability distribution of the robot's pose in the environment and every sensor points is used for updating this distribution according to the reference map. The second group of methods make use of geometric features instead of raw points: therefore they requires a preprocessing in order to extract features (or natural landmarks) from the sensor data. Most of these methods deals with lines, segments, corners and the reference map is thus represented as a set of these features. The main drawback of these kind of methods is that they rely on the availability of features in the environment.

The probabilistic approach to self-localization [4] is based on estimating the most likely pose of the robot given all the information on the environment coming from the sensor devices. The task is usually performed by matching range data against a given reference model (a map) of the environment, in order to determine the absolute pose of the robot in this map. Information coming from map matching are then integrated with odometric information in order to increase the reliability and the precision of localization.

In this paper we describe an approach to self-localization in which map matching is performed in the domain obtained by applying the Hough Transform to range data. This method, originally developed for the RoboCup games [6], applies to any environment that can be represented by a set of segments (polygonal environment) and it provides for a solution to the position tracking problem, assuming that the robot has at every time an initial guess of its pose. The work described in [7] follows our approach using an omnidirectional camera installed on the robot. In this setting, since the robot is able to see at every time all the environment in which it must act, the method provides for a global localization.

The robotic soccer environment provided by the RoboCup organization [1] is an interesting setting for testing solutions for self-localization, and in particular in the F-2000 League, where global positioning sensors are not allowed and thus localization can be based only on sensors that are mounted on the robot. We have successfully tested this method in the RoboCup environment within the ART team [8] during the official competitions by making use of vision based line extraction procedures performing as a range data sensor. The main features of the method are robustness and reliability in a very dynamic environment. The contribution of this paper is twofold: first we extend the approach described in [6] by introducing a probabilistic viewpoint that allows us to address the problem of integrating the Hough map matching process with odometric data; second, besides the good results of the method demonstrated during the official games, we have developed a set of systematic experiments that are described in details in section 4.

2 Hough Transform

In this section we present the Hough Transform and highlight the properties that will be useful for developing our localization method.

The Hough Transform is a robust technique for finding lines fitting a set of 2D points [3]. 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 $P(x, y)$ 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:

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

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.

The Hough Transform has a number of properties that are useful for self-localization: 1) given a set of input points, a local maximum of $HT(\theta, \rho)$ corresponds to the best fitting line of these points; 2) in presence of points originally belonging to several lines, no clustering is needed since local maxima of $HT(\theta, \rho)$ correspond to the best fitting lines for each subset of points relative to each line; 3) 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); 4) measuring displacement of lines in the Cartesian plane corresponds to measuring distance of 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 (we assume $\mid \theta_R \mid \leq \pi$), the Hough Transform of S with respect to the new pose of the robot will be $HT^S(\theta', \rho')$ such that:
if $0 \leq \theta + \theta_R < \pi$ then

$$\begin{aligned} \theta' &= \theta + \theta_R \\ \rho' &= \rho + T_x\,cos(\theta + \theta_R) + T_y\,sin(\theta + \theta_R) \end{aligned}$$

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

$$\begin{aligned} \theta' &= \theta + \theta_R - \pi \\ \rho' &= -(\rho + T_x\,cos(\theta + \theta_R) + T_y\,sin(\theta + \theta_R)) \end{aligned}$$

if $\theta + \theta_R < 0$ then

$$\begin{aligned} \theta' &= \theta + \theta_R + \pi \\ \rho' &= -(\rho + T_x\,cos(\theta + \theta_R) + T_y\,sin(\theta + \theta_R)) \end{aligned}$$

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. 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.

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

3 Hough Localization

In this section we introduce the Hough Localization method, obtained by developing the framework proposed in [6]. Hough Localization is based on a matching between the Hough representation of a known map of the environment and a local map built by the robot's sensors.

The task of estimating the most likely pose of the robot in the environment can be addressed by 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 computing 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 $\mathcal{M}$).

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.

The overall 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 predicted and the actual 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 is usually adopted when a *bounded error assumption* can be reasonably made. This assumption means that the positioning error of the robot is within a certain threshold, that usually depends on the characteristics of the environment. For example in the RoboCup environment these thresholds can be up to 50 cm and 45 degrees, and this allows the robot to deal with some of the collisions taking place during the games.

In the general case, in which it is not possible to rely on any information about the current position of the robot (*global localization*), a different technique must be integrated with the Hough Localization. In [6] we describe specific solutions for global localization in the RoboCup environment based on landmark recognition and active localization.

3.1 Position tracking

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 the variance of this information.

Under the bounded error assumption, the line correspondence problem can be easily addressed by adopting a closest matching approach. Given a reference point $(\theta^{\mathcal{M}},\rho^{\mathcal{M}})$ and a local maximum of HT^S (θ^S,ρ^S), a match will be considered if and only if $(\theta^{\mathcal{M}}-\theta^S)^2+(\rho^{\mathcal{M}}-\rho^S)^2<\delta$.

In other words, the HT grid can be partitioned in a number of regions (one for each reference point in $\mathcal{M}$),

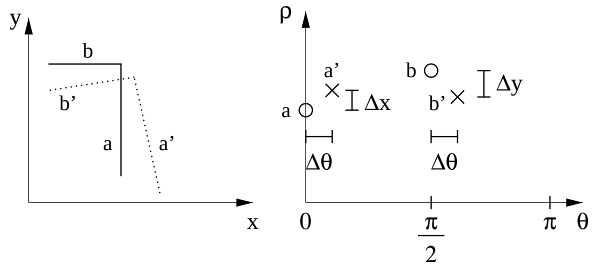


Figure 1: Map matching in the Hough domain

such that a matching will be considered only if a local maximum of HT^S is within the corresponding region.

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. 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 bounded error assumption, the correspondence problem is solved in this case by assigning a' to a and b' to b. For the Property 1 described in section 2, the displacement between the estimated and the actual pose of the robot is determined by first computing the orientation $\Delta\theta$ (with ρ constant) and then the translation Δx, Δy (with θ constant).

The computational complexity of the map matcing process is $O(n\,k)$, where n is the number of points returned by the range sensor and k is the number of segments in the map $\mathcal{M}$. Indeed, since a local search around each reference point is performed, HT^S is computed only in a limited region around the reference points. This complexity bound makes the method suitable for real time implementation (typical computation time is below 10 ms on a Pentium CPU).

3.2 Integrating map matching and odometry

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 [5].

We can describe the dynamics of the robot, with internal state $l_k = (x_k, y_k, \theta_k)^T$, input from odometry $u_k = (\delta_k, \alpha_k)^T$ and output $z_k = (\hat{x}_k, \hat{y}_k, \hat{\theta}_k)^T$, like follows

$$\begin{aligned} l_{k+1} &= l_k + B_k u_k + W_k w \\ z_k &= l_k + v \end{aligned}$$

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.

$$\begin{aligned} l^-_{k+1} &= l_k + B_k u_k \\ P^-_{k+1} &= P_k + W_k Q_k {W_k}^T \end{aligned}$$

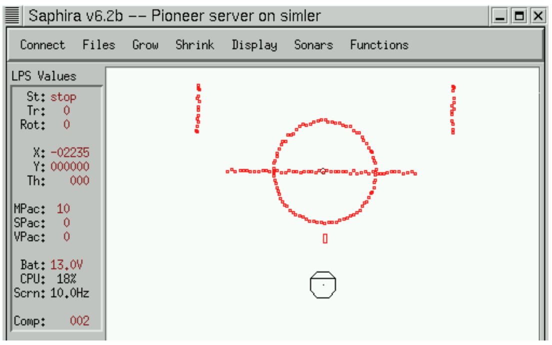
2. Correction. The pose of the robot is corrected by the result of the map matching process. Indeed z_{k+1} represent the new pose of the robot according to map matching.

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

As we will see in section 4, extended Kalman filter provides for an improvement in precision and stabilization of the robot's pose.

4 Experiments

In this section we describe several experiments for evaluating the precision and the robustness of the Hough Localization: first experiments that make use of a simulator in which we could test the effectiveness of the approach under controlled conditions, and then experiments with real robots. The accuracy of a localization method usually depends on the precision of the range sensor. If we consider an ideal range sensor, the noise introduced by the Hough method is due only to the discretizazion of the Hough grid. Therefore, the grid intervals $\delta\theta$ and $\delta\rho$ characterize the accuracy of the Hough localization method itself and must be tuned according to the precision of the range sensor.


Saphira v6.2b -- Pioneer server on simler
Connect Files Grow Shrink Display Sonars Functions
LPS Values

```
St: stop
Tr:    0
Rot:   0

X: -02235
Y: 000000
Th:   000

MPac: 10
SPac:  0
VPac:  0

Bat: 13.0V
CPU: 18%
Scrn: 10.0Hz

Comp:  002
```

Figure 2: Vision range simulator

4.1 Experiments with a simulator

We are using a simulator for mobile robots that includes a mathematical model of several different errors that occurs during robot navigation and sensor perception. In particular, because of our application oriented to robotic soccer, we emulate a vision range sensor that is able to extract points belonging to the lines in the RoboCup game field (see Fig. 2)[1].

The following errors are considered in the simulator: 1) *odometric error*: the position of the robot is affected by a random noise, such that the position error increases over time; 2) *sensor noise*: sensor data are affected by a random noise that increases with the speed of the robot; 3) *systematic error*: sensor data are affected by a systematic error that corresponds to usual errors in the calibration of the camera; 4) *robot bumps*: random movements of the robot; 5) *false positives and occlusions*: points that do not belong to a line and occlusions due to objects (other robots) that are in the field.

The models of the environments considered in our experiments are formed by sets of segments: in the RoboCup environment we have segments representing the boards and the lines drawn in the field, and one circle drawn in the field; in an office environment segments represent walls of the corridors. These segments are represented as points in the Hough domain for lines, and the circles are represented as points in the Hough domain for circles. Observe that the walls are real obstacles for the robot, while the other lines and the circle are drawn in the field and do not correspond to obstacles. However, the vision range sensor that we are using on our robots [6] is able to extract range information from both of them and thus we consider them also in the simulator.

With the use of a simulator it is possible to know exactly the *actual pose* of the robot at every time and to evaluate the position error as the difference between the actual position and the estimated one. In Fig. 3 we display two typical results of our experiments: in the first experiment (Fig. 3a) we have considered only odometric errors, while in the second one (Fig. 3b) three bumps of the robot have been simulated. The three lines on the graphs (Fig. 3a, 3b) represents the odometric error (red), the error with the Hough Localization and without the Kalman filter (blue), and the error with the Hough Localization and the Kalman filter (bold green). The temporal analysis shows that: 1) Hough Localization provides an upper bound to the localization error, while odometric error generally increases over time; 2) odometric error increases smoothly, while Hough Localization updates the robot's position sharply; 3) the use of a Kalman filter provides for smoothing the robot position updates, while keeping the bound on the localization error.

4.2 Experiments with the robots

Hough Localization has been implemented in our robots by making use of a vision based range sensor (see [6] for details on this sensor) in two different real environments: during the official RoboCup soccer competitions and within the corridors of our lab.

A first qualitative evaluation of the method has been performed during the RoboCup games, by using a monitor displaying the robot pose and by a visual inspection of its position in the field and the estimated position. Furthermore, we have performed more systematic experiments for evaluating the precision of the Hough Localization.

The first kind of experiments follows a classical approach [5]. We chose a number of reference positions in the environment, then we drove the robot on a path and we measured several times the distance between the actual position of the robot and its internal estimation. The results in evaluating the position error in the two operation fields are summarized as follows: average = 13cm, maximum = 29cm, variance = 8cm.

The above procedure attempted to measure an average precision of the self-localization method, but such values are not fully adequate to evaluate our method since they do not consider many sources of errors arising in the actual operation (collisions with obstacles, occlusions) and they may depend on various experimental conditions: type of range sensor, noise in the environment, choice of the path, robot velocity, etc. Moreover, since the samples are acquired only in predefined positions, and when the robot is not moving, it is not possible to monitor the effect of the localization method during robot navigation. For proving also the

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

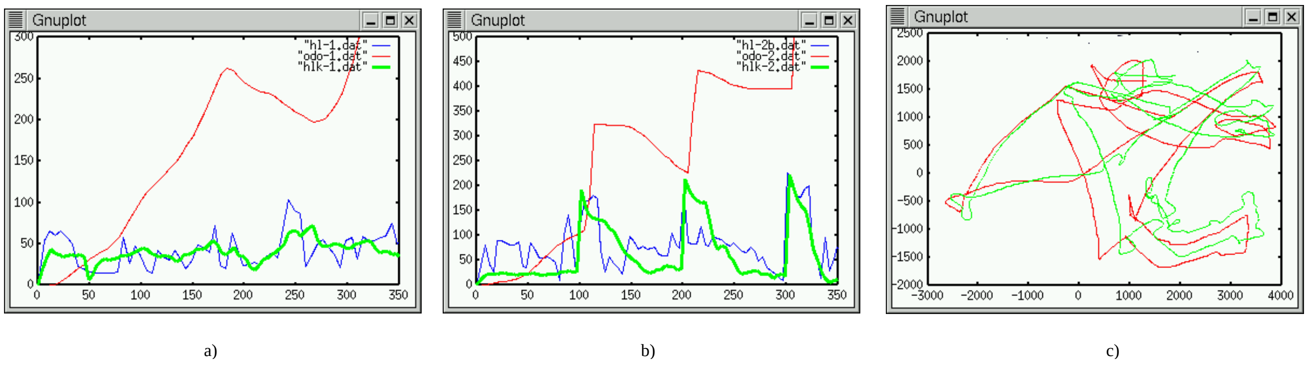


Figure 3: Position error: a) simulator without bumps, b) simulator with bumps, c) real robot.

robustness of a localization method we need a more detailed analysis of the robot position error during the execution of its tasks. We have thus implemented a global vision system, that makes use of a fixed camera positioned outside the game field for measuring the actual position of the robot. 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.

In this setting, as in the experiments with the simulator, we are able to monitor the robot's position error during navigation and thus evaluate the robustness of the self-localization under real conditions. In Fig. 3c) we show a representation of two trajectories computed during a normal activity of one of our soccer robots. The green trajectory has been computed by the internal localization method of the robot, while the red one has been computed by the tracking system of the global vision device. Note that this setting has not been used for evaluating the accuracy of the method, since it is affected by the errors of the global vision system in computing the position of the robot (the average position error of the global vision system is about 16 cm). Instead it is very useful for evaluating the robustness of our localization method, in fact we can show that the position error is always limited within a certain threshold.

5 Conclusion

Hough Localization presented in this article is based on a geometric representation of the reference map: lines and circles. This representation is suitable for the RoboCup environment, but also in office-like environments the availability of straight walls is usually guaranteed. With respect to other map matching techniques, the advantages of using the Hough Localization are: 1) it is computationally efficient, since position tracking is linear in the number of sensor points; 2) the Hough Transform and thus the line extraction process is very robust to occlusions and false positives.

The probabilistic approach to self-localization has been essential for an optimal integration of a robust map matching process based on the Hough Transform and odometric information. The experiments we have performed have proven that the use of an Extended Kalman Filter is relevant for both reducing the overall localization error and for smoothing the position updates.

References

[1] M. Asada. The RoboCup physical agent challenge: Goals and protocols for Phase-I. In H. Kitano, editor, *Robocup-97: Robot Soccer World Cup I*, 1998.

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

[3] R. Duda and P. Hart. Use of the Hough Transformation to detect lines and curves in the pictures. *Comm. of the ACM*, 15(1), 1972.

[4] D. Fox, W. Burgard, S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research* 11, 1999.

[5] J. S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *International Conference on Intelligent Robots and Systems*, 1998.

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

[7] C. F. Marques and P. U. Lima. A localization method for a soccer robot using a vision-based omni-directional sensor. In *Proc. of 4th International Workshop on RoboCup*, 2000.

[8] D. Nardi, et al. ART: Azzurra Robot Team. In *RoboCup-99: Robot Soccer World Cup III*, 1999.