

# Robust Mobile Robot Localisation from Sparse and Noisy Proximity Readings using Hough Transform and Probability Grids

Axel Großmann and Riccardo Poli

*School of Computer Science, The University of Birmingham,  
Birmingham, B15 2TT, UK*

---

## Abstract

We present a robust position tracking method for a mobile robot with seven sonar sensors. The method is based on Hough transform and probability grids. The focus of the paper is on the problem how to handle sparse sensors and noisy data in order to develop a low-cost navigation system for real-world applications. The proposed method consists of three steps. It computes a two-dimensional feature space by applying a straight-line Hough transform to the sonar readings. The detected features are then matched with the world map as reference pattern. The correlation counts obtained in the previous step are used for updating the position probability grid. We demonstrate that this method, on the one hand, avoids the common problems of feature detection in sonar data such as erroneous lines through separate clusters, corner inference, and line artifacts through reflection. On the other hand, it achieves a robustness that dense sensor matching techniques, such as Markov localisation, can only deliver if they use a complex sensor model which takes into account the angle to the object reflecting the sonar beam.

---

## 1 Introduction

We are interested in building mobile robots that accomplish useful tasks without human intervention in real-world environments. In most applications, a mobile robot must be able to determine its position and orientation in the environment using its sensors. The problem of position determination, referred to as localisation problem, can be seen as a constituent part of the more general navigation problem, in which the mobile robot has to find a path to a specified goal position.

In the last ten years, a large number of approaches to robot localisation have been proposed. We can distinguish between behaviour-based [1], landmark or

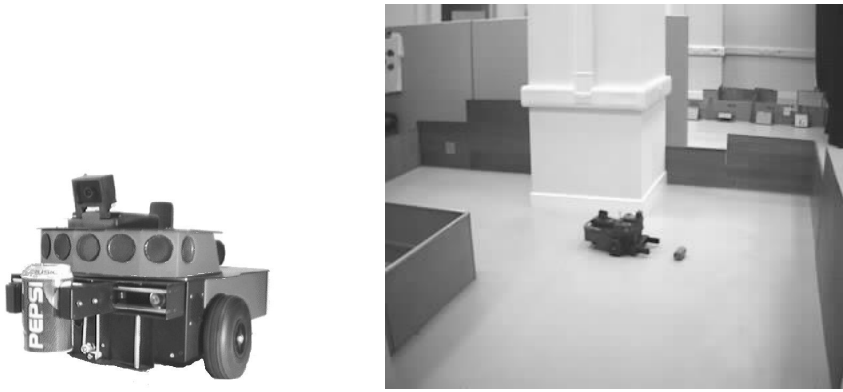


Figure 1. The Pioneer 1 mobile robot and its test environment.

feature-based [15], and dense sensor matching techniques [9]. Behavioural approaches use history information about the robot's interaction with the environment. Feature-based methods rely on the detection of landmark features, such as corners and wall-segments. Markov localisation and scan matching form the third category. The majority of existing localisation methods are passive, that is, independent of the robot control. Recently, also active approaches to localisation have been proposed [3]. Active localisation methods are able to change the robot motion and the orientation of the sensors in order to increase the efficiency and the robustness of localisation.

The robot's position and orientation, also referred to as pose, have to be determined from sensor data. Unfortunately, robot sensors are generally imperfect and their readings normally contain noise. In addition, the readings can be ambiguous, i.e., the environment may contain places which cannot be distinguished. We believe that for a localisation method to be reliable, it must use a representation capable of handling uncertain and ambiguous information. Among the possible forms of representation, we can distinguish between geometric representations, such as hypotheses about landmark features [7], and grid-based representations, such as position probability grids [4]. Both representations have interesting and useful features.

The localisation problem can be divided into two subproblems: the estimation of the absolute position in the environment, usually referred to as absolute localisation, and the tracking of the robot's position relative to a given starting point, commonly referred to as position tracking. The aim of position tracking is to correct the accumulated dead-reckoning errors caused by the inherent inaccuracy of the information provided by the wheel position encoders.

The work reported in this paper was motivated by the need to build a position tracking system for the Pioneer 1, a mobile robot platform manufactured by Real World Interface. The Pioneer 1 is equipped with seven sonar sensors, one on each side and five forward facing (see Figure 1). In comparison to mobile

robots that have a ring of sonars, the sensing capabilities of the Pioneer 1 are rather limited. In our experimental setup, the robot had to collect objects in an office-like environment. The pose of the robot was required as input to a learning algorithm which controlled the robot motors and the gripper [8]. At first, we tried to use a Markov localisation method [4] reported to be successful on mobile robots equipped with sonar sensors. However, this approach failed. The robot became lost when the sonar sensor readings were sparse and noisy, for example, when the robot was moving diagonally through a corridor. In this situation, the walls of the corridor reflect the sonar beams and hardly any distance readings from the front sonars are correct.

In further investigations, we developed the new localisation method described in this paper that works reliably in our setup. The approach consists of three steps. First, we compute a two-dimensional feature space by applying a straight-line Hough transform [14] to the sonar readings. Second, we perform template matching in the feature space by using the world map as reference pattern. Third, we use the correlation counts obtained in the previous step to update a position probability grid. This novel approach to localisation is unique in that it combines the detection of landmark features and matching for sets of features with probability-grid updates.

The paper is organised as follows. In Section 2, we critically examine related localisation methods and discuss their strengths and weaknesses for the given application. In Section 3, we introduce the Hough transform and describe the problems occurring when it is used alone to detect walls, corners, and edges from sonar data. In Section 4, we explain how these problems can be avoided by matching features in the Hough space. The algorithm is summarised in Section 5. An experimental evaluation of the approach is given in Section 6. Finally, we draw some conclusions in Section 7.

## 2 Localisation using ultrasonic sensors

### 2.1 *The problems of sonar sensing*

In order to estimate the robot's position reliably, we look for a localisation method which exploits as many features of the environment as possible. Since only those features can be exploited that are visible to the robot's sensors, the effectiveness of the position estimation depends very much on the interaction between sensors and environment.

In this paper, we are interested in using proximity information. The problem is that ultrasonic sensors do not measure proximity. Instead, they measure

the time elapsed between emitting and receiving a focused sound impulse. For smooth objects, the reception of the sonar echo depends on the angle between the main sonar cone and the reflecting object. The sound waves that hit an object frontally are very likely to reflect back in the direction of the sonar sensor, whereas waves that hit an object at an acute angle are likely to be reflected away in a direction where they cannot be detected. This situation is called total reflection [15]. In other cases, the sound waves are not reflected directly back to the sensor, but will undergo multiple reflections before they are received. Those are usually referred to as specular reflections [15]. Due to the effects above, only some of the sonar readings represent proximity. In addition, there is uncertainty on which part of the beam was actually reflected.

To overcome the effects of reflections and specularities, it is very common to use redundant readings. The most important sources of redundancy are alignment and mobility [6]. The former refers to the fact that the readings which are due to specular reflection rarely align. We can therefore use points that are aligned for the detection of object surfaces. The latter refers to the possibility of taking sonar readings from different observation points. As the viewpoint changes, incorrect distance readings due to specular reflection project onto widely varying surfaces, while correct readings corresponding to an object surface project onto that surface. Therefore, if objects have smooth surfaces, in theory it is possible to identify incorrect distance readings. In practice, this turns out to be very difficult.

Ideally, we want to obtain distance information in all possible directions. This is possible if the robot is equipped with a ring of sonars or a single rotating sensor. In this respect, the capabilities of the Pioneer 1 mobile robot are rather limited. Its seven sonars are mounted at positions of  $0, \pm 15, \pm 30$  and  $\pm 90$  degrees with respect to the heading of the robot, each sensor having a beam angle of about 25 degrees. The sonar readings are accumulated in three buffers, the front and two side buffers (left and right). The front buffer accumulates one reading each time one of the front sonars is fired, regardless of whether an echo is received or whether the robot has changed its position. The two side buffers accumulate sonar readings only when a side sonar actually receives an echo and the robot moves. The size of the front buffer is 20 readings. The two side buffers contain 40 readings each.

As the view of the robot is restricted to the front and sides, visibility and appearance of features in the environment change with the orientation of the robot. Figure 2 demonstrates how the robot's behaviour affects the visibility of walls. If the robot travels through a corridor along a straight line keeping a constant distance from both sides, then the walls of the corridor are clearly visible in the sonar data. However, if the direction of movement is changed repeatedly, for example, when the robot is trying to pick up an object, then the perception of the walls degrades to a set of non-connected line segments.

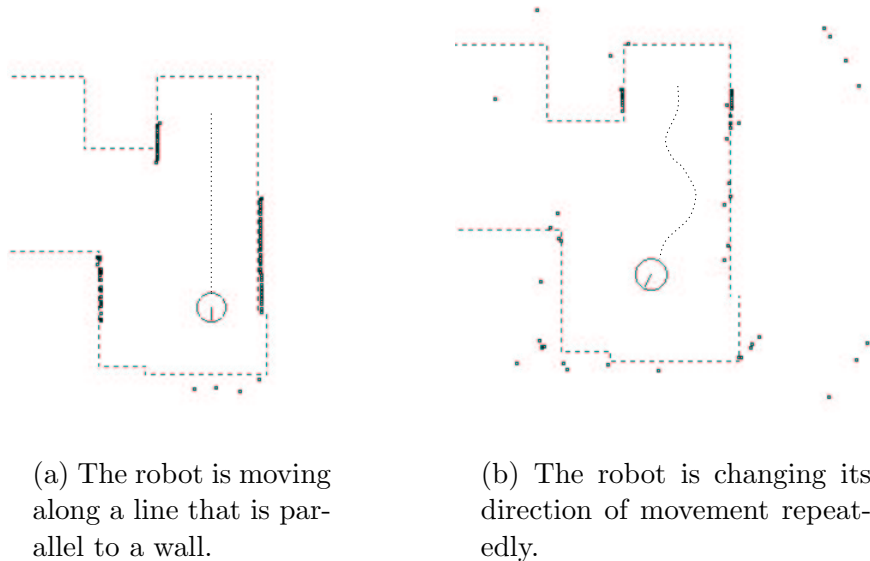


Figure 2. Sonar sensor readings of a Pioneer 1 mobile robot operating in an indoor environment.

Considering this observation, there seem to be two alternatives: the localisation method should not rely on the detection of individual landmark features, or the localisation method should be able to deal with the uncertainty inherent in the detection of landmarks.

## 2.2 Dense sensor matching techniques

Recently, dense sensor methods have been proposed that do not rely on the recognition of landmarks. Instead, they attempt to use whatever sensor information is available. Examples of dense sensor methods are Markov localisation [4], which uses a probability distribution across a grid of robot poses, and scan matching [9], which uses Kalman filtering techniques based on matching sensor scans. In the following paragraph, we describe the method in [4] in more detail, both to illustrate the importance of each method’s assumptions in determining its reliability and to provide some background for our extension of this method presented in Section 4.3.

Burgard *et al.* [4] proposed a Markov localisation method that has been applied successfully in several real-world environments. The method is based on a probabilistic representation that can handle ambiguities and represent degree-of-belief. In particular, they use a position probability grid  $P$  to accumulate in each cell the probability  $p(L)$  that this cell refers to the current pose,  $L$ , of the robot. The grid  $P$  is updated according to Bayes’ formula. Suppose  $p(L)$  is the prior probability that  $L$  is the current pose of the robot. Then the

(posterior) probability of  $L$  referring to the current pose given the new sensor reading,  $s$ , is defined as:

$$p(L|s) = \frac{p(s|L) p(L)}{\sum_{L' \in P} p(s|L') p(L')} \quad (1)$$

This equation is used as an update rule when only a single sensor reading is available. However, the robot receives a set of readings,  $(s_1, \dots, s_n)$ , at each time step. That is:

$$p(L|s_1 \dots s_n) = \frac{p(s_1 \dots s_n|L) p(L)}{\sum_{L' \in P} p(s_1 \dots s_n|L') p(L')} \quad (2)$$

Assuming the readings  $(s_1, \dots, s_n)$  to be statistically independent, the total probability of a set of measurements can be computed as:

$$p(s_1 \dots s_n|L) = \prod_{i=1}^n p(s_i|L) \quad (3)$$

and we can write the update formula as:

$$p(L|s_1 \dots s_n) = \frac{p(s_1|L) \dots p(s_n|L) p(L)}{\sum_{L' \in P} p(s_1|L') \dots p(s_n|L') p(L')} \quad (4)$$

The assumption about the conditional independence of percepts above is probably justified for a mobile robot equipped with a ring of 24 sonars as the one used by Burgard *et al.* – even if the sonar sensor model  $p(s|L)$  does not take into account the sensor’s angle to the object reflecting the sonar beam. For such a robot, we can assume that the measurements provided by the front sonars are statistically independent of the ones from the backward-pointing sonars. However, it remains questionable whether the independence of percepts assumption is correct for the Pioneer 1 robot, unless we take into account the angle of reflection. Suppose that at the current location, the reading of the forward pointing sonar is falsified by multiple reflection, then it is likely that this is the case also for the adjacent sensors. That is the sensor readings are not conditionally independent. Indeed, we re-implemented the position tracking approach by Burgard *et al.* [4] and found that the robot was likely to lose track of its position in situations similar to the one in Figure 2(b), when the sonar readings were sparse and noisy.

Grid-based Markov localisation methods are computationally difficult. The basic problem is that the probability update step (see Equation 4) involves comparing actual sensor readings to readings that would have resulted from having the robot at every possible pose in the given map. This update can be enormously expensive, even for moderately sized maps [12].

The use of a more complex sensor model, which takes into account the sensor's angle to the closest obstacle as well, might improve the performance of the Markov localisation methods. Whether this is possible in practice, however, depends on the internal representation of the world map. If the representation does not allow an efficient computation of the sensor's angle, e.g., using ray-tracing, the localisation method would become even more computationally expensive.

For office-like environments, it is possible to pre-compute the values  $p(s|L)$ . For example, Burgard *et al.* [5] proposed to store for each possible state of the robot the distance which is expected to be measured by the sensor combined with the expected standard deviation. Similarly, the angle of the sensor to the closest obstacle could be stored as well. Please note that this method does not scale up to large environments.

### 2.3 Feature-based localisation techniques

In our application, we do not want to rely on active control to overcome the problems caused by the limited sensing abilities of the robot. Neither we want to use target tracking methods, which rely on complex, prior information about the environment such as the surface reflectance and geometry of the objects. Instead, we focus on landmark-based localisation techniques that correlate simply-shaped objects such as walls, corners, and edges in the sonar data against a given map of the environment. In this section, we briefly review previous research in this area.

Drumheller [7] proposed a feature-based localisation algorithm for mobile robots equipped with a sonar range-finder. He introduced the idea of sonar segments, which are straight-line segments extracted from the measurements obtained in a 360-degree sweep of the sonar range-finder. Moreover, he used a sonar barrier test, which eliminates implausible robot positions by exploiting the physical constraint that sonar beams do not penetrate solid objects. However, the algorithm is not able to exploit readings from different positions. Crowley [6] and others proposed the use of this redundant information to identify erroneous distance readings. In Crowley's approach, a sonar segment is not formed unless three consecutive distance readings align with a certain tolerance.

As discussed in Section 2.1, we encountered the problem that a single wall is not perceived as single landmark but as set of non-contiguous segments. The approaches by Drumheller [7] and Crowley [6] do not perform well in this situation.

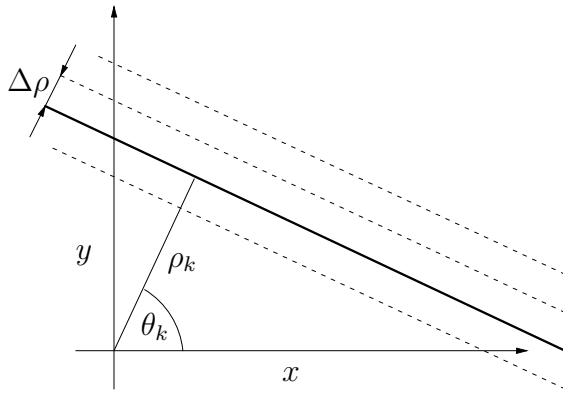


Figure 3. Parameters of a line and quantisation of the Hough space.

### 3 Detecting walls and corners with the Hough transform

The Hough Transform (HT) is well-known in computer vision as a shape detection method [2,14]. In general, its purpose is to detect parametric curves in sets of primitive feature points. It has the advantage of being relatively unaffected by gaps in curves and noise. So it turns out that the Hough transform has the right features to be used for the detection of straight line-segments in sonar data.

#### 3.1 The Hough transform

We consider the straight-line Hough transform. This maps straight lines in the input data to points in the HT space. Each point in the HT space has a strength measure associated with it. The strength of a point in the HT space is proportional to the length and width of the corresponding line in the input space.

A straight line in the two-dimensional  $x$ - $y$  coordinate plane (input space) can be described by the equation  $\rho = x \cos \theta + y \sin \theta$  where  $\rho$  is the shortest distance between the origin and the line, and  $\theta$  is the angle between the line normal and the positive  $x$ -axis (see Figure 3). We assume the origin of the  $x$ - $y$  coordinate system to be in the centre of the input space. We can limit the line angle  $\theta$  to  $0 \leq \theta < \pi$ , if we allow negative values of  $\rho$ . In the  $\theta$ - $\rho$  parameter plane (HT space), the line is mapped to a single point. Given a particular point in the input space, there are infinitely many lines passing through this point. The parameters of all lines going through a point,  $(x_i, y_i)$ , in the input space constitute a sinusoidal curve in the  $(\theta, \rho)$  HT space, given by

$$x_i \cos \theta + y_i \sin \theta = \rho . \tag{5}$$

For an illustration of this transformation, see Figure 4.

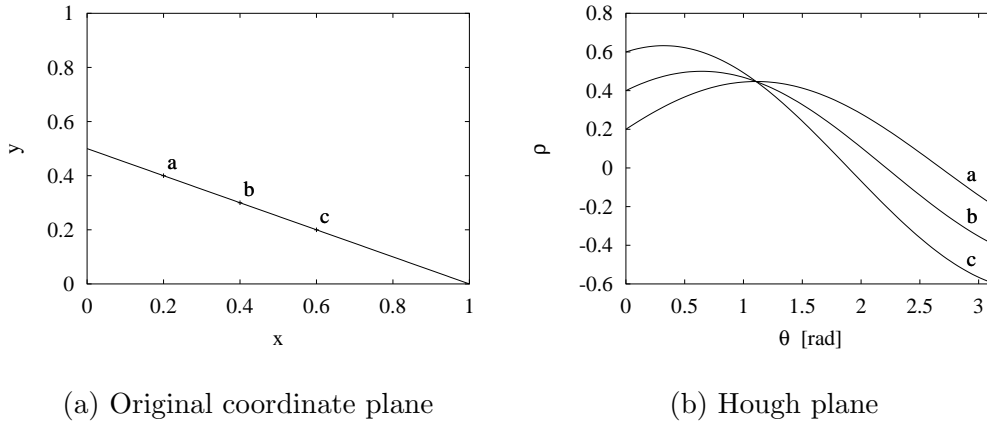


Figure 4. Transformation of  $x$ - $y$  points into sinusoidal curves in the HT space.

The implementation of the Hough transform is based on parameter discretisation. If the values of  $\theta$  and  $\rho$  are discrete, we can form an accumulator array,  $A(\theta, \rho)$ , whose elements are initially zero. The parameter  $\theta$  is quantised in values  $\theta_k$  with  $k = 1, \dots, n$  such that  $\theta_k - \theta_{k-1} = \Delta\theta$ . The Hough transform of a feature point,  $(x_i, y_i)$ , is performed by computing  $\rho$  using Equation 5 for all  $n$  values of  $\theta_k$ . The values of  $\rho$  are then quantised in  $m$  discrete values  $\rho_k$  with  $k = 1, \dots, m$  such that  $\rho_k - \rho_{k-1} = \Delta\rho$ , and the corresponding cells  $A(\theta_k, \rho_k)$  are incremented. This procedure is repeated for all feature points. Collinear feature points now show up as peaks in the accumulator array  $A(\theta, \rho)$ .

The Hough transform as described above can be applied to sonar sensing by using the two-dimensional distance readings as feature points. Recently, Yun *et al.*[18] found that one can use such a straightforward implementation to recognise wall-like features from noisy sonar data. However, the accuracy of detected features depends very much on the quantisation parameters  $\Delta\theta$  and  $\Delta\rho$ . Yun *et al.* used a robot equipped with a ring of 16 sonars. In our experience, a similar approach does not work reliably on a robot with very few sensors.

### 3.2 Compensating the error caused by parameter quantisation

In contrast to Yun *et al.* [18], we believe that the errors due to the parameter quantisation need to be considered and corrected. The quantisation resolutions,  $\Delta\theta$  and  $\Delta\rho$ , determine the mapping of input points to accumulator cells. This can be illustrated as follows. By quantising the values of  $\rho$ , the input space is divided completely into bar-shaped windows for constant values of  $\theta$  (see Figure 3). The smaller  $\Delta\rho$ , which is the width of the windows, the fewer points are included in a line estimation. The quantisation resolutions specify the spread of the points which are allowed in forming a line. If the Hough

transform is to be applied to sonar data, we cannot make the quantisation resolution arbitrarily small, since the input data is very noisy. Clearly, there is a trade-off between the accuracy in the position estimate and the reliability of the detection process.

The effect of the parameter quantisation on the Hough transform has been thoroughly studied in computer vision [17]. Straight-line features are identified by searching for local maxima in the accumulator array. The shape of the peaks is influenced by several factors, which include the quantisation of the input space, the  $\theta$ - $\rho$  parameterisation, and the width of the line segments. Hence, those factors need to be considered in a method that locates the peaks in the HT space.

The input space is formed by the two-dimensional distance readings stored in the sonar buffers. As the sonar buffers keep track of previous readings, the distribution of the input points depends not only on the position of the sensors but also on the movements of the robot. There are situations in which the sonar readings form dense clusters. We need to ensure that the feature points are sampled at a suitable mutual distance on the line segment. Multiple readings obtained from the same sensor and same position would bias the detection process. Therefore, we quantise the input space by superimposing a squared grid on the  $x$ - $y$  coordinate plane. If there is more than one sonar reading per grid field, only the first one is used as feature point for the Hough transform. The size of the cells,  $h$ , in this grid, is chosen to ensure  $h \leq \Delta\rho$ .

In general, a straight-line segment will not have a direction exactly identical to one of the values  $\theta_k$ . That is, the line segment will cross several parallel windows with a direction  $\theta_k$ . Hence, the peak in the HT space will be extended over several accumulator cells in the  $\rho$ -direction corresponding to those windows. To minimise the spread of the peak in  $\rho$ -direction,  $\Delta\rho$  is chosen so that

$$\Delta\rho \cong l \sin\left(\frac{1}{2}\Delta\theta\right) \quad (6)$$

for a given  $\Delta\theta$  and a maximum length  $l$  of the line segments [17].

Up to this point, we have assumed that feature points lie exactly on line segments. However, this does not hold for sonar sensing because of the noise inherent in the sensors. We model the imprecision of the sonar measurements using the line width  $b$ . That is, we assume that the feature points lie exactly on a thick line rather than assuming that the line segment is infinitesimally thin.

A solution to the problems mentioned above is to filter the uncompensated accumulator values using a function of the values in a neighbourhood of each point. In the work described in this paper, we have used the error correction

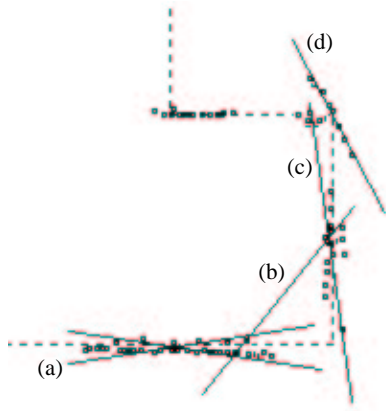


Figure 5. Problems in the detection of walls and corners using the Hough transform: (a) multiple local maxima, (b) erroneous lines through dense clusters, (c) corner inference, and (d) line artifacts through reflection.

method proposed by van Veen and Groen [17]. All points of a line segment are taken into account in the peak searching step by summing the values in  $n_\rho$  cells in the  $\rho$ -direction for each  $\theta_k$  value (moving average filtering) and looking for a maximum of this sum. Given the assumptions mentioned above, the maximum number of cells,  $n_\rho$ , of the peak in the  $\rho$ -direction is:

$$n_\rho = \left\lfloor \frac{\sqrt{l^2 + b^2} \sin\left(\frac{1}{2}\Delta\theta + \arctan\frac{b}{l}\right)}{\Delta\rho} \right\rfloor + 2 \quad (7)$$

in which  $\lfloor z \rfloor$  means the largest integer smaller than  $z$ .

### 3.3 Other sources of errors

In this section, we discuss problems that remain when the Hough transform is used to detect straight-line segments in sonar readings. In contrast to the effects of parameter quantisation discussed before, these problems are specific to the domain of sonar sensing.

Because of the noise inherent in the sonar data, sharp peaks rarely occur, rather all peaks are distorted and diffused. Therefore, we encounter situations in which there is uncertainty in the detection process. Instead of a single line segment, the Hough transform can identify a set of lines, each line having slightly different parameters. An example is given in Figure 5(a). Although the error compensation method mentioned above can ease the problem by smoothing the accumulator array, multiple local maxima can still occur.

The Hough transform has the tendency to identify line segments with a maximum number of points, since the accumulator count is just the number of

points in the line. Therefore, two dense clusters of feature points can give rise to a single erroneous line. As shown in Figure 5(b), a line is detected between any two clusters if these clusters combined contain more points than any other line in the neighbourhood.

Also, the line parameters obtained in the Hough transform can be biased. For example, corners tend to interfere with the parameter estimation. As shown in Figure 5(c), the points belonging to the perpendicular segment alter the slope of the estimated line by pulling it to one side. This problem is particularly severe if we aim to detect small but straight line-segments.

As discussed in Section 2, sonar sensing is subject to reflections and specularities. Feature points due to multiple reflections can give rise to line segments for which there is no corresponding world segment (see Figure 5(d)). There is no straightforward way to detect these false peaks since the Hough transform ignores the relative position of lines. Peaks can be due to line segments in the input space that do not touch each other and that do not even lie near to each other.

Considering these detection errors, in particular the occurrence of line artifacts due to reflection, we conclude that a localisation method based solely on the detection of single features such as wall-like segments cannot provide the reliability required. We overcome this problem by taking constraints about the position of the world segments into account. As described in the next section, this can be done directly in the Hough space.

## 4 Matching in the Hough space

In the following, we assume that a map of the robot’s environment is available and that this world map can be represented as a list of straight-line segments,  $\mathcal{W} = \{w_0, w_1, \dots, w_n\}$ . For each world segment  $w_i$ , the start and end point is known. So, we can compute the corresponding set,  $\mathcal{R}$ , of angle-radius parameters  $(\theta, \rho)$  of all world segments. The idea is to match the sonar readings and the world map directly in the Hough space. As pointed out by Krishnapuram and Casasent [13], similar objects have similar Hough transforms and different objects have different Hough transforms. So we can use  $\mathcal{R}$ , the polar coordinates of the world map, as reference pattern and by trying several rotated and translated versions (templates) of this pattern, we can find the position where the reference pattern matches best the Hough transform of the current sonar readings. The templates can be computed from the reference pattern  $\mathcal{R}$  as follows.

#### 4.1 Rotation and translation in the Hough space

Let  $(\theta, \rho)$  be a point in the HT space corresponding to a line segment in the input space. If all the points in the input space are rotated by an angle  $d_\theta$ , it can be shown that the line segment would now map to a different point  $(\theta', \rho')$  in the Hough space given by

$$\begin{aligned} \rho' &= \rho & \text{and} & & \theta' &= \theta + d_\theta & & \text{if} & & 0 \leq \theta + d_\theta \leq \pi \\ \rho' &= -\rho & \text{and} & & \theta' &= \theta + d_\theta - \pi & & \text{if} & & \theta + d_\theta > \pi \\ \rho' &= -\rho & \text{and} & & \theta' &= \theta + d_\theta + \pi & & \text{if} & & \theta + d_\theta < 0 \end{aligned}$$

It can also be shown that if all the points in the input space are translated by  $(d_x, d_y)$ , the point  $(\theta, \rho)$  will now map to the point  $(\theta', \rho')$  given by

$$\rho' = \rho + t \cos(\theta - \alpha) \quad \text{and} \quad \theta' = \theta \quad (8)$$

where  $t = \sqrt{d_x^2 + d_y^2}$  and  $\alpha = \arctan \frac{d_y}{d_x}$ . These transformations are adopted from [13].

#### 4.2 Template matching

To apply template matching, we need to specify the search space, a set of reference patterns, and a correlation measure.

Let  $A$  be the accumulator array for performing the Hough transform on a set of sonar sensor readings obtained using the estimate  $(p_x, p_y, p_\theta)$  of the robot's pose. The HT space is quantised using the resolutions  $\Delta\theta$  and  $\Delta\rho$ . The last step of the Hough transform is an error compensation with  $n_\rho$ . A point  $(\theta_k, \rho_k)$  in the HT space corresponds to the accumulator cell  $A_{ij}$  with the index functions  $i = \lfloor \theta_k / \Delta\theta \rfloor$  and  $j = \lfloor (\rho_k - \rho_{min}) / \Delta\rho \rfloor$ .

For efficiency reasons, the matching should be performed directly in the Hough space by selecting the appropriate cells in  $A$ . Since the Hough transform is concerned with lines, not line segments, this straightforward approach fails in environments where parallel world segments are close to each other, or the angle between neighbouring world segments is small. In these cases, we have to take into account the individual points that contributed to a particular element of  $A$ . So we divide the line segments in  $\mathcal{W}$  into conflicting and non-conflicting world segments. A line segment  $a$  is said to conflict with a line segment  $b$  if  $|\theta_a - \theta_b| < d_\theta$  and  $|\rho_a - \rho_b| < d_\rho$ . We have used  $d_\theta = 10$  deg and  $d_\rho = 30$  cm. In addition, we do not consider world segments shorter than 50 cm.

The Hough transform,  $A$ , of the sonar readings is matched with several rotated and translated versions of  $\mathcal{R}$ . This search is performed for rotations  $d_\theta$  quantised in  $\Delta\theta$  intervals and for translations  $d_x$  and  $d_y$  quantised in  $\Delta\rho$  intervals to the degree required for the given situation. For each template, there is a corresponding displacement  $(d_x, d_y, d_\theta)$ . For each  $r \in \mathcal{R}$ , we compute a correlation value,  $c$ , given by

$$c(r) = \begin{cases} A_r(i', j') & \text{if } r \text{ are the coordinates of a conflicting world segment} \\ A(i', j') & \text{otherwise.} \end{cases} \quad (9)$$

with  $(i', j')$  being the image of  $r = (i, j)$  under the displacement transformation  $(d_x, d_y, d_\theta)$ .  $A_r(i, j)$  is the number of points that contributed to  $A(i, j)$  and that lie between the begin and end points of the segment that corresponds to  $r$ . Consequently, the correlation value for a template  $(d_x, d_y, d_\theta)$  is

$$c_{\mathcal{R}} = \sum_{r \in \mathcal{R}} c(r) \quad (10)$$

### 4.3 Position probabilities

After determining the template  $(d_x, d_y, d_\theta)$  that yields the maximum correlation value, we want to compute a new estimate of the robot's pose. If it is based solely on the current best match, such an estimate is still subject to ambiguity. For example, in situations where the sonar readings are sparse, it might not be possible to identify any world segment with confidence. It is also possible that the detected world segments have the same orientation. To remove these ambiguities, we decided to use the correlation values obtained by template matching to compute position probabilities. That is, we use a position probability grid on top of a feature-based localisation approach.

The crucial component in the update equation of Markov localisation as discussed in Section 2.2 is  $p(s|L)$ , the likelihood of observing the sensor reading  $s$  at the position  $L$ , which can be computed from the world map and a model of the sonar sensors [4]. We have already discussed that the sonar sensor model as it was used in previous applications of Markov localisation, which does not take into account the angle to the object reflecting the sonar beam, is not appropriate for mobile robots with a small number of adjacent sensors. We claim that Markov localisation can be made more robust by filtering out misleading sensor readings that are likely to be caused by multiple reflection. Instead of the individual sonar readings, we propose therefore to use the likelihood that a particular correlation count is observed at the location. We obtain the

following update formula:

$$p(L|c) = \frac{p(c|L) p(L)}{\sum_{L' \in P} p(c|L') p(L')} \quad (11)$$

The update formula requires a model for  $p(c|L)$ . In the initial experiments, we have assumed that  $p(c|L)$  is proportional to the value of  $c$ .

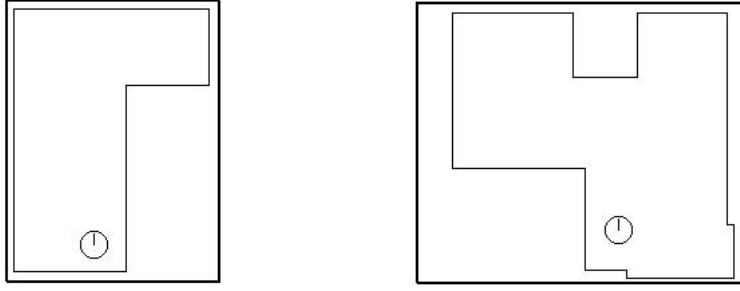
The values  $p(L|c)$  are stored in a position probability grid  $P$ . At the beginning, the probability grid is initialised using the *a priori* probability that the location that corresponds to the particular grid cell refers to the start position of the robot. An update is performed by multiplying the value in each grid field by the correlation count,  $c$ , obtained at the corresponding position,  $L$ . Then, we normalise  $P$ .

## 5 The position-tracking algorithm

The structure of the algorithm is similar to Kalman filtering [10], a popular technique for integrating sensor information over time. Kalman filtering uses two kinds of updates, which are often referred to as prediction and correction. In the prediction phase, the change of the state due to control actions is modelled. The correction phase corrects the state description by combining the estimation of the state with incoming sensor data.

The prediction step, which uses a model of the robot’s odometry sensors, is performed at every time step. However, the correction step, which uses the results of template matching in the Hough space, changes the probability distribution about the robot’s position only if there is sufficient evidence. We require that at least one sonar segment is detected per time step. If the pose cannot be corrected for a long time due to a lack of consistent sonar readings, we consider the odometry errors. In each time step where no position update is available, the position probability grid is updated using an error model of the odometry sensors. The algorithm can be summarised as follows:

- (0) Initialisation
  - Initialise the probability grid  $P$  for the given start position  $p = (x, y, \theta)$ .
- (1) Feature detection
  - (1.1) Pre-processing of sensor data
    - Get current estimate of the robot pose,  $p$ .
    - Get current sonar sensor readings and map them into the global coordinate system used for localisation.
    - Keep track of sensor readings in the front and side sonar buffers.



(a) Size 3.5m  $\times$  4.6m.

(b) Size 5.2m  $\times$  4.4m.

Figure 6. Laboratory environments used for the localisation experiments. The photograph given in Figure 1 shows the test environment (b).

(1.2) Detection of sonar segments

- Compute accumulator array  $A$  using the readings in the sonar buffers and perform error compensation on  $A$ .
- Obtain a set of sonar segments the accumulator count (number of feature points) of which is greater than  $A_{min}$ .
- If any such segments were found then perform both prediction (2) and correction (3) and continue with (1). Otherwise, perform prediction (2) only and continue with (1).

(2) Prediction

- Update the position probabilities in  $P$  using an error model of the odometry sensors.

(3) Correction

- Perform template matching in  $A$  at neighbouring positions of  $p$ .
- Update the position probabilities in  $P$  using the correlation values obtained during template matching.
- Determine the displacement  $(d_x, d_y, d_\theta)$  that corresponds to the maximum value in  $P$ .
- Update the estimate of the robot pose,  $p$ , at time step  $t$ , such that  $p^{(t)} = p^{(t-1)} + (d_x, d_y, d_\theta)$ .
- Match the detected sonar segments to segments in the world map (optional step).

## 6 Experiments

The experimental evaluation of localisation methods is difficult. The performance usually depends on the environment, the behaviour of the robot and the hardware [9]. We have tested our position tracking system in two laboratory environments. As shown in Figure 6, both environments consist of straight

wall segments. In particular, the second environment is difficult to navigate using sonar sensors due to the existence of many corners and edges.

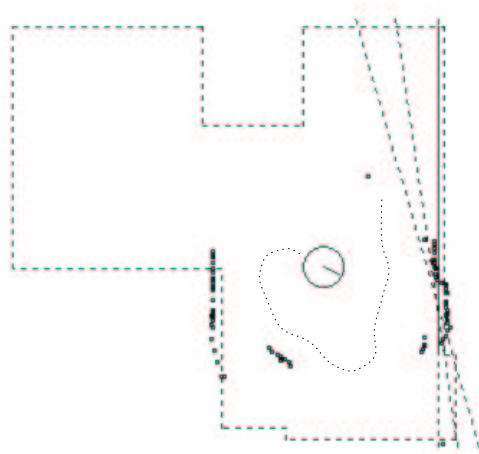
Position tracking experiments were performed using the Pioneer 1 mobile robot in two scenarios: a can collection task in which the robot had to pick up soda cans from the floor and take them to a collection point, and a navigation task in which the robot had to reach goal positions randomly chosen within the environment. The robot control program used in the second case made the robot constantly change its direction, which can be considered to be the worst-case scenario for a position tracking algorithm that relies on the detection of straight-line features in the sonar sensor readings.

The position tracking method described in this paper has only very few parameters that depend on the application. These are (1) the maximum length  $l$  of the world segments, (2) the desired angular resolution,  $\Delta\theta$ , and (3) the minimum number of sonar readings,  $A_{min}$ , required for a straight-line segment to be detected. We have chosen  $l = 70$  cm,  $\Delta\theta = 8$  deg, and  $A_{min} = 22$ .

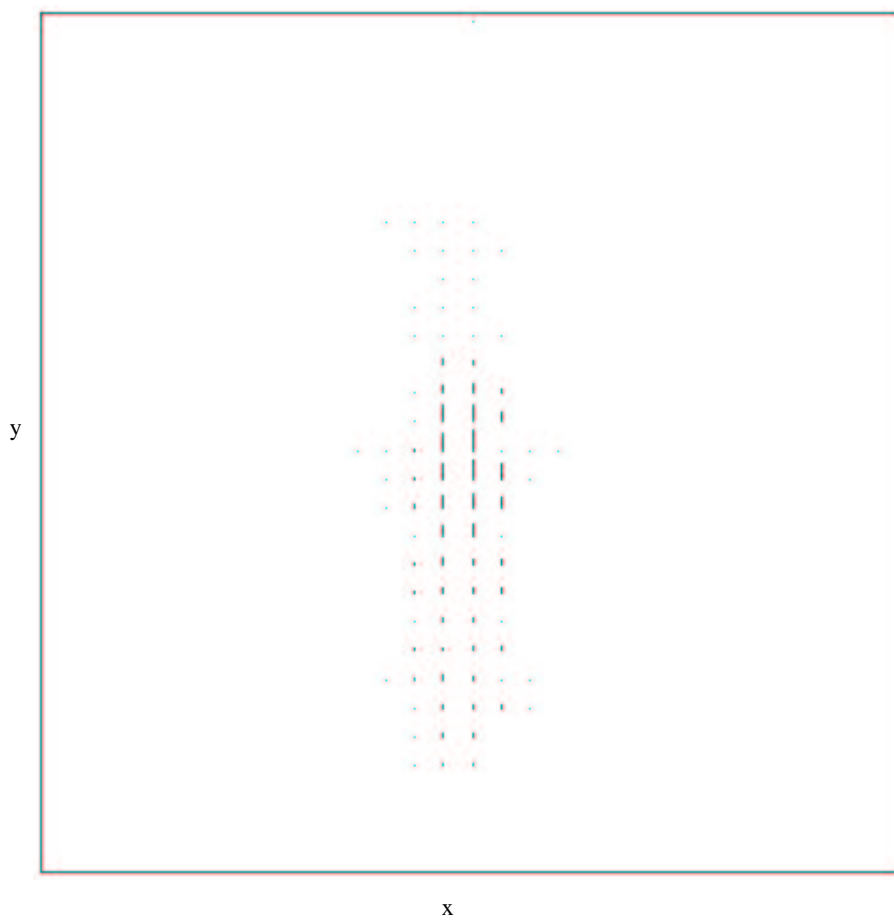
Clearly, the choice of  $l$  depends on the environment the robot operates in. But this setting is not critical as it influences mainly the effectiveness of the error compensation in the Hough transform. The choice of  $A_{min}$  depends on the number of sonar sensors, the size of the sonar buffers, and the average speed of the robot. If  $A_{min}$  is too small then many false line segments will be detected. If  $A_{min}$  is too large then short world segments will not be detected. We had no difficulties in finding an appropriate setting for  $A_{min}$ , by trying several values in a small number of test runs. The remaining parameters can be computed directly from  $l$  and  $\Delta\theta$  using the equations given in Section 3.2. In this way, we have obtained  $\Delta\rho = 5$  cm,  $h = \Delta\rho$ , and  $n_\rho = 5$ .

In long-term experiments, we found that the proposed method is able to keep track of the robot's position reliably. In the two scenarios, no catastrophic localisation failure was experienced at any time. The system behaved robustly even in situations when the quality of information received from the sensors was degraded over long periods of time. In addition to the two scenarios mentioned above, we have repeatedly used the position tracking system in experiments on robot learning, in which the robot was learning a navigation task over extended periods of time (2 to 3 hours). In this application as well, no catastrophic localisation failure was experienced at any time.

Figures 7 and 8 show situations from a position tracking experiment, demonstrating how the algorithm deals with uncertainty. The panel in the top shows the sonar sensor readings and the detected sonar segments with respect to the current estimate of the robot's pose,  $p$ . In the bottom part, you find a representation of the position probability grid,  $P$ .  $P$  is a function of three variables: the position in the  $x$ - $y$  plane and the robot's orientation,  $\theta$ . We represent the

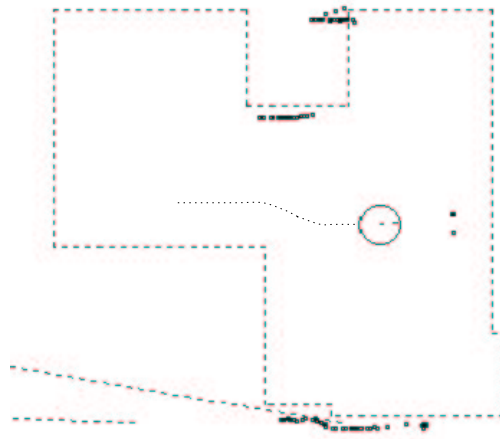


(a) Trajectory and perceptions.

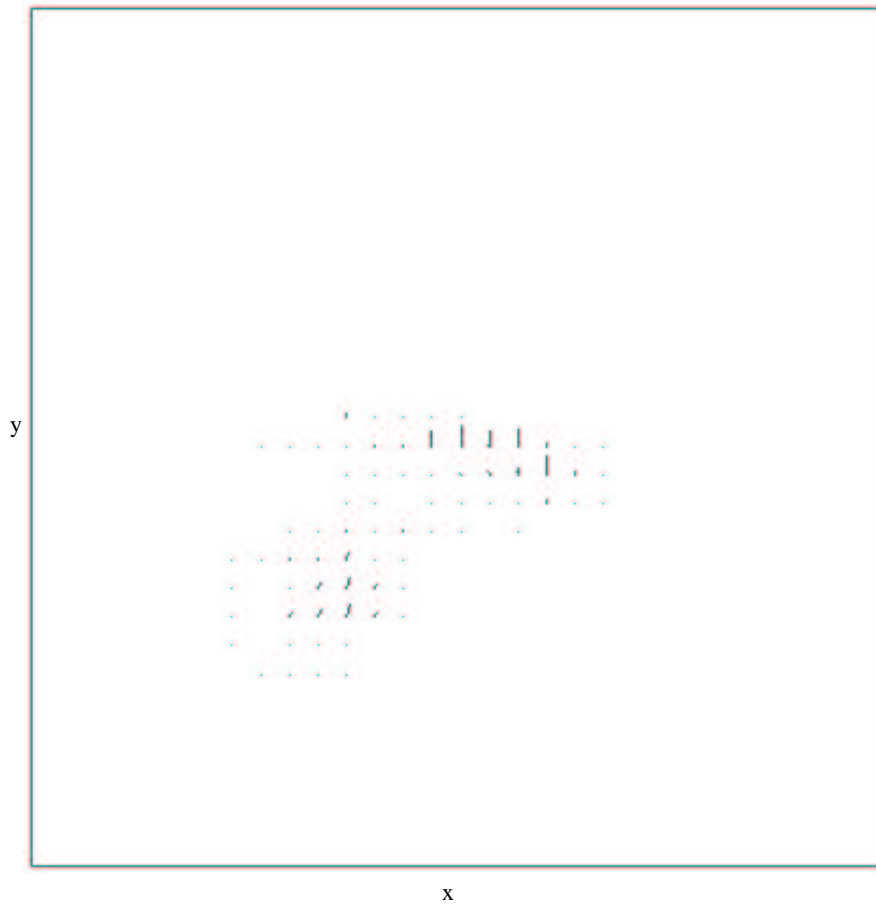


(b) Plot of the probability grid.

Figure 7. A typical situation in the experiments: uncertainty in  $y$  direction.



(a) Trajectory and perceptions.



(b) Plot of the probability grid.

Figure 8. A typical situation in the experiments: several local maxima.

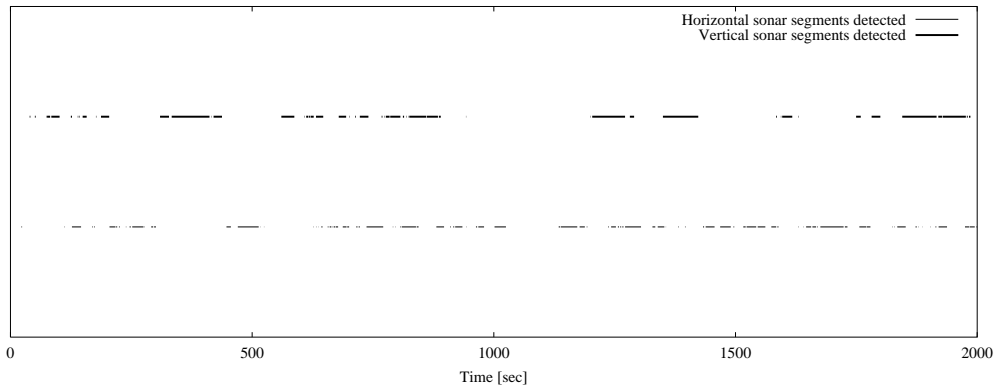
four dimensions using directed line segments. For each cell in  $P$  there is one directed line segment. The base of the line segment represents the position  $(x, y)$ , its direction represents  $\theta$ , and its length corresponds to the value of the grid cell. The probability grid is centred at  $p$ . In our case,  $P$  consists of 30 cells in  $x$  direction, 30 cells in  $y$  direction and 22 cells in  $\theta$  direction. The resolution of the grid in  $x$ - $y$  direction is  $\Delta\rho$ , the resolution in  $\theta$  direction is  $\Delta\theta$ . Many of the probability values are negligible. In Figure 7, the robot is situated in a corridor. Because the walls of the corridor are in  $y$ -direction, it was not able to detect any features in  $x$ -direction for some time. Therefore, the robot's pose in  $y$ -direction became uncertain over time. In Figure 8,  $P$  has several local maxima. That is, there is global uncertainty. Several positions are considered as current pose at the same time. Over time, when sonar segments are detected, this uncertainty will be resolved.

In the following, we provide experimental results about the performance of the position tracking algorithm. First, we provide data that has been obtained while the robot was performing the navigation task in the environment shown in Figure 6(b). In this scenario, the robot is constantly changing its direction of travel and the sonar sensor readings are very sparse. The performance of the position tracking algorithm under these worst-case conditions provides us with an indication of the maximum error in the position estimate. Second, we provide data that has been obtained using the simulator while the robot was performing the same task. In the simulation, we know the correct position of the robot and can compute the absolute error of the position estimate at any time.

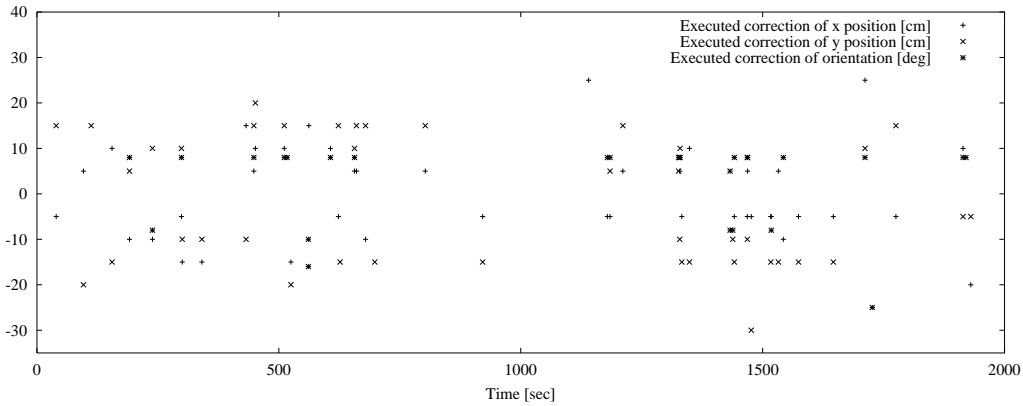
The position tracking algorithm performs a correction step only when sonar segments are detected. Figure 9(a) shows the length of the time intervals in which no horizontal or vertical sonar segments were detected. During the experiment, which lasts over 30 minutes, the robot is moving all the time. Because it changes its direction of travel constantly, there are periods of over 2 minutes in which no correction was possible. For the same experiment, the executed corrections of the robot's position and orientation are given in Figure 9(b). The maximum correction in  $x$ -direction is 25 cm, in  $y$ -direction 30 cm, and in orientation 16 deg.

Additional information about the same experiment is given in Figure 10. It reports on sonar segments in vertical direction and the correction in  $x$ -direction for the time interval from 0 to 400 sec. A correction of the robot's position and orientation computed by the position-tracking algorithm is only executed by the robot controller if the change is greater than 10 cm in the position and 3 deg in the orientation.

The data presented so far has been obtained using the real robot. We have tested our position tracking system also in simulations using the Pioneer 1 sim-



(a) Detection of sonar segments.

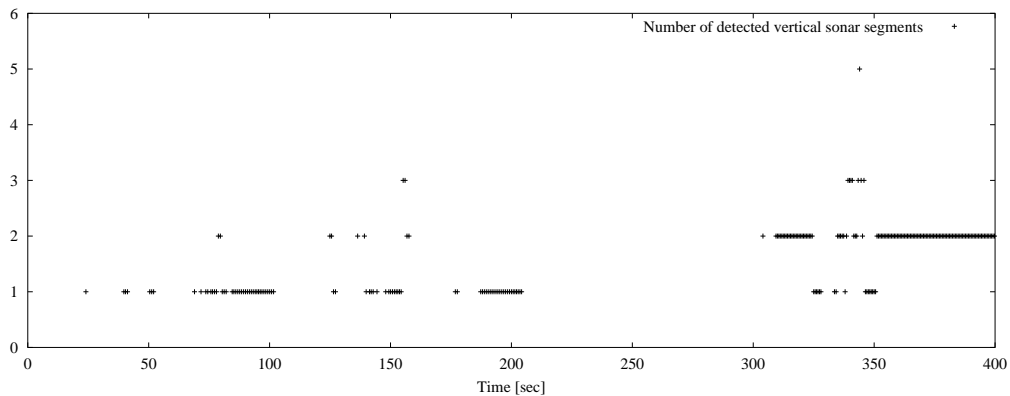


(b) Corrections of the robot's position and orientation.

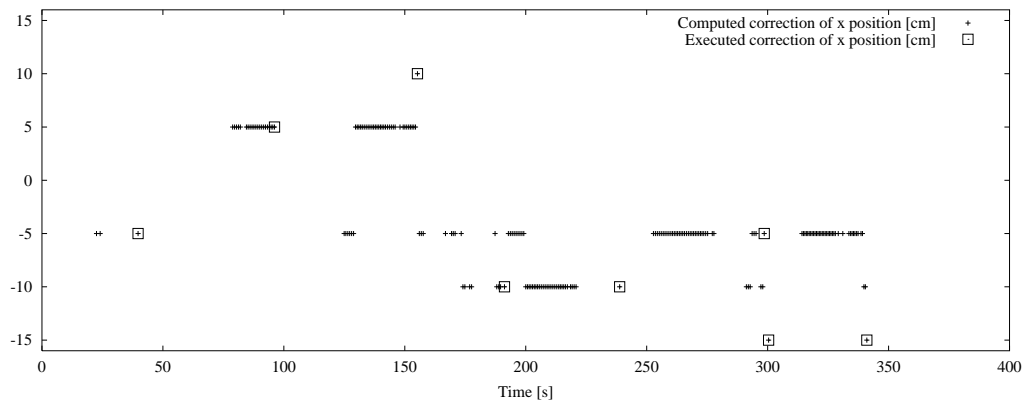
Figure 9. Experimental results for the Pioneer 1 performing the navigation task.

ulator developed by Konolige [11]. The simulator uses a model of the sonar and odometry sensors. The dead-reckoning capabilities of the robot are affected by three contributing sources of error: distance, turn angle (error in rotation), and drift (change in angle as the robot heads in a straight line). The simulator uses a distance error of 1 percent, a rotation error of 2 percent, and a drift error of 3 deg per meter. The model of the sonar sensors takes into account the distance and angle to the obstacle reflecting the sonar beam. If the reflection angle is at least 75 deg, the sensor is assumed to return correct distance readings. Otherwise, an additional distance is added which is due to specular reflections. The offset depends on the reflection angle and the true distance to the object. All sensor readings are randomised as well.

Figure 11 shows the length of the time intervals in which no horizontal or vertical sonar segments were detected for the navigation task in simulation. The results are similar to the data obtained on the real robot. The absolute



(a) Number of detected vertical sonar segments.



(b) Correction of  $x$ -position.

Figure 10. Data of Figure 9 in the interval 0 ... 400 sec.

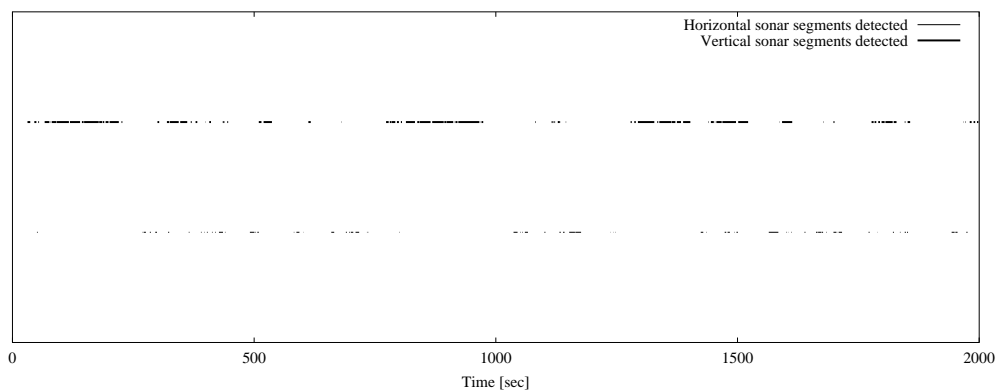
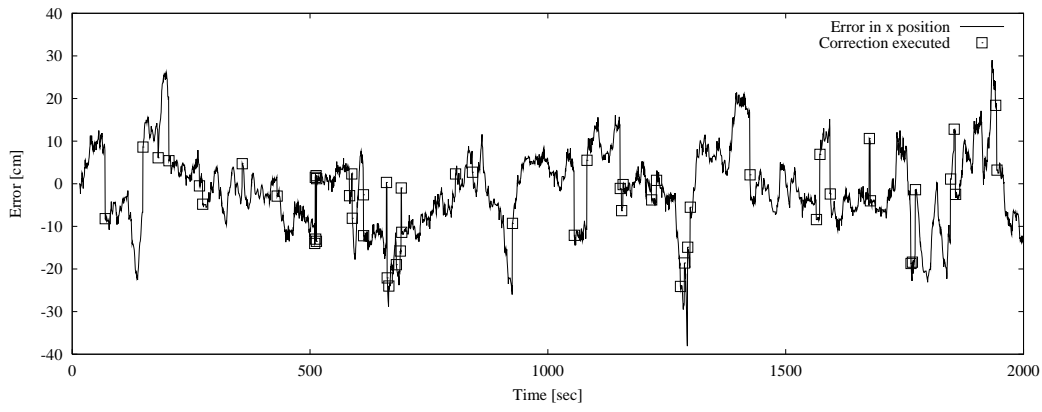
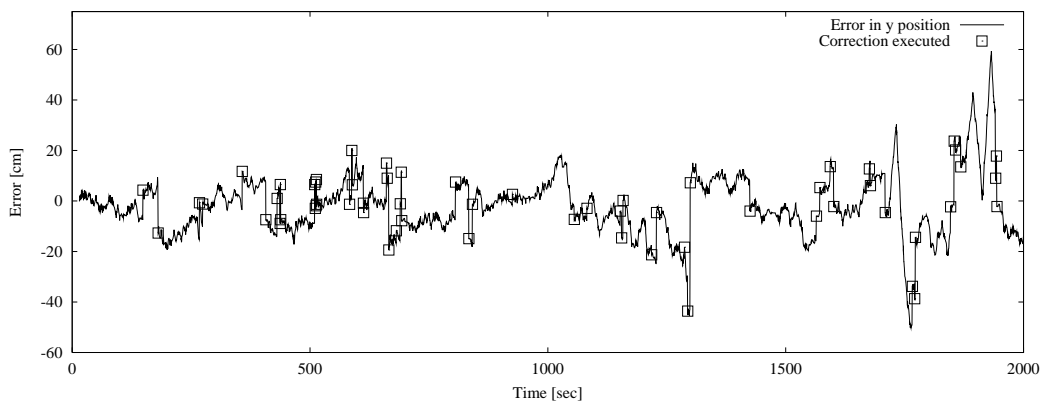


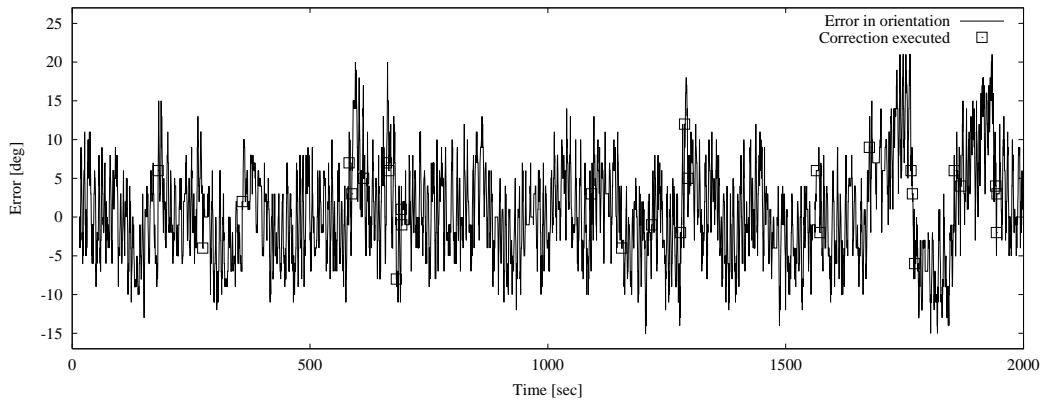
Figure 11. Detection of sonar segments for the navigation task in simulation.



(a) The error of the robot's position in  $x$ -direction.



(b) The error of the robot's position in  $y$ -direction.



(c) The error in the robot's orientation.

Figure 12. The navigation task in simulation.

errors in the robot's position and orientation are given in Figure 12. The maximum error in  $x$ -direction is 40 cm, in  $y$ -direction 50 cm, and in orientation 24 degrees. That is, the performance of the position tracking algorithm in the simulator is worse than on the real robot. This is due to the amount of randomness added by the simulator's sensor models. The data shown in Figure 12 demonstrates that each time a correction was performed, the absolute error in position or orientation was reduced. The large error in some situation is due to the fact that for long periods of time, no sonar segments were detected. It should be noted that the results have been obtained in a worst-case scenario. In other applications, e.g., the can-collection task, the error is less.

We do not know whether the proposed approach will work in highly cluttered environments. However, the experimental evidence suggests that it may be applied to typical office environments in which the walls are cluttered by various small objects. The method is still able to detect wall-like features present in the environment. Likewise, the presence of people does not cause severe problems, provided that the robot's sensors are not obstructed for long periods of time. When we have used the position tracking algorithm in robot-learning experiments, a few people were present in the environment, as it is the case for many office environments. The algorithm did work robustly in these conditions.

The proposed method can also be applied to large environments by using a dynamic window approach, in which only a small area of the world map is considered during localisation. A similar approach is used also by Markov localisation methods in order to deal with large environments [4]. The heuristic technique to distinguish parallel line segments, introduced in Section 4.2, works robustly. Recently, we have used the algorithm successfully in a corridor of the Computer Science department.

In terms of computation, the approach proposed in this paper is considerably more efficient than the re-implemented Markov localisation method [4]. This is mainly due to the fact that the feature detection using the Hough transform and the subsequent matching in the Hough space can be done very efficiently – a fact that is well-known from applications in computer vision. In our approach, we can easily compute a full update on the basis of 80 sonar readings in a tenth of a second on a 200 MHz Pentium PC.

## 7 Conclusions

We have presented a novel position tracking method for a mobile robot with seven sonar sensors. The method is based on Hough transform and probability grids. The method is robust and computationally efficient. To our knowledge,

it is the only existing position tracking for the Pioneer 1 mobile robot using sonar sensors.

The method relies on the detection of straight-line features in the sonar sensor readings. On the one hand, this is the reason for its computational efficiency. On the other hand, this also restricts its applicability to environments with wall-like features, such as offices and corridors. The paper tried to make a contribution in exploring what is possible using a mobile robot with a small number of sonar sensors.

The main contribution of this paper is to understand why this approach to localisation is more robust than other techniques. From our point of view, the main reasons are as follows: This approach is less dependent on individual sonar sensor readings than a traditional Markov localisation approach. We perform the correction step only when straight-line features were detected in the sonar readings. Rather than using the product of the likelihoods,  $p(s|L)$ , we use the correlation count (a combined feature) for the computation of position probabilities. In that way, misleading sensor readings that are likely to be caused by multiple reflection are filtered out.

The results reported in this paper indicate that grid-based Markov localisation will work if a sonar sensor model is used that takes into account the angle to the closest obstacle [16] or if non-plausible sonar sensor readings are filtered out before performing the probability update. The former method is not practical due to the amount of additional computation involved.

*Acknowledgements.* The authors wish to thank the participants of the IJCAI-99 Workshop on Reasoning with Uncertainty in Robot Navigation for useful discussions and comments. Also, thanks to the anonymous reviewers for many useful suggestions in improving this paper.

## References

- [1] R. C. Arkin. Integrating behavioural, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, 6:105–122, 1990.
- [2] D. H. Ballard. Generalising the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [3] W. Burgard, D. Fox, and D. Hennig. Active mobile robot localisation. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, 1997.
- [4] W. Burgard, D. Fox, and D. Hennig. Fast grid-based position tracking for mobile robots. In G. Brewka, C. Habel, and B. Nebel, editors, *Proceedings of the*

- 21st Annual German Conference on Artificial Intelligence (KI-97): Advances in Artificial Intelligence*, pages 289–300. Springer, 1997.
- [5] W. Burgard, D. Fox, and S. Thrun. Active mobile robot localisation by entropy minimisation. In *Proceedings of the Second Euromicro Workshop on Advanced Mobile Robots (Eurobot-97)*. IEEE Computer Society Press, 1997.
  - [6] J. J. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation, Scottsdale, AZ, USA*, pages 674–680, 1989.
  - [7] M. Drumheller. Mobile robot localisation using sonar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):325–332, 1987.
  - [8] A. Großmann and R. Poli. Continual robot learning with constructive neural networks. In A. Birk and J. Demiris, editors, *Learning Robots. Proceedings of the Sixth European Workshop (EWL-97)*, number 1545 in Lecture Notes in Artificial Intelligence. Springer, 1998.
  - [9] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localisation methods. In *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-98), Victoria, Canada*, 1998.
  - [10] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME. Journal of basic engineering*, (82):35–45, March 1960.
  - [11] K. Konolige. *Saphira Software Manual. Version 6.1*. ActivMedia, Inc., Peterborough, NH, USA, Oct. 1997.
  - [12] K. Konolige and K. Chou. Markov localisation using correlation. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 1154–1159, 1999.
  - [13] R. Krishnapuram and D. Casasent. Hough space transformations for discrimination and distortion estimation. *Computer Vision, Graphics, and Image Processing*, 38:299–316, 1987.
  - [14] V. F. Leavers. *Shape Detection in Computer Vision Using the Hough Transform*. Springer, London, UK, 1992.
  - [15] J. K. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic, 1992.
  - [16] S. Thrun. Personal communication, Aug. 1999.
  - [17] T. Veen and F. C. A. Groen. Discretisation errors in the Hough transform. *Pattern Recognition*, 14(1-6):137–145, 1981.
  - [18] X. Yun, K. Latt, and J. S. Glennon. Mobile robot localisation using the Hough transform and neural networks. In *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC/CIRA/ISAS Joint Conference), Gaithersburg, MD, USA*, pages 393–400. IEEE, 1998.