Active Global Localisation for a Mobile Robot Using Multiple Hypothesis Tracking

Patric Jensfelt
Signals, Sensors and Systems
Royal Institute of Technology
SE-10044 Stockholm, Sweden
patric@nada.kth.se

Steen Kristensen
Intelligent Systems Group
DaimlerChrysler Research and Technology
Alt-Moabit 96A, D-10559 Berlin, Germany
steen.kristensen@daimlerchrysler.com

Abstract

In this paper we present a probabilistic approach for mobile robot localisation using an incomplete topological world model. The method uses multi–hypothesis Kalman filter based pose tracking combined with a probabilistic formulation of hypothesis correctness to on–line generate and track Gaussian pose hypotheses. Apart from a lower computational complexity, this has the advantage over traditional grid based methods that incomplete and topological world model information can be utilised. Furthermore, the method generates movement commands for the platform in order to optimise the information gathering for the pose estimation process.

1 Introduction

The problem of global localisation—or "the kidnapped robot problem"—is that of, from little or no a priori pose[1] information, estimating the correct pose of a robot with respect to some global reference frame. This is fundamentally different from the *pose tracking* problem which is that of keeping the robot's pose estimate correct when the robot moves but knows where it was at some previous point in time. With *pose estimation* we will understand the general problem of determining where the robot is.

Pose estimation is essential to mobile robot navigation and as a consequence a substantial amount of work has been performed in this area [Borenstein *et al.*, 1995]. However, the large majority of this work has been concentrated on the pose tracking problem whereas global localisation has received relatively little attention. This is probably due to the fact that many robot systems get their initial pose supplied directly or indirectly by the user, which means that global positioning can in principle be omitted which is seldom the case for pose tracking. However, we think it is crucial for robots, which must be truly autonomous and operate over extended periods of time, that they possess the ability to autonomously locate themselves. This is relevant not only at startup, but also during operation for recovery in case the pose tracking process is lost or cannot unambiguously determine the robot's pose.

The general approach to global localisation (when not using GPS or similar "artificial" beacons) is to compare information—or features in the widest sense—extracted from sensor readings with an a priori given map associated with the global reference frame. Each comparison carries some evidence about where the robot can be, and the challenge is then to, as efficiently as possible, rule out all but the correct pose. The efficiency thus depends on two factors, namely the amount of evidence acquired with each feature/map comparison and the efficiency of the evidence fusion (relative to the effort of performing these operations). It is evident that this again strongly depends on the chosen framework and the representation of maps, "pose space", and pose hypotheses.

Our system is based on a topological world model [Kuipers and Byun, 1991], extended with landmarks and information about from where these landmarks can be seen (for more details see Figure 4 and [Kristensen *et al.*, 1998]). The world model is taught interactively by having the user lead the robot around in the surroundings while it on–line extracts information using its sensors and feature extractors. While this approach is fast, user friendly, and generates world models that are perfectly matched to the robot's perception capabilities it cannot be guaranteed that the model is complete in the sense that all possible landmarks of some kind have been seen or that the world cannot look differently at some later point in time when the robot is navigating autonomously in its surroundings. Except for mobile robot systems navigating in extremely controlled environments this is a general problem which means that the framework should be able to handle incomplete and partly incorrect world models. We have therefore chosen to use a probabilistic framework since this means that we can explicitly represent and treat model errors. In our system, however, we use a Kalman filtering technique to do the pose tracking of our robot. As this is a standard and well proven approach used in many mobile robot systems we would

[1] We will distinguish between the robot *position* which is a 2–tuple, (x, y), and the robot *pose*, which includes the orientation thus being a 3–tuple, (x, y, ϕ).

like to keep this and have a seamless integration with the global localisation scheme. We have therefore developed a hybrid localisation method using Kalman filtering to track a number of Gaussian pose hypotheses and probability theory to do evidence fusion. Furthermore, in order to optimise the amount of new evidence gathered by the perception components, the pose hypotheses and the world model information is used to generate movement commands for the robot, thus making the localisation process *active*.

In the remainder of this paper we will explain how we have done this and compare our approach with other existing methods which are outlined in Section 2. In Section 3 the generation and tracking of pose hypotheses is described which in Section 4 is followed by a description of how we probabilistically estimate the correctness of each hypothesis. In Section 5 it is explained how movement commands for the robot are generated in order for it to gather new information. The experiments conducted with the implemented system are outlined in Section 6 and finally in Section 7 the results are discussed, conclusions are drawn, and topics of further work are outlined.

2 Related Work

Most of the pose estimation literature is, as mentioned in the Introduction, concerned with the pose tracking problem. Often this is not stated explicitly but assumed implicitly in the sense that either the pose estimation problem is simply defined as being the pose tracking problem or the methods applied assume a good initial estimate of the robot pose.

An early example of addressing the global localisation problem is [Kuipers and Byun, 1991] where, in the context of topological map building, the *rehearsal procedure* is introduced, which basically consists in testing a global pose hypothesis by examining if the robot can move in the topological graph as predicted.

In [Stopp and Küttner, 1997] the global localisation problem is addressed as a case of the general object recognition problem. The basic observation is that the localisation problem is solved if the robot can recognise some place (a room, a hallway etc.), represented as a 2D object, and determine the relative pose to it. This is done efficiently using geometric hashing [Lamdan and Wolfson, 1988]. A similar approach is taken in [Crowley *et al.*, 1998] where PCA analysis is used to generate pose hypotheses which are—quite like in the work presented in this paper—tracked with Kalman filters until a unique "match" is determined. Common to both these approaches is that they require a relatively complete model of the environment but in return they are robust towards a certain degree of sensor noise and model uncertainty. Unfortunately it is difficult to evaluate exactly what happens in the presence of noise/uncertainty.

However, recently approaches explicitly dealing with sensing, model, and movement uncertainty have appeared [Koenig and Simmons, 1998; Simmons and Koenig, 1995; Kaelbling *et al.*, 1996; Thrun *et al.*, 1998; Fox *et al.*, 1999b]. Common to these approaches is that they use a probabilistic formulation to represent and update the pose of the robot which has the advantage of enabling them to handle uncertainty in a natural and convenient manner. These approaches, also known as Markovian methods, use a spatially discretised representation of the environment where each cell holds the probability that the robot occupies the area represented by the cell, and use a two–step procedure to update this representation, namely the *move step* where the fact that the robot moves is accounted for by shifting probability mass between cells according to the robot movement, and the *sense step* where Bayesian updating is used to incorporate new evidence stemming from a feature/map comparison. In general, the sense step concentrates probability mass in some areas and the move step disperses it. This "blurring" is due to the fact that the probability mass is not only shifted but also smeared in order to account for robot movement inaccuracies. In order for the global relocalisation to "converge" is it important that the evidence achieved in the sense step more than compensates for the additional pose uncertainty introduced by the move step. This fact stresses the importance of having a sensible movement/sensing strategy, i.e., to do active sensing, since moving randomly in general does not bear the promise of gaining evidence efficiently enough.

The approaches described in [Koenig and Simmons, 1998; Simmons and Koenig, 1995; Kaelbling *et al.*, 1996] and in [Thrun *et al.*, 1998; Fox *et al.*, 1999b], respectively, mainly differ in the chosen discrete pose representation. The former approaches use a coarse, topological representation (cell size approx. 1 meter) whereas the latter ones use a fine–grained grid (cell size approx. 0.1 meter) representation similar to the well–known occupancy grids for representing obstacles and free–space. The advantage of the grid representation, except of course for having a higher resolution and thereby being able to represent pose hypotheses more accurately, is that they facilitate the use of low level ("raw") sensor data for updating the cell probabilities whereas topological models require the extraction of abstract features such as T–junctions, walls and other landmarks. The disadvantage of the grid representation is their large number of cells which makes the updating computationally expensive (in [Fox *et al.*, 1999a] a solution to this using Monte Carlo methods is proposed) and the calculation of optimal sensing/movement strategies infeasible. This, on the other hand, is the advantage of the topological representations which are very nicely suited for the generation of optimal sensing/movement policies using the framework of Partially Observable Markov Decision Processes (POMDPs) and Bayesian Decision Theory [Kristensen, 1996].

For all the above mentioned works, the representation of pose state space and the a priori given world models are tightly matched meaning that [Koenig and Simmons, 1998; Simmons and Koenig, 1995], and [Kaelbling et al., 1996] use topological world models and [Thrun et al., 1998] and [Fox et al., 1999b] use occupancy grid type world models. This is a perfectly sensible approach since it eliminates the need for mappings between estimated pose and the world model and facilitates easy feature/map comparisons. Following this reasoning we should have chosen a discrete, topological representation of pose space. We were, however, interested in a seamless integration with our existing system which uses a *continuous* representation of pose in the form of a Kalman filter doing the pose tracking.

The typical argument against using a continuous pose representation is that this way only a restricted set of distribution types, typically unimodal ones, can be represented [Koenig and Simmons, 1998; Kaelbling *et al.*, 1996; Fox *et al.*, 1999a]. Unimodal distributions are particularly unsuited for global localisation since they cannot represent the case where the pose is ambiguous, i.e., the case where the robot might as well be at one place as at another. This criticism, however, largely fails the point since it ignores that fact that multiple unimodal distributions can be used to represent the overall probability distribution. In our approach, we have chosen to use a set of Gaussians to represent the probability distribution, where each Gaussian represents one pose hypothesis. This has the obvious advantage that we can use our standard Kalman based pose tracking to independently update each hypothesis in a simple scheme commonly known as Multiple Hypothesis Tracking (MHT) or Multiple Target Tracking [Bar-Shalom and Li, 1993; 1995] which is well–understood and computationally efficient. A further advantage of using a set of Gaussians is that it enables us to explicitly reason about each hypothesis whereas with discrete representations, in principle all cells (and their probabilities) have to be taken into account before acting which is normally too time consuming, especially with the grids, and is thus often circumvented by performing a thresholding on the probabilities. As a matter of fact, in [Fox *et al.*, 1999b] the grid representation is thresholded in order to come up with exactly a set of Gaussians. The reason for choosing Gaussians is, except for purely practical considerations, that, when using Kalman filtering for pose tracking, the pose estimate is a weighted sum of single measurements and thus the Central Limit Theorem means—given the usual and reasonable constraints of finite but non-zero measurement variance—that the true distribution in the limit (i.e., as the number of measurements goes towards infinity) will be normal, i.e., a Gaussian.

The drawback of using MHT is that the hypotheses per se do not carry any information about their correctness (apart from the fact that hypotheses with small variances may tend to be the correct ones) since they all have "probability mass" 1. It is therefore necessary to explicitly estimate the probability of each hypothesis being correct and for this we have chosen to use a probabilistic approach since it seems to be the most adequate framework due to the explicit and elegant handling of sensing and model uncertainties. How this is formulated when using MHT and a topological model is described after the next section where the basic algorithm for hypothesis creation and updating is presented.

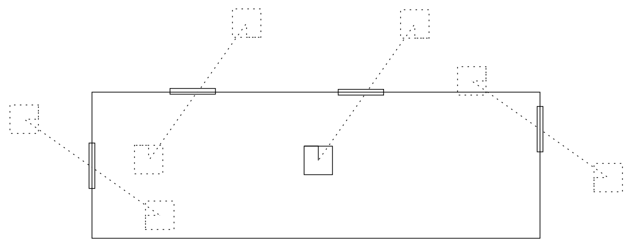


Figure 1: The idea behind the multiple pose hypothesis generation and tracking illustrated in a simple environment with only one room with four doors.

3 Pose Hypothesis Generation and Tracking

The idea behind the multiple pose hypotheses generation and tracking technique presented in this paper is best illustrated with a figure. Figure 1 shows a situation where the true position of the robot is given by the solid square in the middle of the room in the figure. An observation of a door is made slightly to the right in front of the robot. Given that the mapped environment only consists of a single room with four doors, there are eight possible poses for the robot from which it can see a door in front, slightly to the right. These eight poses give rise to eight hypotheses regarding the pose of the robot. The idea is that by making more observations of features in the environment and matching these to the hypotheses, the hypothesis corresponding to the true position of the robot will gain most evidence, making it distinguishable from the other, false, hypotheses.

It is obvious that the amount of information that can be extracted from a certain observation will depend on, e.g. the type of feature that is observed and the performance of the algorithm that extracts the features. The latter factor will be taken care of in Section 4 where the probability of a hypothesis being true is discussed. The first factor will lead us to introduce the concept of creative and supportive features.

3.1 Features

The features are extracted from sensor data by what we have chosen to call *recognizers*. Each recognizer is specialised in detecting a certain feature type. The recognizers will only report the occurrence of a new feature in the sensor data once. This is possible by storing the location of each detected feature and comparing the newly extracted features with the stored ones [Kristensen *et al.*, 1998].

Some features (e.g. doors) carry enough information to completely specify the pose of the robot, others (e.g. lines) will in the general case only determine the pose to be somewhere along some curve. We have chosen to divide the features into two sets of features, namely creative ($\mathcal{C}$) and supportive ($\mathcal{S}$). A creative feature carries enough pose evidence to initiate a new hypothesis, whereas the supportive ones can only support already existing hypotheses. It is obvious that $\mathcal{C} \subseteq \mathcal{S}$, i.e. a creative feature can of course also give support to an already existing hypothesis.

3.2 Pose Hypotheses

Let the true pose of the robot be given by $\mathbf{x} = (x, y, \theta)^T$. The pose is modelled to evolve according to

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \mathbf{u}(k) + \mathbf{v}(k) \tag{1}$$

where $\mathbf{u}(k)$ is the odometric information. We assume the noise, $\mathbf{v}(k)$, associated with the odometric model, to be a zero-mean, white and Gaussian noise process with covariance matrix $Q(k)$.

Each hypothesis, H_i, is an estimate of the true position, based on some assumptions regarding the association of extracted features to the features in the world model. The hypotheses are updated using a Kalman filter.

A hypothesis is represented by a pose, $\hat{\mathbf{x}}_i = (\hat{x}, \hat{y}, \hat{\theta})_i^T$, with an associated covariance matrix, Σ_i, and information about the probability of the hypothesis being the correct one, $P(H_i)$.

The pose estimates and their covariance matrices are driven by the odometric information according to the "time-update formula":

$$\begin{aligned} \hat{\mathbf{x}}_i(k+1|k) &= \hat{\mathbf{x}}_i(k|k) + T_i\mathbf{u}(k) \\ \Sigma_i(k+1|k) &= \Sigma_i(k|k) + T_iQ(k)T_i^T. \end{aligned} \tag{2}$$

where T_i is the rotation matrix from the odometric frame of reference to that of hypothesis H_i. $\hat{\mathbf{x}}_i(k+1|k)$ should be interpreted as the predicted pose, given by hypothesis H_i at time $k+1$, using measurements up to and until time k.

3.3 Data Association and Pose Hypotheses Update

Each detected feature creates a set of possible poses for the robot (compare with figure 1). Let us call these possible poses, pose candidates, C_j. The representation for these pose candidates consists of a pose $\mathbf{z}_j$, a covariance matrix R_j, and information about what feature created it. The reason for introducing the pose candidates, that are similar to the pose hypotheses, is to be able to make all extracted features provide homogeneous information. Let us also extend the concept of creative and supportive to include the pose candidates, i.e. a creative feature gives a creative pose candidate.

The pose candidates can be seen as measurements of the pose, generated by

$$\mathbf{z}_j(k) = \mathbf{h}(\text{detected feature}) + \mathbf{w}_j(k) \tag{3}$$

where $\mathbf{h}$ contains the map of the environment and the characteristics of the recognizers and $\mathbf{w}_j$ is measurement noise. These pose candidates will be used to update the

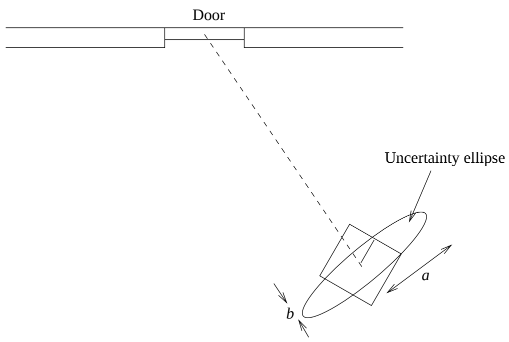


Figure 2: Creating a pose candidate from a door feature.

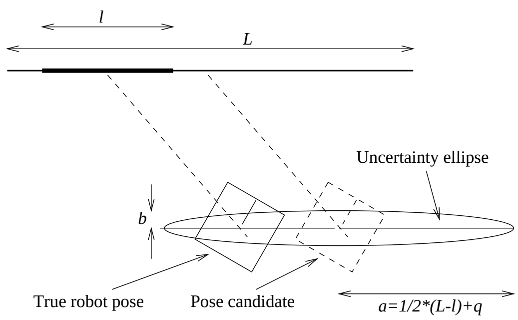


Figure 3: Creating a pose candidate from a line feature.

pose information of the pose hypotheses and also potentially initiate new hypotheses. Figures 2 and 3 give a flavour for how pose candidates are created from extracted features, illustrated with a door and a line feature. We make a Gaussian approximation regarding the uncertainty associated with the pose candidates, enabling the covariance matrix representation, R_j.

It is crucial that the data association problem be solved, i.e to determine what pose candidates correspond to what pose hypotheses. For each pair $(\mathbf{z}_j(k+1), \hat{\mathbf{x}}_i(k+1|k))$ of pose candidate and pose hypothesis, the innovation and its corresponding covariance can be defined as:

$$\begin{aligned} \nu_{i,j}(k+1) &= \mathbf{z}_j(k+1) - \hat{\mathbf{x}}_i(k+1|k) \\ S_{i,j}(k+1) &= \Sigma_i(k+1|k) + R_j(k+1). \end{aligned} \tag{4}$$

To be able to determine if a pose candidate matches a pose hypothesis, a validation gate is used. A pose candidate, C_j, is successfully matched against a pose hypothesis, H_i, if the following criteria is fulfilled

$$\nu_{i,j}(k+1)S_{i,j}(k+1)^{-1}\nu_{i,j}(k+1)^T \leq \gamma \tag{5}$$

where γ is the gate threshold.

Assuming that we have found a match between pose candidate C_j and pose hypothesis H_i, the pose candidate can be seen as a measurement of the robot's pose. Let $\mathbf{x}_i$ denote the position of the robot, given that $\hat{\mathbf{x}}_i$ is the correct hypothesis. Then $\mathbf{z}_j(k)$ is modelled as being generated by

$$\mathbf{z}_j(k) = \mathbf{x}_i(k) + \mathbf{w}_j(k). \tag{6}$$

The pose hypothesis is updated using the "measurement-update formula"

$$\begin{aligned} W_{i,j}(k+1) &= \Sigma_i(k+1|k)S_{i,j}(k+1)^{-1} \\ \hat{\mathbf{x}}_i(k+1|k+1) &= \hat{\mathbf{x}}_i(k+1|k) + W_{i,j}(k+1)\nu_{i,j} \\ \Sigma_i(k+1|k) &= \Sigma_i(k|k) - \\ & W_{i,j}(k+1)S_{i,j}(k+1)W_{i,j}(k+1)^T. \end{aligned} \quad (7)$$

Each unmatched, creative, pose candidate, C_j, will initiate a new pose hypothesis, H_i, according to:

$$\begin{aligned} \hat{\mathbf{x}}_i(k|k) &= \hat{\mathbf{x}}_j(k) \\ \Sigma_i(k|k) &= R_j(k). \end{aligned} \quad (8)$$

3.4 Track Splitting

Each time a pose candidate is used to update a pose hypothesis, there is a risk of making an association error, a risk that grows with the degree of uncertainty in hypothesis and candidate. In order not to loose the true pose hypothesis by updating it falsely, each hypothesis is split in two identical copies before the update is done. The updating, according to equation 7, is performed only on the original hypothesis. This procedure is in the literature of target tracking called track splitting. In order to keep the number of hypotheses low, we merge split hypotheses if they have not moved significantly away from each other (in a Mahalanobis distance sense).

4 Hypothesis Probability Estimation

In this section we will develop a formulation of the probability of a given hypothesis being correct.

If $P(H_i)$ describes the probability of the i'th hypothesis being correct, the new probability after the receipt of some sensor report (in our case about a newly detected feature), r, can—using Bayes' well–known inversion formula—generally be written as:

$$P(H_i|r) = \frac{P(r|H_i)P(H_i)}{P(r)} \quad (9)$$

where $P(r)$ can be calculated as:

$$P(r) = \sum_{i=1}^{N} P(r|H_i)P(H_i) \quad (10)$$

where N is the number of hypotheses. It is seen that $P(r)$ is effectively a scale factor which ensures that the $P(H_i|r)$'s sum to 1. This implicitly assumes that *one* of the hypotheses is correct which is true only under a closed world assumption[2]. In our case, however, this is not generally true, since it cannot be ensured that at any point in time a correct, modelled feature was seen and thus a correct pose hypothesis was generated. To overcome this problem we "close the world" by defining a hypothesis, H_0, which is the hypothesis encountering for all of the poses not being accounted for by hypothesis 1, ..., N. The introduction of H_0 also has the advantage of generalising the framework since we as initialisation, when no hypotheses have yet been generated, can account for all the probability mass by setting $P(H_0) = 1.0$. Therefore, instead of calculating $P(r)$ we simply normalise so that:

$$\sum_{i=1}^{N} P(H_i|r) = 1.0 - P(H_0) \quad (11)$$

The term $P(r|H_i)$ expresses the probability of receiving the report, r, given that the robot is at the position described by H_i. This is normally estimated using a map of the environment and can be re-written as:

$$\begin{aligned} P(r|H_i) &= P(r|f_j)P(f_j|H_i) + \\ & \quad P(r|\neg f_j)P(\neg f_j|H_i) \end{aligned} \quad (12)$$

where $P(r|f_j)$ is the probability that a report of type r is generated given that a feature of type f_j is seen by a sensor and $P(f_j|H_i)$ is the probability that a feature of type f_j can be seen given that the robot is at the position described by H_i. In other words, $P(r|f_j)$ is a model of the reliability of the recogniser extracting feature f_j from sensor data and $P(f_j|H_i)$ is a "map" of the environment. Thus the first term in equation 12 expresses the probability that a feature is actually detected given that it exists in the world. The second term correspondingly expresses the probability that a feature is detected although there is *no* such feature present, i.e., the probability of a false positive, or "phantom" feature. Again, however, equation 12 normally assumes a closed world in the sense that it is often assumed that $P(f_j|H_i)$ is fully specified by the map of the world[3]. To make explicit what in an incomplete model is known about $P(f_j|H_i)$ and what is not, we expand $P(f_j|H_i)$ as follows:

$$P(f_j|H_i) = P(f_{jm}|H_i) + P(f_{j\neg m}|H_i) \quad (13)$$

where $P(f_{jm}|H_i)$ expresses the probability that a modelled feature of type j can be seen from the position described by H_i and $P(f_{j\neg m}|H_i)$ the probability that a non–modelled feature of type j can be seen. Substituting this into equation 12 we get:

$$\begin{aligned} P(r|H_i) &= P(r|f_j)[P(f_{jm}|H_i) + P(f_{j\neg m}|H_i)] + \\ & \quad P(r|\neg f_j)P(\neg f_j|H_i) \end{aligned} \quad (14)$$

From this expression it is seen how accounting for unmodelled features adds an "in–flux" of probability–mass to $P(r|H_i)$. This has the effect that when a feature is seen where none was predicted according to some hypothesis and the map, this hypothesis will not be completely ruled out since the feature could simply be unmodelled (or a phantom). The advantage of using expression 13 is that it enables the updating to account for the fact that not all feature types may be equally thoroughly (completely) modelled.

Note that we do not explicitly treat the case where modelled features for some reason do no longer exist in

[2]For the Markovian methods this corresponds to the assumption that the robot is always inside the area represented by the grid or the topological map.

[3]Sometimes $P(r|H_i)$ is directly referred to as "the map".

the world (either because they were removed or because they are occluded). This is because the approach is data driven, i.e., something only happens when a feature *has* been seen, and thus non–existing but modelled features only influence the framework by causing some extra hypotheses to be generated.

For the calculation of $P(f_j|H_i)$ we exploit the fact that we have a finite number of well–defined hypotheses. If a match between H_i and a from f_j generated candidate, C_l, has been established, we make the assumption that $P(f_{j\neg m}|H_i) = 0$ and use the fact that the likelihood of C_l being a observation of f_j given H_i is given as [Larsson *et al.*, 1994]:

$$L(f_j|H_i) = \frac{1}{(2\pi)^{\frac{3}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(\hat{\mathbf{x}}_i - \mathbf{z}_l)\Sigma_i^{-1}(\hat{\mathbf{x}}_i - \mathbf{z}_l)^T} \quad (15)$$

To get an estimate of $P(f_j|H_i)$ we simply normalise this to lie between 0 and 1, i.e.:

$$P(f_j|H_i) = \exp^{-\frac{1}{2}(\hat{\mathbf{x}}_i - \mathbf{z}_l)\Sigma_i^{-1}(\hat{\mathbf{x}}_i - \mathbf{z}_l)^T} \quad (16)$$

When a H_i has not been matched to any C_l generated by f_j we use $P(f_j|H_i) = P(f_{j\neg m}|H_i)$ which in our system is an a priory given constant depending on the type of feature.

We can now formulate an expression for $P(H_0)$, i.e., the probability that none of the hypotheses $1, \ldots, N$ are correct. Since a correct hypothesis is generated as soon as a modelled, creative feature is detected, the probability that no correct hypothesis exists is the probability that only non–modelled and phantom features have so far been detected, thus:

$$P(H_0) = \begin{cases} \prod_{k=1}^{M} \sum_{i=1}^{N} [P(r|f_k)P(f_{k\neg m}|H_i) \\ \quad +P(r|\neg f_k)P(\neg f_k|H_i)]P(H_i) & \text{if } M > 0 \\ 1.0 & \text{if } M = 0 \end{cases} \quad (17)$$

where M is the number of creative features detected. If we denote the probability after receiving n features $P^n(H_0)$ (previously also referred to as $P(H_0|r)$) we can express the decrease in $P(H_0)$ upon the receipt of the $n+1$'th creative feature as $\Delta P^n(H_0) = P^n(H_0) - P^{n+1}(H_0)$ which is exactly the amount of probability mass, the on the basis of the $n+1$'th feature generated hypotheses carry. When $P(H_0)$ has decreased below some limit, γ, no new hypotheses are spawned.

Furthermore, in order to keep the number of hypotheses low, we delete hypotheses for which $P(H_i) < \eta$. In an ideal case, this may eventually lead to the situation where only one hypothesis exists thus automatically changing the approach from localisation to pure position tracking.

Finally some remarks on independence. The fact is that all of the above formulas require that the observations are independent. This is in general only true when the robot position is known, which is in the nature of the matter not the case in this application. Actually, the whole method lives from the fact that there *is* a certain dependency between measurements, namely the one hopefully given by the map. Due to the construction of the recognisers, however, which independently track extracted features and thus only report them as "new features" once, we do not get repeated observations of the same features just because the robot for example stands still for a while. This, and the fact that other research has shown that Bayesian updating seems to be quite robust against a violation of the independency assumption, leads us to believe that we can use the formulas derived above in spite of dependent observations.

4.1 Overall Algorithm

The overall algorithm for the update of the hypotheses, including the probabilistic update of the $P(H_i)$'s, can be formulated as:

Wait for feature
Generate pose candidates
Predict new pose for all hypotheses (eq. 2)
loop over all candidates, $C_j,\ j = 1, \ldots, M$
  loop over all hypotheses, $H_i,\ i = 1, \ldots, N$
    if(C_j matches H_i) (eq. 5)
      Split hypothesis H_i: $H_{N+1} := H_i,\ N := N+1$
      Update hypothesis H_i (eq. 7)
      calculate the nominator of eq. 9
    else if($C_j \in \mathcal{C}$ and $P(H_0) \geq \gamma$)
      init new hypo. (eq. 8) with $P^0(H_i) = \frac{\Delta P^n(H_0)}{L}$
Normalise $P(H_i)$'s (eq. 11)
Delete H_i's for which $P(H_i) < \eta$

where L is the number of new hypotheses generated from a detected creative feature.

5 Generation of Movement Commands

As previously established is it important that the robot has an active exploration strategy since it is doubtful that it by mere coincidence may be able to resolve ambiguities and gather information efficiently enough to compensate for odometry drift.

We have implemented a simple strategy based on the following heuristics:

- Moving on the edges of the topological graph makes it likely that the robot will move on a path that is free from obstacles.
- Avoid going to the same position twice as it is unlikely to provide any new information.
- Try to go to a place where a maximum of new features can be seen.

At startup, when no hypotheses exist, we use a simple exploration strategy to move the robot. When at least one pose hypothesis has been generated, we choose the one with the highest $P(H_i)$ (in case of a draw we simply select the one that happens to be first in the list) and search the topological graph from the current location of this hypothesis to find the one of the neighbouring nodes not previously visited and containing the largest number of features. This node is then selected as the next node to go to.

It is clear that this greedy strategy is not optimal and one of the next steps will be to improve this with either POMDP style planning or along the lines described in [Fox *et al.*, 1999b] where the actions are selected so as to minimise the expected entropy after moving and sensing. An alternative strategy, however, is to exploit the fact that we have a well–defined set of hypotheses and then just select the action that the best discriminates the two most likely hypotheses. While being much simpler and computationally efficient, new research indicates that this may lead to equally good results [Larsen *et al.*, 1998].

6 Experimental Results

The framework developed in this work has been tested on a mobile robot (see Figure 4) using a Sick laser scanner as sensor. The features used were doors and line segments, being creative and supportive features, respectively.

The topological world model used contained 29 doors, 139 line segments (generally corresponding to walls), and 33 nodes covering an area of 1500 m^2. The model was taught in simulation, i.e., by leading the robot around in a simulated world based on a coarse CAD model of our office environment. This kind of model generation has the advantage that we can teach large models without having problems with odometry error and corresponding model inconsistencies (algorithms for dealing with these problems in the real world are under development). However, using CAD data also means that the model is inherently uncertain and incomplete since the real building does not correspond exactly to the blueprints and since all the inventory is not modelled. In Figure 4 a part of the model including pose hypotheses is shown.

The work presented in this paper is work in progress and we therefore do not yet have results statistically describing the performance of the global relocalisation. What all our experiments show, though, is that the MHT as well as the Bayesian probability estimation works very reliably, quickly determining the true pose of the robot. Typical data for a global localisation attempt is shown in Table 1.

When the global localisation fails, it is always due to the movement strategies not being sufficiently intelligent. There are two failure modes: one where the robot starts far away from a creative feature (i.e., a door) and the initial exploration strategy fails to take the robot to a position where a creative feature can be seen. The second failure mode is due to situations where the strongest hypothesis is a false one on which the movement strategy bases its decisions where to move the robot. This occasionally causes the robot to get stuck before seeing enough features to make the true hypothesis the strongest one. This situation basically corresponds to the case where one of Kuipers and Buyn's rehearsals fails and should thus have an effect on the $P(H)$ of the corresponding hypothesis. We have not yet developed this feedback loop which is a topic of further work.

7 Discussion, Conclusion, and Further Work

In this paper we have presented a hybrid approach to global localisation for a mobile robot based on Bayesian probability theory and multiple hypothesis tracking using Kalman filtering of Gaussian pose hypotheses.

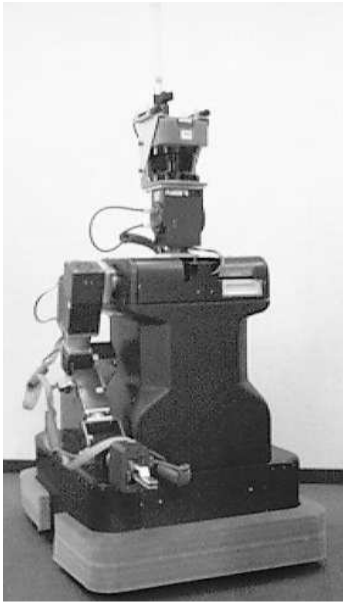
The reason for choosing a continuous pose representation in the form of Gaussians versus the common discretised representations used by other probabilistic approaches were multiple:

- Standard Kalman filtering, already present in our and many other robot systems, can be used to track and update the pose hypotheses.
- There is no trade–off between representation accuracy and size, i.e., we maintain high accuracy with a very low demand for memory.
- Having a few (typically < 100) hypotheses it is possible to on–line generate a sensible (active) movement/sensing strategy.
- It is in general computationally efficient.

The drawbacks of using MHT is that it is necessary to solve the data association problem, i.e., to explicitly match measurements and hypotheses and that it is not possible to represent an arbitrary probability distribution of robot pose. We believe the latter problem to be of purely theoretical importance and the former was solved using standard techniques from MHT such as track splitting and matching using the Mahalanobis distance.

An important issue that has to be addressed when using MHT for global localisation, though, is that of assessing the likeliness of each hypothesis being the correct one. For this it was chosen to use Bayesian probability theory since this offers an elegant framework for evidence fusion and an explicit treatment of model and sensing uncertainty.

The drawback of using hybrid approaches is normally that two or more techniques have to be implemented and applied in parallel. However, the Kalman representation of pose hypotheses as Gaussians directly gives a solution to one of the traditionally hard problems in the Markov/Bayesian approaches, namely to estimate the probability of a measurement given some pose hypothesis. So apart from using the frameworks we find to be the most suitable for hypothesis update and assessment, respectively, we have in addition gained a synergistic effect between them clearly easing the formulation and implementation of the latter, which, as a further benefit resulting from the fact that only a few pose hypotheses need to be treated, can be implemented very easily without the optimisation "tricks" otherwise necessary for the grid based approaches.



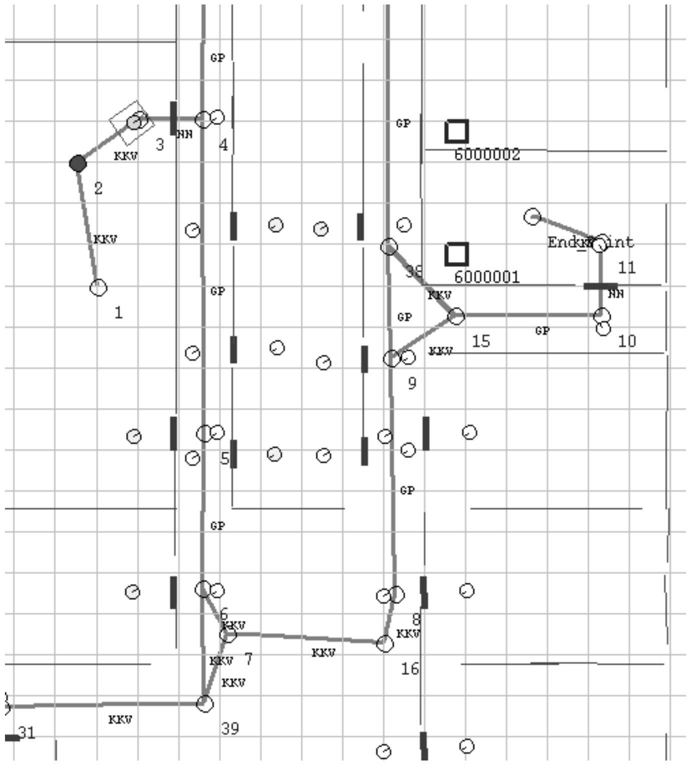


Figure 4: **Left:** The mobile platform used for the experiments. The robot consists of a re–fitted LabMate base equipped with a 7–DOF Amtec manipulator and a Sick lidar plus a set of colour CCD cameras mounted on a pan/tilt unit. The robot has two on–board P200 single board computers and radio Ethernet providing a facility for sending status information to a stationary workstation. **Right:** The world model used for the localisation. The thick lines and the circles they are connecting is the basic topological graph. The thin lines are the walls, the black bars are the doors, and the small circular robot icons are the hypotheses. The larger, quadratic robot icon in the upper left corner (by node 3) shows where the robot really was. The two squares to the right represent 3D docking objects which are used for other purposes.

snapshot	time [s]	robot pose	M_{door}	M_{line}	N_{hyp}	$P(H_{1st})$	$P(H_{2nd})$
1	4.5	(0.00,0.00,90.0)	2	5	114	0.105	0.052
2	10.5	(0.00,0.58,90.2)	3	7	114	0.110	0.054
3	16.5	(0.02,1.22,75.1)	3	9	82	0.167	0.058
4	22.2	(0.01,1.20,31.4)	4	10	52	0.252	0.084
5	28.5	(0.02,1.20,329.8)	4	12	46	0.415	0.093
6	28.5	(0.02,1.20,282.6)	6	12	38	0.352	0.121
7	39.3	(0.02,0.71,270.4)	6	13	31	0.447	0.100
8	46.9	(0.14,-1.73,272.5)	11	20	11	0.919	0.031

Table 1: Typical data for a localisation attempt. The development is illustrated with "snapshots" which show the system state at various time instances. The robot pose is the odometry relative to the starting point, i.e., it illustrates the robot's movement, not the pose of the currently best estimate. $P(H_{1st})$ and $P(H_{2nd})$ are the probabilities of the currently best and second best hypothesis, respectively.

The global localisation approach was implemented and tested on a real mobile robot and was found to work according to the expectations. When the global localisation failed it was due to the simple exploration strategy not being able to take the robot to a point where a creative feature could be seen or that the robot got stuck while pursuing a wrong hypothesis. Since the correct pose hypothesis normally becomes the strongest one after observing just a few features, this is generally only the case just after the first creative feature has been seen. The MHT and the Bayesian evidence fusion has been found to work flawlessly, only using very limited computational resources.

Further work will therefore go in the direction of improving the active sensing and the exploration strategies and modelling the effect of a failed robot motion on $P(H_i)$. Moreover, we will seek to enlarge the set of features, especially the creative ones. Here, f.ex. geometric hashing [Stopp and Küttner, 1997] seems to be a promising method for efficiently extracting pose candidates from lower level features such as (incomplete) line segments. Also line segments that are believed to be completely seen and having same length as some model lines may be used as creative features.

The feature based approach used in this work was partly motivated by the fact that our navigation system is based on a topological model extended with landmark information. We, however, do not think that the approaches presented in this paper are restricted to systems using such high level models. Using a grid based world model it should be possible to extract pose candidates from laser scans or even sonar readings, although it is obvious that the formulation of model uncertainty has to be modified.

References

[Bar-Shalom and Li, 1993] Yaakov Bar-Shalom and Xiao-Rong Li. *Estimation and Tracking: Principles, Techniques, and Software.* Artech House, Norwood, MA, 1993.

[Bar-Shalom and Li, 1995] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-Multisensor Tracking: Principles and Techniques.* YBS, 1995.

[Borenstein *et al.*, 1995] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Sensors and Techniques.* A. K. Peters, Ltd., Wellesley, MA, 1995.

[Crowley *et al.*, 1998] J. L. Crowley, F. Wallner, and B. Schiele. Position estimation using principal components of range data. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 3121–3128, Leuven, Belgium, May 1998. IEEE.

[Fox *et al.*, 1999a] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization—efficient position estimation for mobile robots. In *Submitted to AAAI'99*, 1999.

[Fox *et al.*, 1999b] D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. *Robots and Autonomous Systems*, 25(3-4):195–207, 1999.

[Kaelbling *et al.*, 1996] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Partially observable markov decision processes for artificial intelligence. In L. Dorst, M. van Lambalgen, and F. Voorbraak, editors, *Reasoning with Uncertainty in Robotics*, Lecture Notes in Artificial Intelligence 1093, pages 146–162, Berlin, 1996. Springer. Proceedings of the International Workshop RUR'95, Amsterdam, The Netherlands, December, 1995.

[Koenig and Simmons, 1998] Sven Koenig and Reid G. Simmons. Xavier: A robot navigation architecture based on partially observable markov decision process models. In David Kortenkamp, R. Peter Bonasso, and Robin Murphy, editors, *Artificial Intelligence and Mobile Robots*, pages 91–122. AAAI Press/The MIT Press, 1998.

[Kristensen *et al.*, 1998] S. Kristensen, V. Hansen, K. Kondak, and S. Horstmann. A modular architecture for a flexible autonomous service robot. In *Proceedings of the Sixth International Symposium on Intelligent Robotic Systems 98*, pages 93–100, Edinburgh, UK, July 1998.

[Kristensen, 1996] Steen Kristensen. Sensor planning with bayesian decision theory. In L. Dorst, M. van Lambalgen, and F. Voorbraak, editors, *Reasoning with Uncertainty in Robotics*, Lecture Notes in Artificial Intelligence 1093, pages 353–367, Berlin, 1996. Springer. Proceedings of the International Workshop RUR'95, Amsterdam, The Netherlands, December, 1995.

[Kuipers and Byun, 1991] Benjamin J. Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63, 1991.

[Lamdan and Wolfson, 1988] Y. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc. of the 1988 IEEE Int. Conf. on Robotics and Automation (ICRA'88)*, pages 1407–1413, Philadelphia, PA, April 1988. IEEE.

[Larsen *et al.*, 1998] T. D. Larsen, N. A. Andersen, and O. Ravn. Sensor management for identity fusion on a mobile robot. In *Proceedings of the Fifth International Conference on Control, Automation, Robotics and Vision (ICARCV'98)*, 1998.

[Larsson *et al.*, 1994] U. Larsson, J. Forsberg, and Å. Wernersson. On robot navigation using identical landmarks: Integrating measurements from a time–of–flight laser. In *Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 17–26, Las Vegas, Nevada, October 1994. IEEE.

[Simmons and Koenig, 1995] Reid Simmons and Sven Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. of the 14th IJCAI*, pages 1080–1087, Montreal, Canada, August 1995.

[Stopp and Küttner, 1997] A. Stopp and L. Küttner. A unified approach for fast recognition, learning and interpretation of the environment by an autonomous mobile robot. In *Proceedings of the International Symposium on Intelligent Systems and Semiotics - A Learning Perspective*, Gaithersburg, Maryland, 1997.

[Thrun *et al.*, 1998] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schmidt. Map learning and high–speed navigation in RHINO. In David Kortenkamp, R. Peter Bonasso, and Robin Murphy, editors, *Artificial Intelligence and Mobile Robots*, pages 21–52. AAAI Press/The MIT Press, 1998.