

# Active Global Localisation for a Mobile Robot Using Multiple Hypothesis Tracking

Patric Jensfelt and Steen Kristensen

*Abstract*— In this paper we present a probabilistic approach for mobile robot localisation using an incomplete topological world model. The method, which we have termed Multi Hypothesis Localisation (MHL), uses multi-hypothesis Kalman filter based pose tracking combined with a probabilistic formulation of hypothesis correctness to generate and track Gaussian pose hypotheses on-line. Apart from a lower computational complexity, this approach 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 to enhance the gathering of information for the pose estimation process. Extensive experiments are presented from two different environments, a typical office environment and an old hospital building.

*Keywords*— Global Localisation, Feature Based, Bayesian, Multiple Hypothesis

## I. INTRODUCTION

THE problem of global localisation—or “the kidnapped robot problem”—is that of, from little or no a priori pose<sup>1</sup> information, estimating the correct pose of a mobile robot with respect to some global reference frame. This is fundamentally different from the *pose tracking* problem, i.e. maintaining an estimate of the robot’s pose as the robot moves. With *pose estimation* we will denote 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 [1]. However, the large majority of this work has been concentrated on the pose tracking problem whereas global localisation has received less attention. This is due to the fact that many robot systems get their initial pose supplied directly or indirectly from the user which means that global positioning may be omitted. However, for the general problem of autonomous navigation, initialisation is crucial. This is relevant not only at startup, but also during operation for recovery in case of pose tracking failure.

The general approach to global localisation (when not using GPS or “artificial” beacons such as bar codes and transponders) is to compare information—or features in the widest sense—extracted from sensor readings with an a priori map associated with the global reference frame. Each comparison carries some evidence about where the

robot may be, and the challenge is then, as efficiently as possible, to find the correct pose. The efficiency thus depends on two factors, i) the amount of evidence acquired with each feature/map comparison and ii) the efficiency of the evidence fusion (relative to the effort of performing these operations). It is evident that this strongly depends on the chosen framework and the representation of maps, “pose space”, and pose hypotheses.

Our approach is based on topological world models [2], extended with landmarks and visibility information (for more details see Figure 5 and [3]). However, it cannot be guaranteed that the models are complete or that the world cannot look differently at some later point in time when the robot is navigating autonomously in the environment. Except for mobile robot systems navigating in extremely controlled environments this is a general problem. Therefore the framework should be able to handle incomplete and partially incorrect world models.

In our system, we use a Kalman filtering technique to represent and update the pose of the robot. As this is a standard and well proven approach [4] used in many mobile robot systems we would like to keep it and have a seamless integration with the global localisation scheme. We want to explicitly represent and treat model errors, which we think is best done using a probabilistic framework. We have therefore developed a hybrid localisation method using Kalman filtering to track a number of pose hypotheses and probability theory for evidence fusion. Furthermore, to optimise the amount of new evidence gathered by the perception system, the pose hypotheses and the world model information are used to generate movement commands for the robot, thus making the localisation process *active*.

In Section II we outline other work done on the global localisation problem. In Section III the generation and tracking of pose hypotheses is described, followed by a description of how we estimate the correctness of each hypothesis in Section IV. In Section VII it is explained how movement commands for the robot are generated in order for it to gather new information. The experiments conducted are presented in Section VIII and finally in Section IX, the results are discussed and conclusions are drawn.

## II. RELATED WORK

MOST of the pose estimation literature has concentrated on 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 assume a good initial estimate of the robot pose.

An early example of addressing the global localisation

P. Jensfelt works within the Centre for Autonomous Systems, Royal Institute of Technology, Fiskartorpsv. 15A, SE-100 44 Stockholm, Sweden. E-mail: patric@nada.kth.se S. Kristensen works with the Cognition and Robotics Group, DaimlerChrysler Research and Technology, Alt-Moabit 96A, D-10559 Berlin, Germany. E-mail: steen.kristensen@daimlerchrysler.com

<sup>1</sup>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,  $(x, y, \theta)$ .

problem is [2] where, in the context of topological map building, the *rehearsal procedure* is introduced. This consists in testing a global pose hypothesis by examining if the robot can move in the environment as predicted by the topological graph.

In [5] 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 [6]. A similar approach is taken in [7] where PCA analysis is used to generate pose hypotheses which are tracked with Kalman filters until a unique “match” is determined. Common to 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.

However, most recently approaches explicitly dealing with sensing, model and movement uncertainty have appeared [8], [9], [10], [11], [12]. Common to these approaches is that they use a probabilistic formulation to represent and update the pose of the robot. This 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. They use a two-step procedure to update this representation, namely (using their nomenclature) 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 to account for robot movement inaccuracies. For the global localisation to “converge”, it is 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 an efficient movement/sensing strategy, i.e., to do active sensing, since moving randomly in general does not guarantee gaining evidence efficiently enough. In [13] the movement strategy of the robot is determined by minimising the expected entropy.

The approaches described in [8], [9], [10] and in [11], [12], respectively, mainly differ in the 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 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 of the pose space is the large number of cells which results in an increased computational complexity and makes the calculation of optimal sensing/movement strategies very challenging. 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 [14].

In [15] a new version of the Markov Localisation method is presented, called Monte Carlo Localisation (MCL). By representing the probability distribution using a set of samples, almost arbitrary distributions can be represented and no discretisation has to be done. The computational complexity is also reduced significantly. The sample based representation has also successfully been applied to the map building problem [16].

For most of the above mentioned works, the representation of pose state space and the a priori given world models are tightly matched. Topological world models are used in [8], [9], [10], whereas [11] and [12] use occupancy grid type world models. This is a natural approach, since it eliminates the need for mappings between the estimated pose and the world model and facilitates easy feature/map comparisons. Following this reasoning, we might have chosen a discrete, topological representation of the pose space. We were, however, interested in a seamless integration with our existing systems which use a *continuous* representation of pose in the form of a Kalman filter doing the pose tracking. In [17] it is concluded that the Kalman filter methods provide high accuracy and efficiency.

The typical argument against using a continuous pose representation is that only a restricted set of distribution types, typically unimodal ones, can be represented this way [8], [10], [15]. 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, ignores that fact that multiple unimodal distributions can be used to represent the overall probability distribution. In [18] a set of Gaussians was used, each Gaussian representing one pose hypothesis. The advantage of this is that a standard Kalman based pose tracking module can be used to independently update each hypothesis in a simple scheme commonly known as Multiple Hypothesis Tracking (MHT) or Multiple Target Tracking [19], [20]. The multi Gaussian representation has also been used in [21] and [22]. The main difference to the work presented in this paper is that they do not explicitly consider world modelling aspects and active sensing strategies.

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 computationally challenging and is therefore often circumvented by performing a thresholding on the probabilities. As a matter of fact, in [12] the

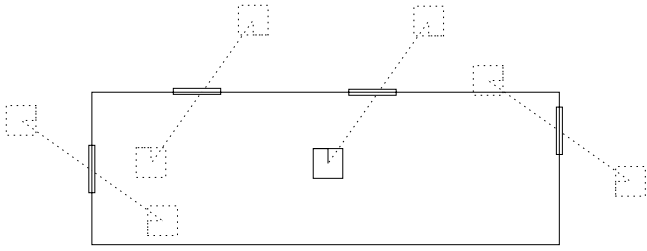


Fig. 1. The idea behind the multiple pose hypothesis generation and tracking illustrated in a simple environment with only one room with four doors.

grid representation is thresholded and approximated with a set of Gaussians.

A drawback of using a set of Gaussians 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 correctness confidence of each hypothesis. 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.

### III. 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.

A door is detected slightly to the right in front of the robot. Matching this feature to the map, consisting of four door in one room, results in eight likely robot poses. 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 pose 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 the feature type and the uncertainty associated with map and feature detection. The latter factor will be taken care of in Section IV 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.

#### A. Features

The features are extracted from sensor data by *recognisers*. Each recogniser is specialised in detecting a certain feature type. The recognisers 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 [3].

Some features (e.g. doors) are full rank features, i.e. they

constrain all three degrees of freedom, others (e.g. walls) will in the general case only determine the pose to be somewhere along a manifold. 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 give support to an already existing hypothesis.

#### B. 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

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{v}(k)) \quad (1)$$

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 pose, 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 estimate,  $\hat{\mathbf{x}}_i = (\hat{x}, \hat{y}, \hat{\theta})_i^T$ , with an associated covariance matrix,  $\Sigma_i$ , 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”:

$$\begin{aligned} \hat{\mathbf{x}}_i(k+1|k) &= \mathbf{f}(\hat{\mathbf{x}}_i(k|k), \mathbf{u}(k), 0) \\ \Sigma_i(k+1|k) &= F_i \Sigma_i(k|k) F_i^T + G Q G^T. \end{aligned} \quad (2)$$

where  $F_i$  and  $G$  are the Jacobians of  $\mathbf{f}$  with respect to  $\hat{\mathbf{x}}_i(k|k)$  and  $u(k)$  respectively.  $\hat{\mathbf{x}}_i(k+1|k)$  should be interpreted as the predicted pose, given by hypothesis  $H_i$  at time  $k+1$ , using measurements up to and until time  $k$ .

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. 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

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

where  $\mathbf{h}$  contains the map of the environment and the characteristics of the recognisers and  $\mathbf{w}_j$  is measurement noise. These pose candidates will be used to update the pose information of the pose hypotheses and also potentially initiate new hypotheses.

We make the assumption that all pose candidates can be represented with a mean value,  $\mathbf{z}_j(k)$ , and a covariance

matrix,  $R_j$ . This will limit the kind of features that can be used, or alternatively make the degree of approximation higher for some features.

### C. Data Association and Pose Hypotheses Update

It is crucial that the data association problem is solved, i.e., to determine the correspondence between pose candidates and 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:

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

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 criterion is fulfilled

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

where  $\gamma$  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  $\hat{\mathbf{x}}_i$  denote the pose of the robot, given that  $H_i$  is the correct hypothesis. Then  $\mathbf{z}_j(k)$  is modelled as being generated by

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

The pose hypothesis is updated using the “measurement-update formula”

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

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

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

It is evident that this simple matching scheme in general would be likely to produce false matches since no attempt is made to find an “optimal” feature-hypothesis association across all features and pose hypotheses. However, the fact that creative features and so the corresponding pose hypotheses tend to be located relatively far apart significantly reduces the risk of making false associations. Furthermore, to cope with this problem, should a mismatch occur anyway, track splitting is employed (see Section V).

## IV. HYPOTHESIS PROBABILITY ESTIMATION

**I**N this section we develop a formulation for probability that a given hypothesis is 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,  $r_t$ , indicating that a new feature of type  $t$  has been detected, can—using Bayes's inversion formula—be written as:

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

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

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

where  $N$  is the number of hypotheses. It is clear that  $P(r_t)$  is effectively a scale factor which ensures that the  $P(H_i|r_t)$ 's sum to 1. This implicitly assumes that *one* of the hypotheses is correct which is true only under a closed world assumption<sup>2</sup>. 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 accounting for the fact that hypothesis 1, ...,  $N$  might be incorrect. The introduction of  $H_0$  also has the advantage of generalising the framework since we at 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  $P(r_t)$  we simply normalise so that:

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

The term  $P(r_t|H_i)$  expresses the probability of receiving the report,  $r_t$ , 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:

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

where  $P(r_t|f_t)$  is the probability that a report of type  $t$  is generated given that a feature of type  $t$  is seen by a sensor and  $P(f_t|H_i)$  is the probability that a feature of type  $t$  can be seen given that the robot is at the position described by  $H_i$ . In other words,  $P(r_t|f_t)$  is a model of the reliability of the recogniser extracting feature  $f_t$  from sensor data and  $P(f_t|H_i)$  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 expresses the probability that a feature is detected although there is *no* such feature present, i.e., the probability of a false positive, or a “phantom” feature. Again, however, equation 12 normally assumes a closed world in the sense that it is often assumed that  $P(f_t|H_i)$  is fully specified by the map of the world<sup>3</sup>. To explicitly

<sup>2</sup>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.

<sup>3</sup>Sometimes  $P(r|H_i)$  is directly referred to as “the map”.

distinguish between what is modelled and what is not we expand  $P(f_t, H_i)$  as follows:

$$P(f_t|H_i) = P(f_{tm}|H_i) + P(f_{t-m}|H_i) \quad (13)$$

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

$$P(r_t|H_i) = P(r_t|f_t)[P(f_{tm}|H_i) + P(f_{t-m}|H_i)] + P(r_t|\neg f_t)P(\neg f_t|H_i) \quad (14)$$

From this expression it is seen how accounting for unmodelled features adds an “in-flux” of probability-mass to  $P(r_t|H_i)$ . 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 completely modelled.

Note that we do not explicitly treat the case where modelled features for some reason no longer exist in the world (either because they were removed or because they are occluded). This is because our 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  $P(f_t|H_i)$  we exploit the fact that we have a finite number of well-defined hypotheses. If a match between  $H_i$  and a candidate,  $C_j$ , generated from  $f_t$  has been established, we make the assumption that  $P(f_{t-m}|H_i) = 0$  and use the fact that the likelihood of  $C_j$  being generated from an observation of  $f_t$  given  $H_i$  is given as [23]:

$$L(f_t|H_i) = \frac{1}{(2\pi)^{\frac{3}{2}} |S_{i,j}|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(\hat{\mathbf{x}}_i - \mathbf{z}_j) S_{i,j}^{-1} (\hat{\mathbf{x}}_i - \mathbf{z}_j)^T} \quad (15)$$

To get an estimate of  $P(f_t|H_i)$  we normalise:

$$P(f_t|H_i) = \exp^{-\frac{1}{2}(\hat{\mathbf{x}}_i - \mathbf{z}_j) S_{i,j}^{-1} (\hat{\mathbf{x}}_i - \mathbf{z}_j)^T} \quad (16)$$

When a  $H_i$  has not been matched to any  $C_j$  generated by  $f_t$  we use  $P(f_t|H_i) = P(f_{t-m}|H_i)$  which in our system is a constant given a priori and 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, ...,  $N$  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 de-

tected, thus:

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

where  $M$  is the number of creative features detected. The problem with the sum in this equation is that it is extremely difficult to assess the relevant probabilities, especially for the term with  $i = 0$  since the  $H_0$  hypothesis is not Gaussian. In the implementation, we have therefore omitted the conditioning on  $H_i$  and simply used some conservative constants as estimates for  $P(f_{t-m}|H_i)$  and  $P(\neg f_t|H_i)$ .

The decrease in  $P(H_0)$  upon the receipt of the  $(n+1)$ 'th creative feature,  $\Delta P^n(H_0) = P^n(H_0) - P^{n+1}(H_0)$ , is exactly the amount of probability mass, the hypotheses generated on the basis of the  $(n+1)$ 'th feature carry. When  $P(H_0)$  has decreased below some limit,  $\xi$ , no new hypotheses are spawned.

Furthermore, in order to keep the number of hypotheses low, we delete hypotheses for which  $P(H_i) < \eta$  (and return the probability mass  $P(H_i)$  to  $P(H_0)$ ). 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 pose tracking.

## V. TRACK SPLITTING

Each time a pose candidate is used to update a pose hypothesis, there is a risk of making an association error. This risk grows with the degree of uncertainty in hypothesis and candidate. In order not to lose the true pose hypothesis by updating it incorrectly, each hypothesis is split in two identical copies before the update is done. Each hypothesis receives half of the probability mass of the original one. The updating, according to equation 7, is performed only on the original hypothesis. This procedure is in the literature of target tracking termed track splitting [19], [20]. To keep the number of hypotheses low, we merge hypotheses if they have not moved significantly away from each other (in a Mahalanobis distance sense) after having been split apart. This is identical to testing whether a pose candidate and a pose hypothesis match (equation 5). When merging, the  $P(H_i)$ 's are simply added and the position of the updated copy is used.

## VI. 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 report,  $r_t$ 
Generate pose candidates
Predict new pose for all hypotheses (eq. 2)
loop over all candidates,  $C_j$ ,  $j = 1, \dots, M$ 
  loop over all hypotheses,  $H_i$ ,  $i = 1, \dots, N$ 
    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  $P(H_0) \geq \xi$ )
      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.

## VII. GENERATION OF MOVEMENT COMMANDS

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

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 providing the largest number of features. This node is then selected as the next node to go to.

When the robot fails to move to the next selected node this corresponds to one of Kuipers' rehearsal procedures failing, indicating that the robot is not where it assumes to be. Accordingly, we scale the probability of the currently best hypothesis, on the basis of which the move was planned, with a factor  $P_{ft}$  which is the probability that a move fails although the current hypothesis represents the true pose of the robot.

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 [12] where the actions are selected so as to minimise the expected entropy after moving and sensing. An alternative strategy is to exploit the fact that we have a well-defined set of hypotheses and then select the action that the best discriminates the two most likely hypotheses. While being

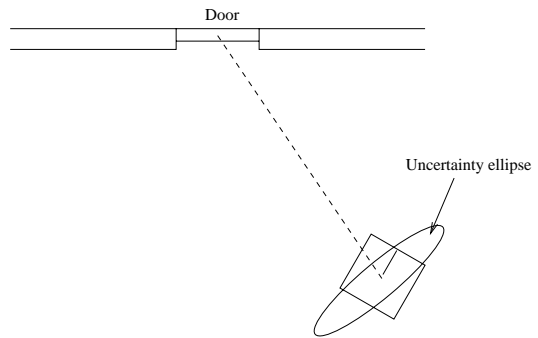


Fig. 2. Creating a pose candidate from a door feature.

much simpler and computationally efficient, new research indicates that this may lead to equally good results [25].

## VIII. EXPERIMENTAL RESULTS

THE framework developed in this work has been tested on two mobile robots in two different environments; at DaimlerChrysler Research and Technology in Berlin, Germany and at the Centre for Autonomous Systems at the Royal Institute of Technology in Stockholm, Sweden.

We performed 20 runs in each place where the robot was positioned at starting points throughout the environment. The starting points were not chosen at random but so as to investigate the strengths and weaknesses of the localisation method.

The constants  $\xi$ , and  $\eta$  where in both places chosen to 0.05, and 0.001 respectively. The probabilities of detecting resp. not detecting a feature were found by making experiments with the recognisers. Therefore these values were not identical at DaimlerChrysler and at the Centre for Autonomous Systems. However, at both places the experiments performed to determine the appropriate values were very simple which is an indication of the fact that this method (similar to many other probabilistic algorithms) is quite insensitive towards small parameter variations.

### A. Features

The features used in the experiments were the following:

#### A.1 Door Features

A door specifies all three degrees of freedom for the robot, that is it gives full pose information. Figure 2 illustrates the idea behind the calculation of a door based pose candidate.

#### A.2 Line Features

Contrary to doors, a line only constrains two of the degrees of freedom for the robot. As Figure 3 shows, the pose of the robot is given to be at a certain distance from the detected line and with a certain orientation. The last degree of freedom has a large uncertainty associated with it. A feature map match is deemed possible if the length,  $l$ , of the detected feature is shorter than or equal to the length,  $L$ , of a map feature (i.e.  $l \leq L$ ). The larger the difference in length ( $L - l$ ), the larger the uncertainty along the line.

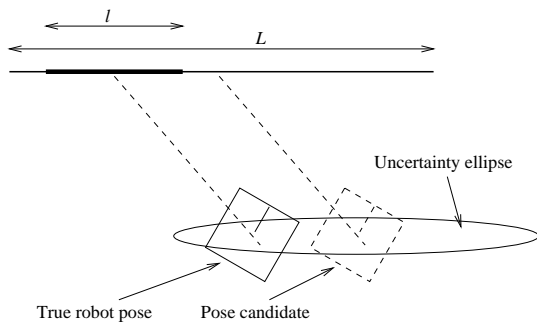


Fig. 3. Creating a pose candidate from a line feature is assumed possible if the map line is longer than the detected line feature ( $L \geq l$ ).

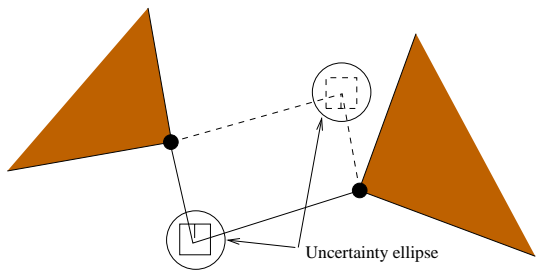


Fig. 4. Creating two pose candidates from a point pair feature.

The pose candidate is assumed to correspond to a match in the middle of the map line. Alternatively, for long lines it may make sense to generate several hypotheses along the line. This has not been done in this work, though.

### A.3 Point Pair Features

Single point features on their own cannot be used, pairs of points, though, give two possible poses only (see Figure 4). Even though the true uncertainty is not Gaussian, we make this simplifying assumption. The point pair features were not used in the experiments at DaimlerChrysler.

## B. Experiments at DaimlerChrysler Research and Technology (DC)

The mobile robot, Clever (see Figure 5), used for the experiments at DC is 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 contained 30 doors, 139 line segments (generally corresponding to walls), and 33 nodes. 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. 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 5 a part of a model including pose hypotheses is shown. The entire model used for the exper-

Run	$\Delta x$ [m]	$\Delta D$ [m]	time [s]	$M_{line}$	$M_{door}$
1	7.56	9.32	211.3	19	10
2	5.13	5.59	41.8	13	11
3 F	-	-	-	-	-
4	7.88	8.50	56.8	30	5
5	13.08	13.62	52.9	20	7
6	6.77	11.43	121.5	20	6
7	15.38	18.05	104.1	32	10
8	4.81	12.38	112.8	27	6
9	8.72	9.91	80.4	34	8
10	0.58	35.12	181.6	79	13
11	10.32	44.72	163.8	55	13
12	9.10	9.63	65.8	42	8
13 F	9.09	17.86	138.4	30	10
14	13.67	16.74	93.6	43	9
15	3.33	22.46	137.6	33	13
16	3.65	34.35	239.8	125	12
17	3.46	18.93	125.7	45	13
18	5.81	6.14	45.8	18	9
19	8.59	36.83	267.1	73	20
20	0.94	1.09	41.9	22	8
avg.	7.26	17.51	120.1	40.0	10.1

TABLE II

RESULTS OF 20 EXPERIMENTS AT DC, BERLIN.  $\Delta x$  IS THE DISTANCE BETWEEN START AND END POINT OF THE LOCALISATION,  $\Delta D$  IS THE TOTAL DISTANCE TRAVELLED. AN F IN THE FIRST COLUMN DENOTES THAT THE LOCALISATION FAILED.

iments is shown in Figure 6.

Typical data for a global localisation attempt is shown in Table I. The data from all of the 20 experiments are listed in Table II. An experiment is ended when the probability of the strongest hypothesis exceeds 0.8 or when the exploration strategy is unable to find any creative features so that no hypotheses are created (failure). A successful run is defined as one where the estimated position, i.e. the centre of the Gaussian hypothesis, is so close to the true position that the normal pose tracking algorithm of the robot can take over. We set that limit to 50 cm and 10 degrees, but it is estimated that the pose in all successful runs was within  $\pm 20$  cm and  $\pm 5$  degrees. From Table II it is seen that the robot successfully localised in 90% of the trials. On average, the robot used 2 minutes and travelled 17.5 meters to perform the localisation. The longer and more time consuming runs were typically those where the robot was started somewhere in the semi-circular hallway (or one of the adjacent offices). This is due to the highly uniform structure of this hallway and the difficulties involved with extracting reliable (straight) line segments. It however turned out that the robot would correctly localise itself as soon as it made it to the end of the hallway where the environment is no longer uniform.

In run no. 3, the robot was placed in a cluttered office and the initial exploration strategy failed to take the robot to a position where a creative feature (i.e. a door) could be

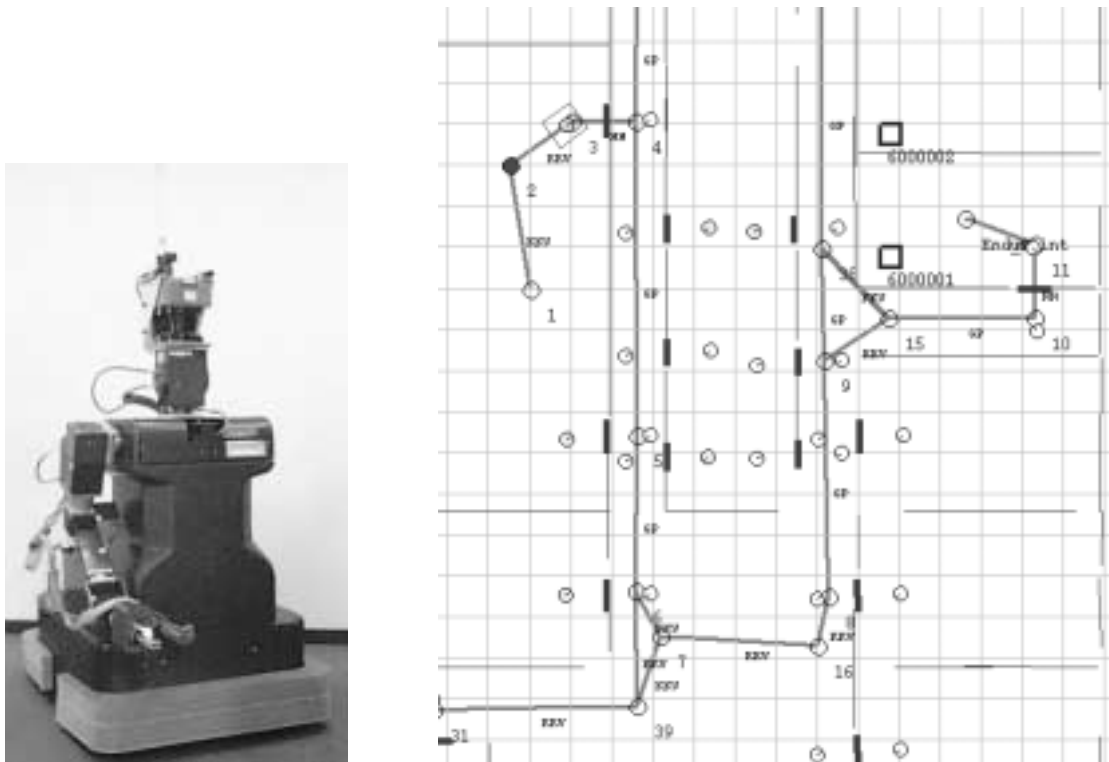


Fig. 5. **Left:** The mobile platform used for the experiments. The robot consists of a re-fitted LabMate base equipped with a 7-DOF Amtec manipulator and a Sick LMS laser scanner plus a set of colour CCD cameras mounted on a pan/tilt unit. The robot has two on-board P200 single board computers and radio Ethernet providing a facility for sending status information to a stationary workstation. **Right:** Part of the world model used for 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 after detecting a door feature. Note that all hypotheses have the same relative pose with respect to a modelled door feature.

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 I

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 CURRENT BEST ESTIMATE.  $P(H_{1st})$  AND  $P(H_{2nd})$  ARE THE PROBABILITIES OF THE BEST AND SECOND BEST HYPOTHESIS, RESPECTIVELY.

seen. This shows one weakness of feature based localisation which is that there might be areas of the environment where there is no information and where an active strategy thus cannot take over the information gathering. It is however doubtful whether a non-feature based localisation strategy would have been able to take the rather large, non-holonomic robot out of the office and subsequently localise it.

In run no. 13 the robot was put in a hallway (on the upper right in Figure 6) which has several non-modelled

doors leading to rooms with technical equipment where the robot should never go. These doors have been left out in the model which was created for office delivery tasks. In reality, this hallway looks very much like the other one running parallel to it, but due to the left out doors, this is not the case in the model. The effect of this is that the robot thought that it was located in the hallway with all doors modelled. What this shows is that although model incompleteness has been accounted for in the probabilistic formulation, this cannot cope with the case where a num-

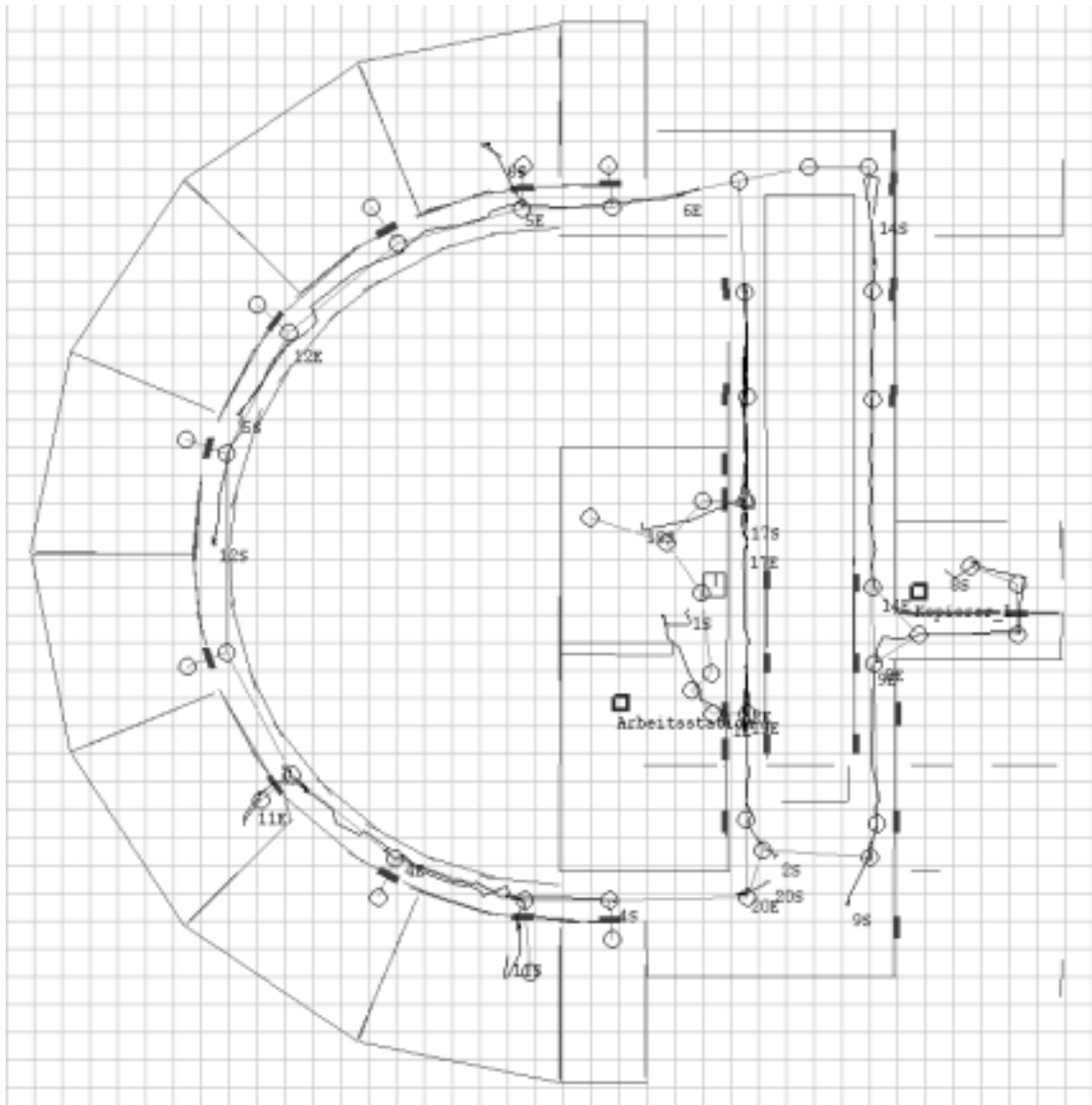


Fig. 6. World model used for the experiments at DC. Grid size is 1 meter. The trajectories followed by the correct hypothesis are plotted (the platform pose during the experiments was unknown). Some trajectories have been omitted to keep the plot intelligible. The start and end points have been labelled with the experiment number followed by an 'S' and an 'E', respectively.

ber of features have systematically been excluded from the model.

### C. Experiments at the Centre for Autonomous Systems (CAS)

At CAS the robot Asterix, a Nomad200, was used to carry out the 20 experiments. Asterix is also equipped with a Sick laser scanner which is mainly used for feature extraction (lines and doors). A ring of sonar sensors provides information for local navigation as well as extraction of features (point landmarks).

Figure 7 shows the world model that covers 900 m<sup>2</sup> and consists of 136 lines, 108 doors and 738 point landmarks. The point landmarks have been combined in pairs to give a total of 9578 point landmark pairs (see Section VIII-A.3). It should be pointed out that a physical door is represented by one door from each direction it can be observed. The

108 doors in the world model correspond to approximately physical 70 doors (the difference corresponds to the number of doors leading to unmapped area). As can be seen from Figure 7 each room is modelled as a rectangle, even though many of the rooms in reality are highly non-rectangular. The rectangles and the positions of the doors have been measured by hand. The sonar point landmarks on the other hand were collected by driving the robot through the environment with a joystick.

The world map was not, like the one at DC, built around a graph with nodes and arcs. It does however contain a set of goal-points which the robot can be directed to during missions. A simple algorithm was therefore designed to join nearby goal-points with arcs to generate a graph. Information about what features can be seen from where was approximated using the distance to the features and the angle at which it was observed (doors).

Table III summarises the results from the experiments at CAS. The robot localised successfully in 95% of the runs. Localisation at CAS took on average 4.5 minutes under which the robot on average travelled 19 meters. As we can see there is a significant difference in the time it took to localise in the two environments, which can be explained by a higher density of features (mainly doors) at DC and a faster moving robot. In Figure 7 the trajectories that the robot followed during exploration are shown.

In run no. 4 the robot was placed in a small office on the lower floor (see Figure 7). This office is very cluttered, leaving the robot very limited room to manoeuvre, thus making the initial exploration very slow. To be able to disambiguate the office from the office next to it, the robot must reach the corridor, which in this experiment took 330 seconds.

In run no. 10 the experiment was performed in the room lying on its own on the upper region on the lower floor. This room is identical in size to the room a bit further down on the map, where the robot reported being. The true hypothesis successfully matched 0 pairs out of 20, 6 lines out of 14 and 2 doors out of 3. The winning, false, hypothesis matched 3 out of 20 pairs, 12 out of 14 lines and 1 out of 3 doors. The reason for the bad matching result for the true hypothesis is that this room has been completely restructured since the map was constructed. The furniture has been rearranged completely which effects the point landmark pairs and a wall that was previously covered with cupboards have now been uncovered, resulting in one of the sides in the rectangular line model not matching anymore. This is similar to what happened in run no. 13 at DC.

In run no. 15 the robot started on the lower end in the map on the upper floor and travelled upwards, not being able to detect any of the (at the time) closed doors on the left hand side of the corridor. It was not until the robot went into the kitchen in the middle of the corridor that the true pose could be singled out.

In the last experiment, run no. 20, the robot was placed in the almost feature-less left hand side corridor on the lower floor, only being able to detect one of the doors in this corridor (the others were closed). Acting according to the greedy exploration strategy the robot travelled upwards in the map to the region where no new features were found. It was not until it had been all the way up to the last node that it went back and localised close to the turn to the other corridor bit. The 10 strongest hypotheses all matched the same number of lines and doors, highlighting the high degree of symmetry. However, the point landmark pairs eventually singled out the true hypothesis. Had the robot initially travelled down towards the other corridor, it would have been able to localise faster and with less ambiguity.

## IX. DISCUSSION, CONCLUSION, AND FURTHER WORK

**I**N 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

Run	$\Delta x$ [m]	$\Delta D$ [m]	time [s]	$M_p$	$M_l$	$M_d$
1	3.79	7.84	129.5	7	13	2
2	4.88	15.66	238.1	11	8	3
3	10.40	15.99	244.0	24	13	3
4	4.96	9.62	354.4	5	8	2
5	12.05	22.25	293.6	11	19	4
6	6.44	20.01	409.0	6	11	3
7	4.60	29.56	395.9	19	24	2
8	5.21	6.74	121.5	14	12	3
9	12.93	22.22	280.5	19	35	4
10 F	4.07	12.77	287.1	3	12	1
11	4.68	13.14	274.1	7	10	2
12	13.08	14.25	246.5	7	15	5
13	3.61	7.19	183.6	9	6	3
14	7.59	13.56	252.6	22	6	3
15	31.42	42.78	536.1	19	26	3
16	5.85	15.90	236.1	7	11	2
17	15.83	30.74	473.1	13	21	3
18	5.02	7.74	168.6	8	8	5
19	4.73	10.47	224.6	3	9	3
20	6.87	58.17	362.5	27	77	2
avg.	8.4	18.8	285.6	12	17	3

TABLE III

RESULTS FROM EXPERIMENTS AT CAS.  $\Delta x$  IS THE DISTANCE BETWEEN START AND END POINT OF THE EXPLORATION, WHEREAS  $\Delta D$  IS THE TOTAL DISTANCE TRAVELLED. RUNS 6, 7, AND 20 ILLUSTRATE THAT THE DIFFERENCE BETWEEN THESE TWO DISTANCE MEASURE CAN BE QUITE LARGE. AN F IN THE FIRST COLUMN DENOTES THAT THE LOCALISATION FAILED.

Kalman filtering of Gaussian pose hypotheses, for short Multi Hypothesis Localisation (MHL).

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 computationally efficient since the computational demand grows linearly with the number of hypotheses and is independent of the size of the environment.

The drawbacks of using MHL 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 (with a finite number of hypotheses). We believe the latter problem to be of purely theoretical importance and the former was solved using standard techniques from MHT

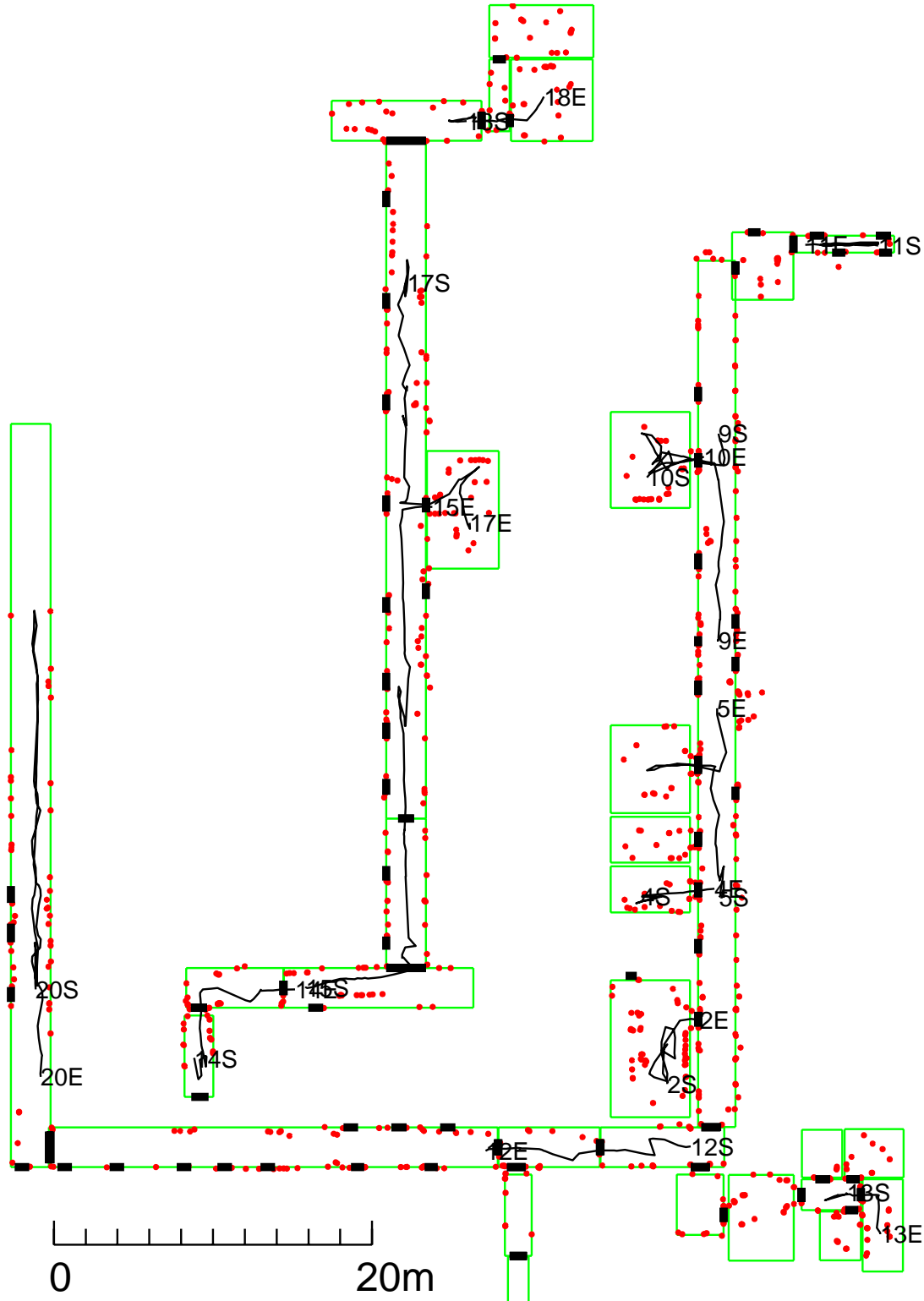


Fig. 7. The world model used for the experiments at CAS. The two disjoint parts belong to two different floors, connected through stairs and two elevators. The middle part is the upper floor. Three types of features can be seen in the figure, lines, doors and sonar based point features. Each room in the map is modelled as a rectangle, capturing only the large scale structure of the environment. These rectangles result in four lines per room. Doors are marked as thick lines. Point landmarks extracted from sonar sensor data can be seen as the dots scattered throughout the building, mostly along the walls. The points are combined to form pairs (not shown). The true trajectories of the robot are plotted in the figure. Some trajectories have been omitted to keep the plot intelligible. The start and end points have been labelled with the experiment number followed by an 'S' and an 'E', respectively.

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. It was chosen to do this using 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 (equation 11). 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. This clearly eases 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.

A comparison between MHL and the recent Monte Carlo Localization (MCL) technique [15], [26] is also warranted. MCL is easier to implement, but MHL provides a more intuitive representation with a small number of pose hypotheses. It is not as easy to employ active exploration strategies for MCL as it is for MHL. MCL has the advantage of being able to incorporate all kinds of data, whereas MHL requires that a Gaussian approximation is possible. MHL is better at taking advantage of strong features. In the initial phase MCL must explicitly model the global uncertainty by distributing samples throughout the environment thereby spending massive amount of computations, whereas the zero-hypothesis of MHL takes care of this effortlessly.

The MHL method was implemented and tested on two real mobile robots and was found to work according to the expectations. When the 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 due to the fact that the model in terms of its features looked significantly different than the real world. It is however important to point out that the feature based approach is very robust towards “random” changes which is exemplified by the fact that no furniture whatsoever was present in the model used for the DC experiments and that the rooms in the CAS model were approximated with rectangles.

On the basis of these experiments it is however not possible to say much about the lower limit for the number of (modelled) features that have to be present in the environment, and how densely these have to be placed. This will in general depend on how symmetric the environment is and on the quality of the platform’s odometry. What matters is that the robot does not have to travel so far between recognising new features that the uncertainty on the pose

hypotheses makes them all good candidates for a match. In a sparsely but uniformly populated environment this may not be a problem since the pose hypotheses will also be far apart, but if there are distant “clusters” of features this might lead to erroneous results.

The MHT and the Bayesian evidence fusion has been found to work flawlessly, using very limited computational resources.

We believe that it is important to have an exploration strategy since moving randomly is not likely to lead to a successful localisation due to the odometry drift of the platform. Not being the focus of this work we implemented a quite simple, greedy-type exploration strategy which however turned out to be sufficient for the purpose. The important point here is that due to the fact that we have a limited number of well-defined hypotheses, many methods proposed in the literature become computationally tractable and thus easy to employ.

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

#### ACKNOWLEDGEMENT

THIS research has been sponsored in part by the Swedish Foundation for Strategic Research through the Centre for Autonomous Systems. The funding is gratefully acknowledged. We would also like to thank the reviewers for their constructive comments.

#### REFERENCES

- [1] J. Borenstein, B. Everett, and L. Feng, *Navigating Mobile Robots: Sensors and Techniques*, A. K. Peters, Ltd., Wellesley, MA, 1995.
- [2] 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*, vol. 8, pp. 47–63, 1991.
- [3] 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*, Edinburgh, UK, July 1998, pp. 93–100.
- [4] J.J. Leonard and H.F. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.
- [5] 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.
- [6] 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)*, Philadelphia, PA, April 1988, IEEE, pp. 1407–1413.
- [7] 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*, Leuven, Belgium, May 1998, IEEE, pp. 3121–3128.
- [8] Sven Koenig and Reid G. Simmons, “Xavier: A robot navigation architecture based on partially observable markov decision process models,” in *Artificial Intelligence and Mobile Robots*,

- David Kortenkamp, R. Peter Bonasso, and Robin Murphy, Eds., pp. 91–122. AAAI Press/The MIT Press, 1998.
- [9] Reid Simmons and Sven Koenig, “Probabilistic robot navigation in partially observable environments,” in *Proc. of the 14th IJCAI*, Montreal, Canada, August 1995, pp. 1080–1087.
- [10] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, “Partially observable markov decision processes for artificial intelligence,” in *Reasoning with Uncertainty in Robotics*, L. Dorst, M. van Lambalgen, and F. Voorbraak, Eds., Berlin, 1996, Lecture Notes in Artificial Intelligence 1093, pp. 146–162, Springer.
- [11] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlingshaus, D. Hennig, T. Hofmann, M. Krell, and T. Schmidt, “Map learning and high-speed navigation in RHINO,” in *Artificial Intelligence and Mobile Robots*, David Kortenkamp, R. Peter Bonasso, and Robin Murphy, Eds., pp. 21–52. AAAI Press/The MIT Press, 1998.
- [12] D. Fox, W. Burgard, and S. Thrun, “Active markov localization for mobile robots,” *Robots and Autonomous Systems*, vol. 25, no. 3-4, pp. 195–207, 1999.
- [13] Wolfram Burgard, Dieter Fox, and Sebastian Thrun, “Active mobile robot localization by entropy minimization,” in *Proc. of the 2nd Euromicro Workshop on Advanced Mobile Robots*. IEEE/CS, 1997.
- [14] Steen Kristensen, “Sensor planning with Bayesian decision theory,” in *Reasoning with Uncertainty in Robotics*, L. Dorst, M. van Lambalgen, and F. Voorbraak, Eds., Berlin, 1996, Lecture Notes in Artificial Intelligence 1093, pp. 353–367, Springer.
- [15] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte carlo localization—efficient position estimation for mobile robots,” in *Proceedings of AAAI’99*, 1999.
- [16] S. Thrun, W. Burgard, and D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping,” in *Proc. of the IEEE International Conference on Robotics and Automation*, San Francisco, CA, April 2000, IEEE, pp. 321–328.
- [17] Jens-Steffen Gutmann, Wolfram Burgard, Dieter Fox, and Kurt Konolige, “An experimental comparison of localization methods,” in *Proc. of the International Conference on Intelligent Robots and Systems (IROS’98)*, 1998, vol. 2, pp. 736–743.
- [18] Patric Jensfelt and Steen Kristensen, “Active global localisation for a mobile robot using multiple hypothesis tracking,” in *Proc. of the IJCAI-99 Workshop on Reasoning with Uncertainty in Robot Navigation*, Stockholm, Sweden, Aug. 1999, pp. 13–22.
- [19] Yaakov Bar-Shalom and Xiao-Rong Li, *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, Norwood, MA, 1993.
- [20] Yaakov Bar-Shalom and Xiao-Rong Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, YBS, 1995.
- [21] Johannes Reuter, “Mobile robot self-localization using PDAB,” in *Proc. of the International Conference on Robotics and Automation*, San Francisco, CA, April 2000, IEEE, vol. 4, pp. 3512–3518.
- [22] Stergios I. Roumeliotis and George A. Bekey, “Bayesian estimation and Kalman filtering: A unified framework for mobile robot localization,” in *Proc. of the International Conference on Robotics and Automation*, San Francisco, CA, April 2000, IEEE, vol. 3, pp. 2985–2992.
- [23] 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*, Las Vegas, Nevada, October 1994, IEEE, pp. 17–26.
- [24] S. Thrun, “Probabilistic algorithms in robotics,” *AI Magazine*, vol. 21, no. 4, pp. 93–109, 2000.
- [25] 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.
- [26] Patric Jensfelt, Olle Wijk, David Austin, and Magnus Andersson, “Experiments on augmenting condensation for mobile robot localization,” in *Proc. of the International Conference on Robotics and Automation (ICRA’00)*, San Francisco, CA, USA, May 2000, vol. 3, pp. 2518–2524.