Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

ELSEVIER

Robotics and Autonomous Systems 44 (2003) 41–53

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Feature-based multi-hypothesis localization and tracking using geometric constraints

Kai O. Arras[a,*], José A. Castellanos[b], Martin Schilt[a], Roland Siegwart[a]

[a] Autonomous Systems Lab, Swiss Federal Institute of Technology Lausanne (EPFL), CH-1015 Lausanne, Switzerland
[b] Robotics and Real-Time Group, Centro Politecnico Superior, Universidad de Zaragoza, E-50015 Zaragoza, Spain

Abstract

Mobile robot localization deals with uncertain sensory information as well as uncertain data association. In this paper we present a probabilistic feature-based approach to global localization and pose tracking which explicitly addresses both problems. Location hypotheses are represented as Gaussian distributions. Hypotheses are found by a search in the tree of possible local-to-global feature associations, given a local map of observed features and a global map of the environment. During tree traversal, several types of geometric constraints are used to determine statistically feasible associations. As soon as hypotheses are available, they are tracked using the same constraint-based technique. Track splitting is performed when location ambiguity arises from uncertainties and sensing. This yields a very robust localization technique which can deal with significant errors from odometry, collisions and kidnapping. Experiments in simulation and with a real robot demonstrate these properties at low computational costs.

Keywords: Mobile robot localization; Multi-hypothesis tracking; Geometric constraints; Kalman filtering; Data association

1. Introduction

Kalman filter-based position tracking with geometric features [1,6,9,16] has been proven to be a very powerful localization technique with several desirable properties: It operates with minimalistic environment representations, it is robust with respect to environment dynamics and combines unbounded localization accuracy with light-weight implementations.

Clearly, position tracking using an extended Kalman filter (EKF) is a local localization technique with the typical risk of loosing the track and going lost. This is in contrast to the POMDP or Markov approach to localization [12,18,20] which maintains a probability distribution over a topology of nodes, previously overlaid onto the environment. Within this graph the robot can never go lost as long as a location probability is maintained for each node. In this manner, arbitrary densities can be represented in order to cope with the problem of location ambiguity. Recently, new approaches which overcome limitations of earlier methods have been proposed [10,15]. They employ the principle of particle filters where the density function of the robot location is approximated by a set of randomly drawn samples. However, all these techniques maintain constantly a big number of hypotheses which in the case of particle filters has to be carefully weighted, updated and re-distributed. The ability of these techniques to properly react to location ambiguity from environment or sensing is due to the quantity of samples and a distribution strategy which must be appropriately chosen.

* Corresponding author.
E-mail address: kai-oliver.arras@epfl.ch (K.O. Arras).

0921-8890/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0921-8890(03)00009-5

Unlike these methods which can be denoted *location-driven*, our approach to global localization will be *feature-driven*. It reacts directly to the environment in the sense that features tell us *when* and *where* to place a location hypothesis—not an a priori topological graph or a dynamically maintained sample set. This allows to maintain exactly as many hypotheses as necessary and as few as possible. The technique which provides this property is a constraint-based search in an interpretation tree [6,11,13,17]. This tree is spanned by all possible local-to-global associations, given a local map of observed features L and a global map of model features G. We further present a constraint-based tracking splitting filter which employs the same technique for hypothesis tracking. Hypothesis generation and tracking constitute together a framework for global EKF localization.

Earlier work [8] deals with multiple hypotheses for map building. Using segments and corners from ultrasonic sensors, their hypotheses model a *typological* feature ambiguity since the feature type was difficult to distinguish. We believe that with today sensors (laser and vision) feature extraction can be made very reliable and that rather *spatial* feature ambiguity is an issue to address. Other approaches like [21] propose hybrid models to combine advantages from the EKF and POMDP-worlds. However, [21] requires a clearly structured room-hallway topology since otherwise the approach reduces to an EKF technique with the known limitations.

A feature in this context is a geometric primitive containing at least one geometric measure such as angle, range, (x, y)-position or (x, y, θ)-pose. They are models for physical objects in the environment such as doors, walls, corners, columns, or even fire extinguishers. Figures will use point-, angle- and line features for illustration. The approach is, however, completely general with respect to the feature type.

1.1. Motivation and problem statement

With EKF-based position tracking using features,[1] the cause of a lost situation is virtually always an incorrect data association. After extensive experiments with this localization technique on more than 100 km travel distance with three different robots [1,2], we locate the predominant reasons for false associations as follows:

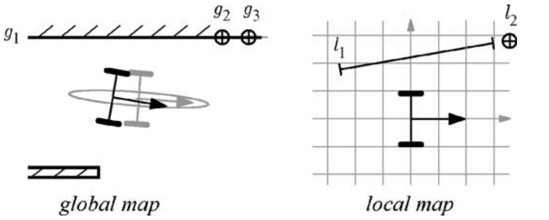

Fig. 1. A situation where the robot goes lost and where this is very difficult to detect: when the vehicle arrives at the end of a corridor with a critical amount of accumulated odometry drift (predicted pose in gray, true pose in black), the local point feature $\{l_2\}$ is wrongly matched even if the uncertainty models are correct. Instead of the pairing $\{l_2, g_2\}$, the wrong pairing $\{l_2, g_3\}$ is produced.

- *Heavy violations of system and system noise models*. Collisions or significant odometry drift in directions which were not correctable by the observations (Fig. 1).
- *Feature discriminance*. Low feature discriminance is spatial sensing ambiguity on the level of extracted features and expresses itself as proximity in the feature's parameter space (Fig. 2).

In practice, single hypothesis tracking can often recover a robot which went lost due to non-discriminant features, since they typically yield close-to-the-truth pose estimates. But in general, both problems, especially in simultaneous occurrence, can lead to false associations and irrecoverable lost situations. Robust localization must therefore address the data association problem. Association ambiguity occurs locally, during tracking, and globally after a lost situation or kidnapping. We present two structurally identical algorithms which rely on the same idea for each problem forming thus a consistent approach to localization.

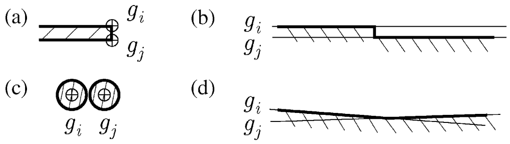

Fig. 2. Examples of feature types which are typically subject to low feature discriminance: (a) angle features modeling corners, (c) point features modeling columns, (b) and (d) line features modeling walls. Less critical are features of higher parameter dimensionality as segments or circles or features of natural discriminance as doors.

[1] We will use the terms *location*, *position* and *pose* interchangeably. They denote all the full (x, y, θ) vehicle pose.

2. Hypothesis generation

Robot location ambiguity is represented by multiple Gaussian location hypotheses. We employ data association on a discrete feature-to-feature basis: A *pairing* $p_{ij} = \{l_i, g_j\}$ is the association of the measurement l_i with g_j saying that l_i and g_j denote the same physical object in the environment (g_j is called an *interpretation* of l_i). The local map of observed features $L = \{l_i\}_{i=1}^{p}$ and the global map of model features $G = \{g_j\}_{j=1}^{m}$ span the search space of all possible data associations which has the structure of a tree with p levels and $m + 1$ branches [13]. p is the number of observed features in L, m the number of modeled feature in G. The extra branch (called *star branch*) allows associations in the presence of outlier observations (false positives) and thus accounts for environment dynamics and map errors. During tree traversal, statistically feasible pairings are sought given all uncertainties associated to the features. In order to test a potential pairing, geometric constraints are applied. Although the problem is of exponential complexity, geometric constraints reduce enormously the space to be explored. They allow to discard whole subtrees each time when an incompatible pairing is found at the root of such a subtree. With the uncertainties associated to the local and global features, all decisions make use of the Mahalanobis distance on a significance level α. If a compatible pairing is found (compatible on the level α), it is added to the *supporting set* $S_h = \{\{l_1, g_{j_1}\}, \{l_2, g_{j_2}\}, \dots, \{l_p, g_{j_p}\}\}$. The supporting set and the robot location, denoted $L_h = (x, P)$ with x and P being the first and second moments, form a location hypothesis $h = \{L_h, S_h\}$. All hypotheses h_i together make up the set of hypotheses $H = \{h_i\}_{i=1}^{n}$.

2.1. Geometric constraints

We can classify geometric constraints into two categories: Location independent constraints can be validated without having an estimation of the robot location. They include *unary* and *binary* constraints.

Unary constraints apply on intrinsic properties of a feature. Examples are feature type, color, texture or dimension such as length or width. Unary compatibility is directly found by comparison (function `satisfy_unary_constraints`). They are powerful since whole subspaces can be excluded from the search beforehand. Example: with line segments, unary compatibility is satisfied if the length of the observed segment l_i is smaller or equal than the length of the modeled segment g_j.

Binary constraints always apply to the features of two pairings and test on relative measures such as angle or distance. Binary constraints are used to validate whether two local features are consistent with two global features (function `satisfy_binary_constraints`). Example: l_i and l_k are lines with the intermediate angle φ_{ik}. Then, the pairing p_{kl} is considered compatible if the angle φ_{jl} is identical. With point features, for instance, the distances l_i–l_k and g_j–g_l must correspond.

The second category are location dependent constraints which come into play as soon as a robot location L_h is available.

Visibility constraints only apply to model features. It tests whether g_j is visible from L_h. Non-visibility can be due to feature properties as relative view direction or due to sensor limitation as maximal range or resolution. Example: lines or segments can always be seen only from one side. If the robot is behind a wall, one of the two lines modeling that wall is invisible and can be discarded from further consideration.

Rigidity constraint. A pairing p_{ij} is considered compatible if l_i and g_j, transformed into the same coordinate system given L_h, coincide (are at the same position). This is what commonly happens in the matching step of an EKF localization cycle. Usually, g_j is transformed into the frame of l_i.

Extension constraints test whether an observed feature is fully contained in the candidate model feature (they completely overlap). This is relevant for features like line segments or circular arcs whose observations can be smaller than the model features in some sense.

2.2. The search algorithm

Tree traversal is realized as a recursive back-tracking search (Algorithm 1, [7]). The strategy of Algorithm 1 is to first find a minimal number of pairings with location independent constraints such that a robot location can be estimated and location dependent constraints can be applied too (part *B*).

When an observation is selected from the local map (function `select_observation`), optional rules can be applied to choose an observation which gen-

```
function generate_hypotheses( $h$, $L$, $G$ )

  if $L = \{\}$ then
    $H \leftarrow H \cup \{h\}$
  else
    $l \leftarrow$ select_observation( $L$ )
    for $g \in G$ do
      $p \leftarrow \{l, g\}$
      if satisfy_unary_constraints( $p$ ) then
        if location_available( $h$ ) then
   A      $accept \leftarrow$ satisfy_location_dependent_cnstr( $L_h$, $p$ )
          if $accept$ then
            $h' \leftarrow h$
            $S_{h'} \leftarrow S_h \cup \{p\}$
            $L_{h'} \leftarrow$ estimate_robot_location( $S_{h'}$ )
          end
        else
          $accept \leftarrow$ true
          for $p_p \in S_h$ while $accept$
            $accept \leftarrow$ satisfy_binary_constraints( $p_p$, $p$ )
          end
          if $accept$ then
            $h' \leftarrow h$
   B        $S_{h'} \leftarrow S_h \cup \{p\}$
            $L_{h'} \leftarrow$ estimate_robot_location( $S_{h'}$ )
            if location_available( $h'$ ) then
              for $p_p \in S_h$ while $accept$
                $accept \leftarrow$ satisfy_location_dependent_cnstr( $L_{h'}$, $p$ )
              end
            end
          end
        end
        if $accept$ then
          $H \leftarrow H \cup$ generate_hypotheses( $h'$, $L \backslash \{l\}$, $G$ )
        end
      end
    end
    $H \leftarrow H \cup$ generate_hypotheses( $h$, $L \backslash \{l\}$, $G$ )
  end

return $H$
end
```

Algorithm 1. Given a local map L and the global map G, the algorithm returns the set of generated location hypotheses H.

erates as few pairings as possible. As soon as a robot location estimate is available (function `location_available`), the algorithm applies location dependent constraints (`satisfy_location_dependent_cnstr`). If a new acceptable pairing is found, it is added to the supporting set $S_h = \{p_1, p_2, \dots, p_p\}$, the location estimate is refined (function `estimate_robot_location`) and the function recurs (part A).

Each time when the algorithm reaches the bottom of the tree, all observed features l_i have been assigned, either to a model feature g_j or to the star branch. Then, we have a valid robot location hypothesis which can be added to H. In the beginning, with $H = \{\}$ and a given, non-empty L and G, h, is not needed. It appears, however, as an argument of `generate_hypotheses` since the algorithm is recursive. The proper h to start with has $S_h = \{\}$ and L_h such that `location_available` returns false.

Note that the significance level α is the only parameter the user has to specify. It decides on acceptance or rejection of pairing candidates.

2.2.1. Estimating the robot location

Given a supporting set S_h, the robot position L_h can be estimated using the EKF. The Kalman filter is, however, a recursive formulation, well suited for tracking applications where there is always an a priori state estimate. For the case of hypothesis generation where no a priori position is available, an adequate reformulation of the EKF is the extended information filter (EIF). The EIF is a batch estimator and resembles directly the weighted mean (refer to [4] for derivation and details).

Let v denote the stacked innovation vector of all pairings $\{l_i, g_{j_i}\}$ and R its associated covariance matrix. Let further ∇h be the $q \times 3$ Jacobian matrix of the linearized feature measurement model (the frame transform) with respect to the robot position. q is the number of observations which is the number of observed features p times their number of feature parameters r. Then the EIF is as follows:

$$P^{-1}(k+1|k+1) = P^{-1}(k+1|k) + \nabla h^{\mathrm{T}} R(k+1)^{-1} \nabla h, \qquad (1)$$

$$\hat{x}(k+1|k+1) = P(k+1|k+1)[P^{-1}(k+1|k) \cdot \hat{x}(k|k+1) + \nabla h^{\mathrm{T}} R(k+1)^{-1} \nabla h \cdot \xi(k+1)], \qquad (2)$$

where $\xi(k+1)$ is a $3 \times q$-matrix such that

$$\nabla h \cdot \xi(x+1) = v(k+1). \qquad (3)$$

Assigning zero weight to the odometry-based state prediction can be elegantly done by setting its inverse—the information matrix—to zero

$$P^{-1}(k+1|k) = 0_{3\times 3}. \qquad (4)$$

By substituting Eq. (4) into Eqs. (1) and (2) and using (3), we obtain a conventional equation system where we can easily see that dependent on q, being greater or smaller than three, the system is over- or underdetermined.

$$\nabla h \cdot \hat{x}(k+1|k+1) = v(k+1). \qquad (5)$$

The solution of (5) is obtained via the pseudoinverse

$$\nabla h' = (\nabla h^{\mathrm{T}} \nabla h)^{-1} \nabla h^{\mathrm{T}} \qquad (6)$$

where we can distinguish between $\nabla h^{\mathrm{T}} \nabla h$ being singular or non-singular. In the latter case, the equation system (5) has a unique solution in the least square sense (`location_available` returns true). In the former case, only a non-unique pose estimate with infinite number of solutions is returned (`location_available` returns false).

3. Hypothesis tracking

With a localized robot doing pose tracking, data association ambiguity can arise as discussed in Section 1.1. In such a situation there are several statistically feasible pairing candidates for an observation. The closest one (in a Mahalanobis sense) is not necessarily the correct one. Choosing it when it is the wrong candidate will lead to filter inconsistency and likely to filter divergence. This is the most widely applied strategy, called nearest neighbor standard filter (NNSF).

Here, we will pursue another strategy. As soon as there is association ambiguity, that is, there is no guarantee anymore for the correct association to be found, we will re-generate hypotheses locally (Fig. 3). This is what `track_hypothesis` does, given a predicted location, a local and a global map. It splits up into multiple offspring hypotheses if statistical compatibility with several supporting sets can be established at that location. This strategy is also known as the track splitting filter [5]. Here we perform track splitting under geometric constraints which bound the number of possible tracks. Algorithm 2 has the identical structure than Algorithm 1 but employs location dependent constraints *only* and does *not* recur with a refined position estimation. In this manner the algorithm finds all supporting sets in the vicinity of the initially predicted location L_h and returns them in form of a hypothesis set H_t. Again, the second recursion call implements the extra branch in the interpretation tree that allows correct associations in the presence of outlier observations and map errors. Since L_h is not reestimated, `track_hypothesis` will generate hypotheses in a region whose size depends on the uncertainty of L_h. If L_h is certain, track splitting is unlikely to occur. If it is uncertain, feasible supporting sets will be found in a larger area around L_h (see also Fig. 4).

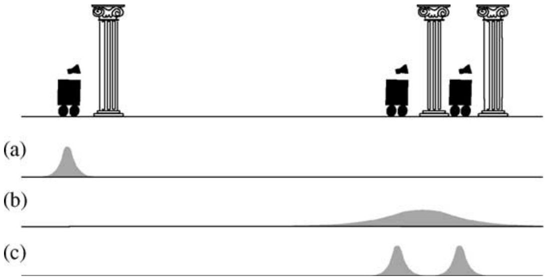

Fig. 3. The idea behind the position tracking Algorithm 2: A well localized robot in (a) moves and observes a *single* feature in (b) where it is impossible to say which is the correct pairing in view of the uncertainties. Instead, the hypothesis splits up in (c) representing thereby all possible pairings at that location. The two hypotheses are tracked using location dependent constraints until a single one remains.

After `track_hypothesis` has been applied for each h_i in H, we can distinguish the three cases verification, falsification and division:

```
function track_hypothesis( $h$, $L$, $G$ )

 if $L = \{\}$ then
   $H_t \leftarrow H_t \cup \{h\}$
 else
   $l \leftarrow$ select_observation( $L$ )
   for $g \in G$ do
     $p \leftarrow \{l, g\}$
     if satisfy_unary_constraints( $p$ ) then
       if satisfy_location_dependent_cnstr( $L_h$, $p$ ) then
         $S_{h_p} \leftarrow S_h \cup \{p\}$
         $H_t \leftarrow H_t \cup$ track_hypothesis( $h_p$, $L \setminus \{l\}$, $G$ )
       end
     end
   end
   $H_t \leftarrow H_t \cup$ track_hypothesis( $h$, $L \setminus \{l\}$, $G$ )
 end

return $H_t$
end
```

Algorithm 2. Given the local map L, the global G and the hypothesis h to be tracked at location L_h, the algorithm returns the set of tracked hypotheses H_t.

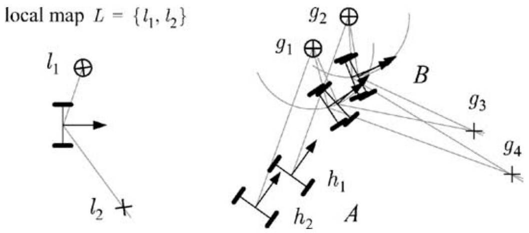

Fig. 4. Hypothesis duplication. Given a local map with a (x, y)-point feature l_1, and an angle-only feature l_2, hypotheses h_1, h_2 split up each into four offsprings after a (very) uncertain movement A to B. This results in eight hypotheses at B, four of them being redundant.

- $|H_t| = 1$ *hypothesis verification*. The hypothesis h_i is confirmed. This is the estimation step in an EKF localization cycle. Given the supporting set S_{h_i}, the robot location is estimated and h_i is admitted to the new H.
- $|H_t| = 0$, *hypothesis falsification*. The hypothesis cannot be held any more by location dependent constraints on the significance level α. It gets rejected.
- $|H_t| > 1$, *hypothesis division*. The track of hypothesis h_i splits up into several offspring hypotheses $\{h_{i,1}, h_{i,2}, \dots , h_{i,o}\}$ which all can be held by location dependent constraints at the predicted robot location. The robot locations are estimated with the EIF using their respective supporting set.

With multi-hypothesis localization, there is no strict distinction of being localized and being lost. There are three cases which can be characterized by n, the number of hypotheses in H: being *lost* is expressed as $n = 0$ (without any valid hypotheses), *not localized* means that there is unresolved location ambiguity, $n > 1$, and being *localized* is simply expressed as having a single location hypothesis, $n = 1$.

3.1. Hypothesis elimination during tracking

When an uncertain hypothesis splits up, it can happen that *duplicate hypotheses* are produced. This is shown in Fig. 4, where two hypotheses h_1, h_2 split up and produce each four hypotheses. If these duplicates are not eliminated, H will contain redundant information, and thus undermining our intent to reach and maintain $n = 1$.

3.1.1. Duplicate detection

Two hypotheses h_i, h_j are identical if they contain the same piece of information which in our case is identical location. Identical location is due to identical supporting sets

$$h_i \equiv h_j \Leftrightarrow (S_{h_i} = S_{h_j}). \tag{7}$$

This condition is further to be generalized with the distinction of a unique (ℓ_{h_i} is true) and a non-unique (ℓ_{h_i} is false) robot location estimate. In the latter case the current observation contains not enough information to uniquely estimate a robot position (e.g. robot observes a single angle-only feature). Then, the EIF is underdetermined and will return an infinite number of solutions. These solutions denote a degree of freedom in the robot position. Along this degree of freedom, condition (7) is unable to distinguish duplicate hypotheses because several distinct hypotheses can be aligned to the same model feature. We therefore add a distance condition along this degree of freedom. Let x_{h_i}, x_{h_j} and P_{h_i}, P_{h_j} be the first and second moments of L_{h_i} and L_{h_j}, respectively, then 'closeness' is defined by means of the Mahalanobis distance $d_{h_i h_j}$. Thus

$$d_{h_i h_j} = (x_{h_i} - x_{h_j})(P_{h_i} + P_{h_j})^{-1}(x_{h_i} - x_{h_j})^{\mathrm{T}}, \tag{8}$$

$$h_i \equiv h_j \Leftrightarrow \begin{cases} (S_{h_i} = S_{h_j}) & \ell_{h_i}, \\ (S_{h_i} = S_{h_j}) \wedge (d_{h_i h_j} < \chi^2_\alpha) & \neg\ell_{h_i}, \end{cases} \tag{9}$$

with χ^2_α the value chosen from a χ^2-distribution with three degrees of freedom at the level α.

3.1.2. Duplicate rejection

Unlike the Bayesian approach to data association, hypotheses generated with our method do not have an individual probability. They are equally plausible robot locations since they satisfy their uncertain geometric relationships on the same given significance level α.

Hypotheses differ, however, in their support by paired features and their geometric quality. The former is measured by the number of valid (non-star-branch) associations p' in S_h and the latter by the joint Mahalanobis distance. The joint Mahalanobis distance is like the Mahalanobis distance (8) except that it applies not only to a single pairing but sums up over the whole supporting set including correlations. It accumulates the weighted squared error distances (residuals), and thus is a goodness-of-fit measure.

The best hypothesis is the one maximizing p' and, in case of a tie in p', minimizing the joint Mahalanobis distance. In other words, we choose the hypothesis which has most paired features and, from its location, satisfies best the rigidity constraint.

4. Experiments

In the simulation experiment, odometry employs two error models (see below) whereas observations and model features receive a typical, constant and uncorrelated uncertainty. In the beginning, the user drops the robot at a position from which—since H is empty—the hypothesis generation phase is started. Tracking is done by manually placing the robot relative to its last true position. These user positions are the predicted odometry positions for which the error models compute the corresponding uncertainties (robots drawn in gray with 95%-ellipses in Fig. 5). The real robot (black in Fig. 5) is subject to errors according to the models and reaches the specified locations only approximately. Finally, kidnapping noise can be introduced as illustrated in the experiment.

The simulation run of Fig. 5 shall test simultaneous hypothesis generation and tracking under conditions of artificially exaggerated odometry errors and low feature discriminance. We inject

- *Wheel space noise* accounting for uneven floors, wheel slippage or resolution artifacts. Error growth factors have been magnified by a factor of 2 with respect to the identified values in [1].
- *Cartesian space noise* accounting for collisions. A simple model with error growth proportional to the relative angular and translational displacement has been taken. Growth factors have been magnified by a factor of 10 of what would be physically suggested.
- *Kidnapping noise* accounting for the case of a robot clandestinely brought away from its true position. This type of noise is unmodeled.

The experiments on the real platform (using normal uncertainty parameters) are shown in Figs. 7 and 8. Fig. 7 illustrates hypotheses generation and Fig. 8 shows an experiment on multi-hypothesis tracking where by random movement the true hypothesis is found. The line extraction method used in the experiments is the one from [2].

4.1. Simulation results

In step 1, the robot has no a priori knowledge on its position and observes two perpendicular lines. This yields 72 hypotheses (Fig. 6a). Steps 2, 3 and 4 are sufficient to localize the robot which stays localized until step 8. This although the robot moves blindly on a long distance between steps 6 and 7, causing the uncertainty to grow extensively and thus the error of the true robot as well. In step 11, the robot tries to move forward but collides with a person. It ends up far from the predicted odometry position. No valid pairings can be produced with the current local map at that prediction yielding zero hypotheses—the robot is lost. Hypothesis generation is therefore activated at step 12 with four observed lines. These four lines turn out to be globally unique in combination and therefore yield a single (the true) hypothesis. During steps 13–17 (Fig. 6b) this hypothesis splits up several times since uncertainties do not allow to uniquely determine the true supporting set. Although the lines which give rise to the track splitting are 40 cm apart, the uncertainties from odometry force `track_hypothesis` to generate two or more hypotheses aligned to these lines. In step 18 we kidnap the robot and bring it far down to the bottom of the corridor. The observation at step 18 is still compatible with its expectation from the predicted position (gray). There is no evidence yet to the robot of what happened. Only at position 19 no location dependent constraints can be satisfied anymore—the robot is lost again. The local map from position 20 consists of three lines and yields 12 hypotheses (Fig. 6c) which can be falsified during the last steps up to the true one (Fig. 6d): the robot is localized again.

During this 23 step path, the following data has been recorded: The average relative displacement between the observations of each step is 1.49 m and −18.0° in θ. The average prediction error—difference of predicted (gray) and true (black) location—is 0.26 m and 10.2°. A total of 31 hypotheses performed track splitting into a total of 70 offspring hypotheses. Further, the number of floating point operations has been determined as 58 kflops in average and 355 kflops maximal.

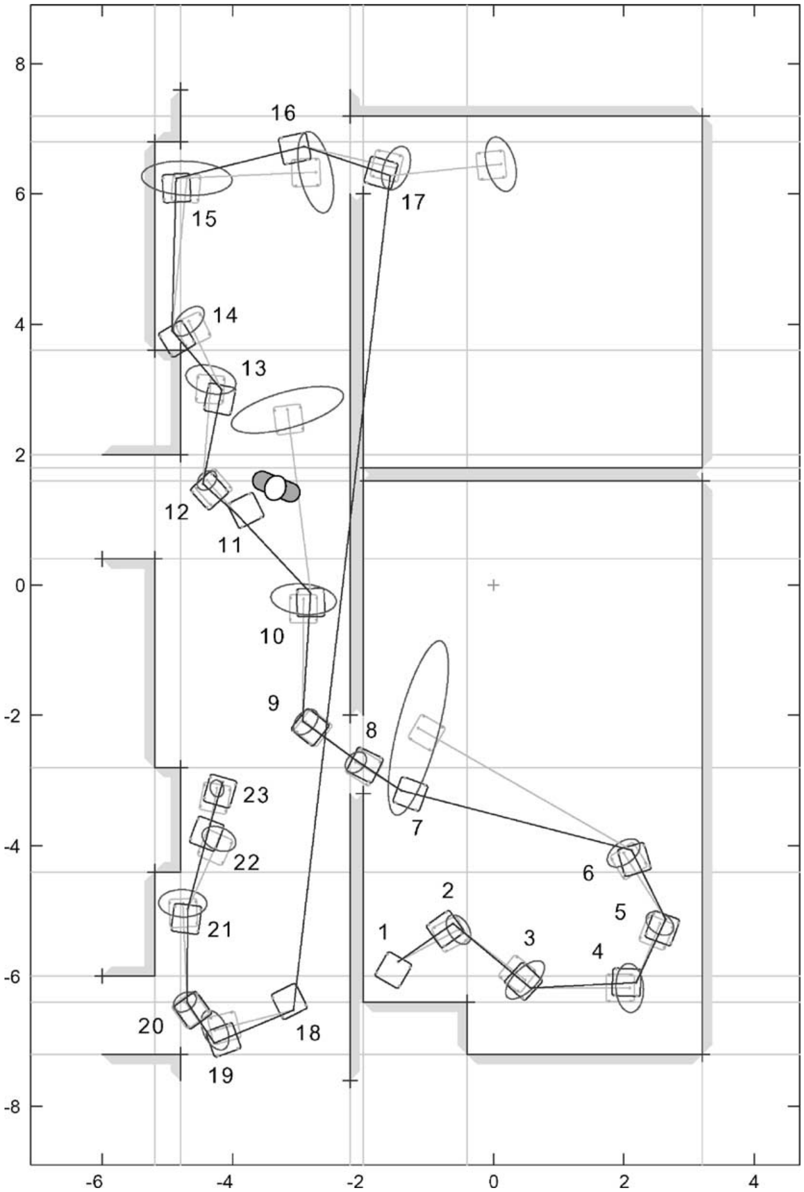

Fig. 5. The simulated test path. Besides extensive odometry uncertainties and errors, the robot collides with a person at step 11 and gets kidnapped at step 18.

The algorithm succeeded always in generating, tracking and confirming the true robot hypothesis. This is remarkable in view of the extent of odometry errors and the average distance between two observations. The robot stays localized in the presence of errors and sensing ambiguities where, drawn from experience, a single hypothesis tracking would fail. This is a dramatic increase in robustness which is made possible with relative small computational costs.

4.2. Results on the real robot

The local map in the experiment of Fig. 7 contains six segments with segment number 202

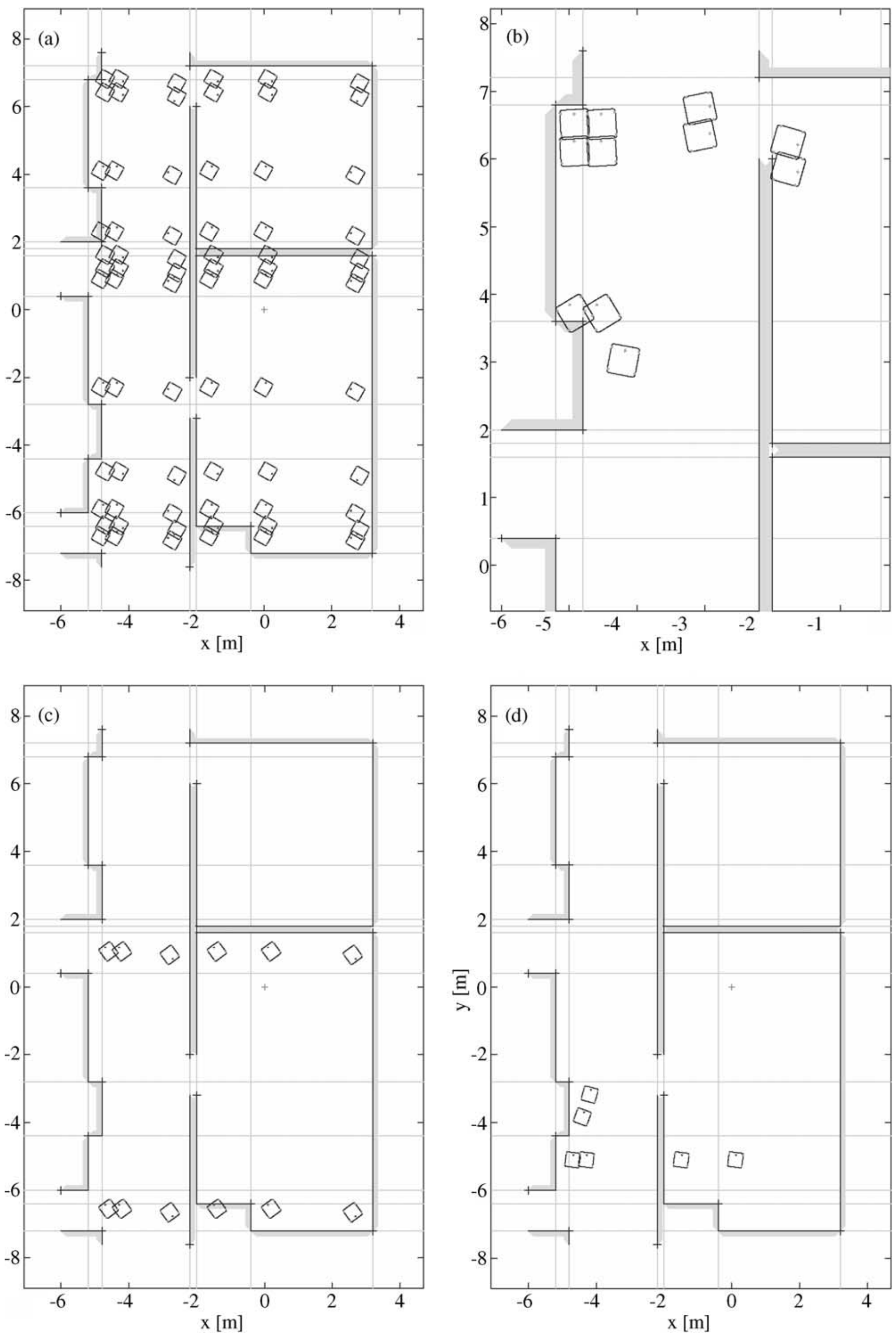

Fig. 6. Hypothesis set *H* at (a) step 1, (b) steps 13–17, (c) step 20 and (d) steps 21 (four hypotheses), 22 and 23. Ellipses in Figs. 5 and 6 denote 95% probability levels.

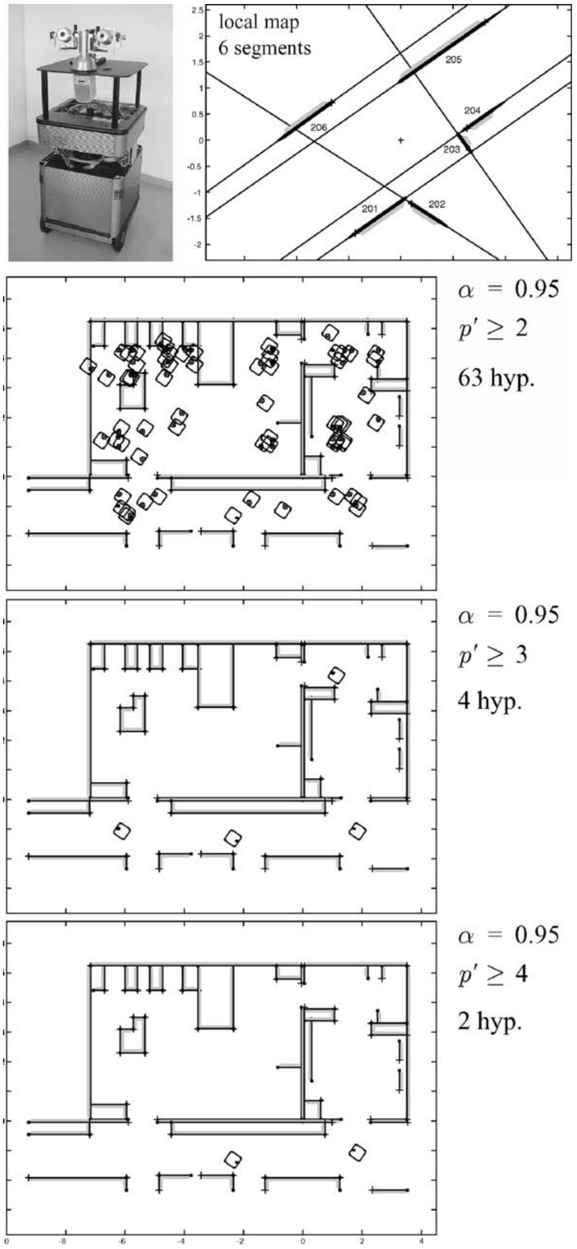

Fig. 7. Generation results for the shown local map on different levels of p'. Execution time is 633 ms. The experiment was carried out on Pygmalion (top left) with a PowerPC604e at 300 MHz CPU.

stemming from a door standing half open. With $p = p' + p^*$ as the number of local observations, let p' be the number of paired observations, and p^* the number of associations to the star branch. With at least two associated local features, we obtain 63 hypotheses, shown in Fig. 7. If we draw only those with $p' \geq 3$, we obtain four hypotheses and two hypotheses for $p' \geq 4$ successful pairings. The only hypothesis with $p' = 5$ is the one at the bottom right of the two remaining ones which is the true hypothesis in the experiment. Since there is an unmodeled feature in the local map (the door), no hypothesis with $p' = 6$, or $p^* = 0$, respectively, is generated. The execution time for hypothesis generation was 633 ms.

Table 1
Statistics of the tracking experiment in Fig. 8

Track no.	Length (m)	Step of elimination
1	0.62	13
2	1.89	47
3	2.52 (end)	82 (end)
4	0.77	20
5	0.77	20

Starting from five location hypotheses in Fig. 8, the robot is able to reject the four incorrect hypotheses after a 1.89 m path. The fact that the robot of tracks 1 and 4 is partially standing within an object is not a rejection criterion since Algorithm 2 only falsifies hypotheses by geometric contradiction and because this kind of information is not contained in the map (feature maps are not free-space maps) (Table 1). The irregular distribution of path points visualizes the instants when the OS scheduler brought the localization task (a non-RT thread) into foreground. The average tracking time is about 10 ms per hypothesis.

5. Related work

An important innovation of this work is the way data association is made. Firstly, features are matched to features—not locations to locations [14]. Secondly, opposed to other approaches using Gaussian hypotheses with features [3,14,19], we do not associate features serially, one at a time, but consider the whole local map L with its binary constraints at once. This has several advantages.

It substantially reduces the number of possible data associations since features must additionally satisfy mutual constraints from their geometric configuration (L) in order to be admitted as valid data association

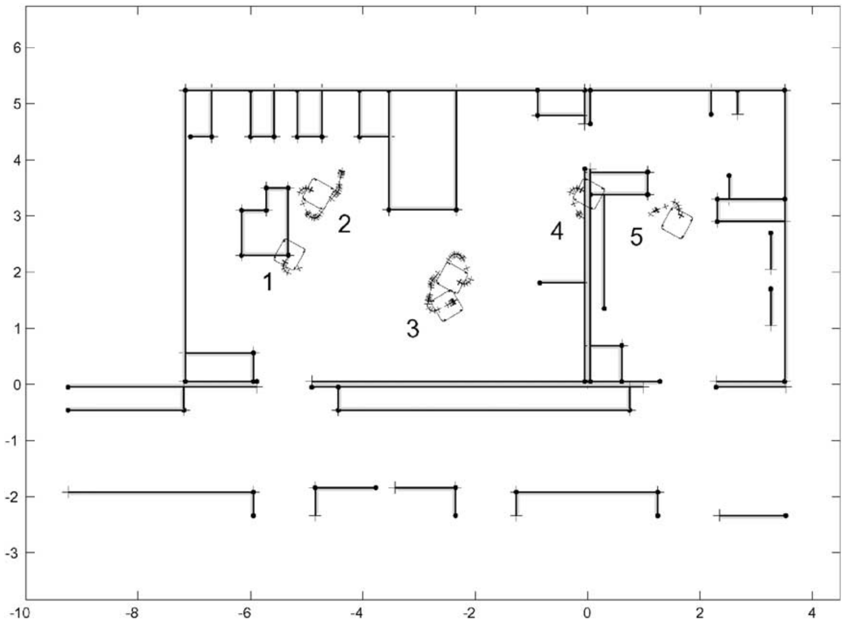

Fig. 8. Multi-hypothesis tracking. Track 3 turns out to be the true one after the last track (2) was rejected at a distance of 1.89 m at step 47.

members. An example illustrates the difference: Given a robot which observes a door and two walls, a global map containing, say, 50 lines and 30 doors and assuming that this particular geometric combination of a door and two walls is globally unique in the map. When considering each feature independently, the observed door will already produce 30 location hypotheses and each wall in L will give rise to 50 further hypotheses, making 130 possible locations.

Our technique will directly generate the single globally unique pose being the only location from which the door and the two walls are seen as in the local map.

A further advantage is in the use of features which yield non-Gaussian location densities such as lines, segments or (x, y)-points. The resulting distributions have the form of a diffuse line, a diffuse segment and a diffuse ring, respectively. This is a common limitation of approaches with Gaussians hypotheses since such distributions are poorly described by the first two moments and require the use of higher-order statistics. With a single feature-to-feature scheme, this limitation is very sensible since each feature must deliver more than two observations. Refs. [3,19] for example, do not consider the above feature types at all. In [14] the distinction between *creative* ($q \geq 3$) and *supportive* features ($q < 3$) is made. For the generation of location hypotheses, they can only use creative features. To overcome this drawback, they form high-level features using several supportive features in order to obtain $q \geq 3$. This, however, requires a specific treatment (in the EIF, for instance) of each combination and number of supportive features.

Our method is significantly less sensitive to that limitation because it requires only that the whole local map L provides more than two observations q (refer to Section 2.2.1 for the definition of q).

6. Conclusions

From the experiments we conclude that the presented approach combines the good properties of EKF localization with globalness. Geometric constraints for data association is a powerful decision mechanism with good convergence properties and allows an optimal management of hypotheses: as many as necessary and as few as possible. The experiments suggest that the approach is practical in a real world embedded system implementation.

References

[1] K.O. Arras, N. Tomatis, B. Jensen, R. Siegwart, Multisensor On-the-fly localization: precision and reliability for applications, Robotics and Autonomous Systems 34 (2–3) (2001).

[2] K.O. Arras, R. Siegwart, Feature extraction and scene interpretation for map-based navigation and map building, in: Proceedings of the SPIE, Mobile Robotics XII, vol. 3210, 1997.

[3] D. Austin, P. Jensfelt, Using multiple gaussian hypotheses to represent probability distributions for mobile robot localization, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, USA, 2000.

[4] Y. Bar-Shalom, X.-R. Li, Estimation and Tracking: Principles, Techniques and Software, Artech House, 1993.

[5] Y. Bar-Shalom, X.-R. Li, Multitarget-Multisensor Tracking: Principles and Techniques, 1995. ISBN 0-9648312-0-1.

[6] J.A. Castellanos, J.D. Tardos, J. Neira, Constraint-based mobile robot localization, in: Proceedings of the 1996 International Workshop on Advanced Robotics and Intelligent Machines, Salford, UK.

[7] J.A. Castellanos, J.D. Tardos, Mobile Robot Localization and Map Building: A Multisensor Fusion Approach, Kluwer Academic Publishers, Dordrecht, 1999.

[8] I.J. Cox, J.J. Leonard, Modeling a dynamic environment using a Bayesian multiple hypothesis approach, Artificial Intelligence 66 (2) (1994) 311–344.

[9] J.L. Crowley, World modeling and position estimation for a mobile robot using ultrasonic ranging, in: Proceedings of the IEEE International Conference on Robotics and Automation, Scottsdale, USA, 1989.

[10] F. Dellaert, D. Fox, W. Burgard, S. Thrun, Monte Carlo localization for mobile robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, USA, 1999.

[11] M. Drumheller, Mobile robot localization using sonar, IEEE Transactions on PAMI 9 (2) (1987) 325–332.

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

[13] W.E.L. Grimson, T. Lozano-Pérez, Localizing overlapping parts by searching the interpretation tree, IEEE Transactions on PAMI 9 (4) (1987) 469–482.

[14] P. Jensfelt, S. Kristensen, Active global localization for a mobile robot using multiple hypothesis tracking, IEEE Transactions on Robotics and Automation 17 (5) 2001.

[15] P. Jensfelt, D. Austin, O. Wijk, M. Andersson, Experiments on augmenting condensation for mobile robot localization, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, USA, 2000.

[16] J.J. Leonard, H.F. Durrant-Whyte, Directed Sonar Sensing for Mobile Robot Navigation, Kluwer Academic Publishers, Dordrecht, 1992.

[17] J.H. Lim, J.J. Leonard, Mobile robot relocation from echolocation constraints, IEEE Transactions on PAMI 22 (9) (2000) 1035–1041.

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

[19] S.I. Roumeliotis, G.A. Bekey, Bayesian estimation and Kalman filtering: a unified framework for mobile robot localization, in: Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, USA, 2000.

[20] R. Simmons, S. Koenig, Probabilistic navigation in partially observable environments, in: Proceedings of the International Joint Conference on Artificial Intelligence, vol. 2, 1995, pp. 1660–1667.

[21] N. Tomatis, I. Nourbakhsh, K.O. Arras, R. Siegwart, A hybrid approach for robust and precise mobile robot navigation with compact environment modeling, in: Proceedings of the IEEE International Conference on Robotics and Automation, Seoul, South Korea, 2001.

Kai O. Arras is a Ph.D. student with the Autonomous Systems Lab at the Swiss Federal Institute of Technology Lausanne (EPFL). He received his Master’s in electrical engineering from the Swiss Federal Institute of Technology Zurich (ETHZ) in 1995 and worked as a research assistant in Nanorobotics at the Institute of Robotics in Zurich. In 1996 he joined Prof. Siegwart to help setting up the Autonomous Systems Lab at EPFL where he is working on several aspects of mobile robotics. His fields of interest include system integration, feature extraction, local and global localization, SLAM, robot art and robots for exhibitions.

José A. Castellanos was born in Zaragoza, Spain, in 1969. He received the M.S. and Ph.D. degrees in industrial-electrical engineering from the University of Zaragoza, Spain, in 1994 and 1998, respectively. He is Associate Professor in the Departamento de Informática e Ingeniería de Sistemas, University of Zaragoza, where he is in charge of courses in automatic control systems, mobile robotics and computer simulation of dynamical systems. His current research interests include perception for robotics applications, multisensor fusion of uncertain information and mobile robot localization and mapping (SLAM/CML).

Martin Schilt is a research assistant with the Autonomous Systems Lab at the Swiss Federal Institute of Technology Lausanne (EPFL). He received his Master’s in computer science from the Swiss Federal Institute of Technology Zurich (ETHZ) in 2002. His research covered mobile robot localization, active localization and multi-robot coordination.

Roland Siegwart (1959) received his M.Sc. ME in 1983 and his Doctoral degree in 1989 at the Swiss Federal Institute of Technology (ETH) Zurich. After his Ph.D. studies he spent one year as a postdoc at Stanford University where he was involved in micro-robots and tactile gripping. From 1991 to 1996 he worked part time as R&D director at MECOS Traxler AG and as a lecturer and deputy head at the Institute of Robotics, ETH. During this time he was mainly involved in magnetic bearings, mechatronics and micro-robotics. Since 1997 he is a full professor for Autonomous Systems and Robots at the Swiss Federal Institute of Technology Lausanne (EPFL), leading a research group of around 20 people. His current research interests are robotics and mechatronics, namely, high precision navigation, space robotics, human–robot interaction, all terrain locomotion and micro-robotics. He lectures various courses in robotics, mechatronics and smart product design at the two Swiss Federal Institutes of Technology and is cofounder of several spin-off companies. He published more than 80 papers and is member of various scientific committees. He namely represents Switzerland in the International Federation of Robotics (IFR) and the Advisory Group for Automation and Robotics (AGAR) of the European Space Agency (ESA). He was program and general chair for various international conferences, including the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems.