Towards Feature-Based Multi-Hypothesis Localization and Tracking

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

[a]Autonomous Systems Lab
Institut de Systèmes Robotiques
Swiss Federal Institute of Technology Lausanne (EPFL)
CH–1015 Lausanne, Switzerland
kai-oliver.arras@epfl.ch

[b] Robotics and Real-Time Group
Centro Politecnico Superior
Universidad de Zaragoza
María de Luna 1
E–50015 Zaragoza, Spain

Abstract

In this paper we present a probabilistic feature-based approach to multi-hypothesis global localization and tracking. Hypotheses are generated using a constraint-based search in the interpretation tree of possible local-to-global-pairings. This results in a set of continuously located position hypotheses of unbounded accuracy. For tracking, the same constraint-based technique is used. It performs track splitting as soon as location ambiguities arise from uncertainties and sensing. This yields a localization technique of extraordinary robustness which can deal with significant errors from odometry, collisions and kidnapping. Simulation experiments successfully demonstrate these properties at very low computational cost. The presented approach is theoretically sound which makes that the only parameter is the significance level α on which all statistical decisions are taken.

1. Introduction

Kalman filter-based position tracking with geometric features has been proven to be a very powerful localization technique and provides several desirable properties. It usually operates with minimalistic environment representations and nevertheless exhibits great robustness with respect to environment dynamics. It further 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 approach to localization [15][14][10] 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 earlier methods have been proposed [8][12]. They employ the principle of particle filters where location hypotheses are maintained as a set of samples. However, all these techniques maintain constantly a big number of hypotheses which in the case of particle filters have to be carefully weighted and updated. The ability of these techniques to properly react when location ambiguities arise from environment or sensing is due to the quantity of samples and a diffusion strategy which must be appropriately chosen.

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 always exactly as many hypotheses as necessary and as few as possible. The technique which provides this desirable property is a constrained-based search in an interpretation tree [11][9][4][13]. This tree is spanned by all possible local-to-global associations, given a local map of observed features and a global map of model features. We consistently employ the same search for hypothesis generation and tracking.

Figures and examples will use point-, angle- and line features for illustration. Please note that the problems discussed in this paper inherently appear in the use of features for robot navigation. Depending of the feature type, they are more or less visible.

1.1 Motivation and Problem Statement

After five years of experience in EKF-based position tracking on more than 100 km overall travel distance with three different robots[1] [1], we locate the most critical failure causes for a localization technique as follows:

1. This is a very conservative estimate. Explicitly logged are 84 km during a small fraction of time where these robots are operational with this localization method.

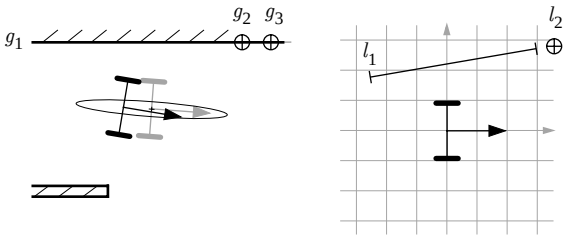

Figure 1. *A situation where the robot goes lost and where this is very difficult to detect: when the vehicle arrives at the end of a corridor with a critical amount of accumulated odometry drift (the estimated position is drawn in gray, the true one in black), the local corner feature* $\{l_2\}$ *is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

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

In practice, single hypothesis tracking can often relocalize a robot which went lost due to non-discriminant features, since they typically yield close-to-the-truth pose estimates. But in the particular case of figure 1 and in general however, both problem sources can lead to false matchings and irrecoverable lost situations. A robust tracking technique shall therefore cope with these problems, especially with their simultaneous occurrence. It shall further be able to re-localize from unavoidable lost situations caused, for instance, by collisions and kidnapping.

2. Hypothesis Generation

2.1 Notation

In this paper we will make regular use of several terms and symbols.

L	the local map $\{l_1, l_2, \ldots, l_l\}$ of l observed features
G	the global map $\{g_1, g_2, \ldots, g_m\}$ of m model features
h	a robot location hypothesis $h = \{L_h, S_h\}$
H	the set of all n hypotheses $H = \{h_1, h_2, \ldots, h_n\}$
L_h	the robot location of h with moments x and P
S_h	the supporting set of h. It contains p pairings $\{l_i, g_j\}$ by which h is held; $S_h = \{\{l_1, g_{j_1}\}, \ldots, \{l_p, g_{j_p}\}\}$
F_h	the set of supporting global features $F_h = \{g_1, g_2, \ldots, g_p\}$ of S_h
$h_{i,j}$	the j-th offspring hypothesis of hypothesis $h_i \in H$

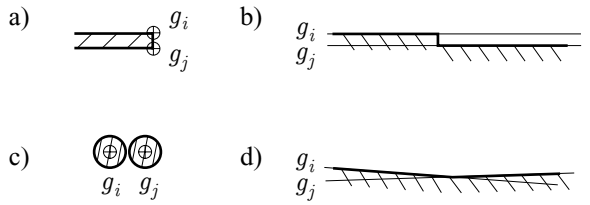

Figure 2. *Examples of feature types which are typically subject to low feature discriminance: a) angle features modeling corners, c) point features modeling columns and b) and d) line features modeling walls. Less critical are features of higher parameter dimensionality as segments or circles or features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

2.2 Geometric Constraints

Given a pairing $p_{ij} = \{l_i, g_j\}$ of a local and a global feature, a geometric constraint is a condition on a property between two features of a pairing or between two pairings. Geometric constraints direct the search in the space of all possible data associations and reduce enormously the complexity of the problem. We can classify geometric constraints into two categories:

2.2.1 Location Independent Constraints

Location independent constraints can be validated without having an estimation of the robot location. They include *unary* and *binary* constraints.

Unary constraints are intrinsic properties of a feature. Examples are feature type, color, texture or geometric dimensions as length or width. In the attempt to find valid pairings, unary compatibility of two features is directly obtained by comparison (function satisfy_unary_constraints). Unary constraints allow simple preprocessing of the map. They are very powerful since whole subspaces can be excluded from the search beforehand.

Binary constraints always apply to two features. Examples include relative measures between features like distance or angle. Binary constraints are used to validate whether two observed features are consistent with two model features (function satisfy_binary_constraints). With uncertain geometric information all comparisons use the Mahalanobis distance and a significance level α.

2.2.2 Location Dependent Constraints

Location dependent constraints come into play as soon as a position estimate is available. The fundamental constraint of this type is *rigidity*, further there are *visibility* and *extension*.

The *rigidity constraint* performs a single-feature global-to-local frame transform also known from the matching

step in a EKF localization scheme. Given a robot location L_h with moments x and P, an observation l_i in the robot frame, and a pairing candidate g_j from the map, rigidity is satisfied if the observed feature matches the model feature in the robot frame (rigidity works in any reference system however).

Let R_i and P_j be the covariance estimates associated to l_i and g_j, ∇h_x the Jacobian of the linearized feature measurement model (the frame transform) with respect to the robot position and ∇h_j the Jacobian with respect to the uncertain feature location in the map. Then the model feature in the robot frame is obtained by the nonlinear transform $\hat{g}_j = h(x, g_j)$, the innovation by $v_{ij} = l_i - \hat{g}_j$ with covariance $S_{ij} = \nabla h_x P \nabla h_x^T + \nabla h_j P_j \nabla h_j^T + R_i$ using first-order error propagation (and assuming vanishing robot-to-feature correlations). Finally, rigidity is satisfied if

$$v_{ij} \cdot S_{ij}^{-1} \cdot v_{ij}^T \leq \chi^2_{\alpha,\, r} \qquad (1)$$

holds on the significance level α where r is the number of feature parameters.

Visibility constraints indicate whether a model feature g_j is visible from a robot location L_h. Non-visibility can be due to feature properties as relative view direction, and due to sensing limitation as maximal range or resolution. Segments and lines, for instance, always have a visible outside toward free space and an invisible inside toward the wall they model. This introduces the concept of the robot being behind a feature which therefore can be prevented from further consideration.

Extension constraints test whether an observed feature is contained in the candidate model feature. This is relevant for features like line segments or circular arcs whose observations can be smaller than the model features in some sense. Rigidity might not be able to test on this. In [4] extension is satisfied if the observed segment is fully contained in the model segment (they overlap).

2.3 Global Localization Using Geometric Constraints

The problem of mobile robot localization is formulated as a matching problem using geometric constraints [9][4] [13]. It is the problem of finding the set of correct associations of observations to model features in the space of all possible ones. 'Correct' denotes statistical compatibility given all involved uncertainties. The search space has the structure of an interpretation tree [11] with l levels and $m+1$ branches. The extra branch allows correct associations in the presence of spurious observations and thus accounts for environment dynamics.

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

  H ← {}
  if L = {} then
    H ← H ∪ {h}
  else
    l ← select_observation(L)
    for g ∈ G do
      p ← {l, g}
      if satisfy_unary_constraints(p) then
        if location_available(h) then
A       {  accept ← satisfy_location_dependent_cnstr(L_h, p)
           if accept then
             h' ← h
             S_h' ← S_h ∪ {p}
             L_h' ← estimate_robot_location(S_h')
           end
        else
B       {  accept ← true
           for p_p ∈ S_h while accept
             accept ← satisfy_binary_constraints(p_p, p)
           end
           if accept then
             h' ← h
             S_h' ← S_h ∪ {p}
             L_h' ← estimate_robot_location(S_h')
             if location_available(h') then
               for p_p ∈ S_h while accept
                 accept ← satisfy_location_dependent_cnstr(L_h', p)
               end
             end
           end
        end
        if accept then
          generate_hypotheses(h', L\{l}, G)
        end
      end
    end
    generate_hypotheses(h, L\{l}, G)
  end

return H
end
```

Algorithm 1. *Given a hypothesis h (with empty S_h and no location in the beginning), the local map L and the global map G, the function returns the set of generated location hypotheses H.*

The search strategy employed here is a depth-first, backtracking search which applies geometric constraints at each tree node to validate whether geometric relations among observations and their associated model features are (still) satisfied. This is realized in a identifying while locating scheme in which pairing formation and location estimation is performed simultaneously (algorithm 1, [5]). The strategy reflects the fact that location dependent constraints are more powerful in falsifying infeasible hypotheses than location independent constraints.

Algorithm 1 tries first to find a minimal supporting set with location independent constraints such that a location estimate can be determined (part B). When an observation is selected from the local map (function select_observation), optional rules can be applied to choose an observation which generates 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, the function recurs in a depth-first manner with a refined location estimate (function estimate_robot_location, see section 2.3.1) and an extended supporting set (part A). Thus, at each tree level, all consistent pairings between the observation l and all model features $g \in G$ are generated. The algorithm also considers the possibility of observations being spurious by recursing without consideration of the previously selected observation. Note that the significance level α is the only parameter the user has to specify. It decides on acceptance or rejection of the geometric constraints.

2.3.1 Estimating the Robot Location

Given a supporting set $S_h = \{\{l_1, g_{j_1}\}, \{l_2, g_{j_2}\}, \ldots, \{l_p, g_{j_p}\}\}$ the robot position L_h can be estimated using the extended Kalman filter. 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 [3] for derivation and further 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 parameters r. Then the EIF is as follows:

$$P^{-1}(k+1|k+1) = P^{-1}(k+1|k) + \nabla h^T R(k+1)^{-1} \nabla h \quad (2)$$

$$\hat{x}(k+1|k+1) = P(k+1|k+1) \cdot [P^{-1}(k+1|k) \cdot \hat{x}(k|k+1) + \nabla h^T R(k+1)^{-1} \nabla h \cdot \xi(k+1)] \quad (3)$$

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

$$\nabla h \cdot \xi(k+1) = v(k+1). \quad (4)$$

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

$$P^{-1}(k+1|k) = \mathbf{0}_{3\times 3}. \quad (5)$$

By substituting equation (5) into equations (2) and (3) and using (4), 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.

$$\nabla h \cdot \hat{x}(k+1|k+1) = v(k+1) \quad (6)$$

The solution of (6) is obtained via the pseudoinverse

$$\nabla h' = (\nabla h^T \nabla h)^{-1} \nabla h^T. \quad (7)$$

where we can distinguish between $\nabla h^T \nabla h$ being singular or non-singular. In the latter case, the equation system (6) 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

Single hypothesis tracking assumes always correctness of its estimate and does not account for data association ambiguities even when they were noticeable. Under conditions of uncertain pose estimates and low feature discriminance, this approach, as mentioned in section 1, is likely to produce incorrect pairings.

Even if a technique for global localization as presented in section 2 were available, a single hypothesis approach to pose tracking would require a method to *detect lost situations*. But detecting lost situations is extremely difficult in general. A lost situation is equal to the case of estimator inconsistency which loosely speaking means that the state moments make mutually incompatible statements. Fig. 1, an example from practice, illustrates a case where an incorrect pairing is performed causing the robot to be wrongly aligned to model corner g_3 instead of g_2, and though its uncertainty to collapse. In such a situation it is useless to try to detect the lost situations by

- looking whether the pose uncertainty grows
- looking for a significantly deviating behavior of the matching statistics
- looking for a significantly deviating behavior of the observation residuals (innovations)
- comparing the state prediction to a locally generated hypothesis set

The tools from estimation theory for estimator inconsistency detection (e.g. testing on innovation whiteness [3]) offer no means to discover the lost situation of figure 1.

The strategic step towards a solution of this problem is to loosen the strict distinction of being localized and being lost. With $H = \{h_1, h_2, \ldots, h_n\}$ as the set of location hypotheses and $n = |H|$ the number of hypotheses, we introduce the following terms

$$n \begin{cases} = 0 & \text{the robot is } \textit{lost} \\ = 1 & \text{the robot is } \textit{localized} \\ > 1 & \text{the robot is } \textit{not localized} \end{cases} \quad (8)$$

3.1 Hypothesis Generation During Tracking

It lies in the nature of the localization task that uncertainties from perception and action are accompanied by sensing ambiguities. Finding correct local-to-global data associations under these conditions can be very difficult. Maintaining multiple robot location hypotheses is a scheme with the ability to cope with this type of ambiguity since different statistically compatible supporting sets can be represented by different location hypotheses (figure 3).

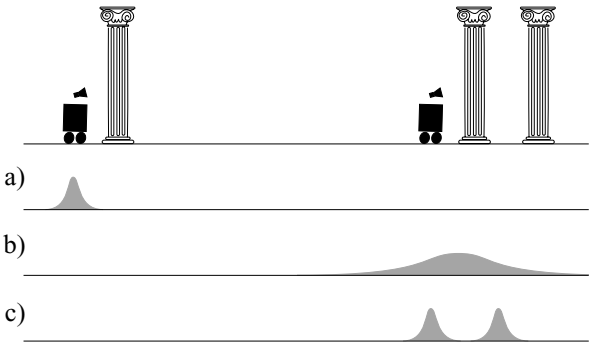

Figure 3. *The fundamental idea behind multi-hypothesis position tracking: A well localized robot in a) moves and observes a* single *feature in b) where it is impossible to say which is the correct pairing in view of the uncertainties. Instead, two hypotheses are generated in c) which represent all possible pairings at that location. The two hypotheses are tracked using location dependent constraints until a single one remains.*

Therefore we look for an algorithm which re-generates hypotheses during tracking as soon as there is no guarantee anymore that the correct matching can be found. This property has validate_hypothesis, which, given a location, a local and a global map, splits up into multiple offspring hypotheses if statistical compatibility with several supporting sets can be established at that location. As for hypothesis generation, algorithm 2 generates at each level of the interpretation tree all consistent pairings between the observation l and all model features $g \in G$. If a new acceptable pairing is found, the function recurs with an extended supporting set but this time *not* with a refined position estimation. In this manner the algorithm finds all supporting sets in the vicinity of the initially given location L_h and returns them in form of a hypothesis set H_v. Again, the second recursion call implements the extra

```
function validate_hypothesis(h, L, G)

  H_v ← {}
  if L = {} then
    H_v ← H_v ∪ {h}
  else
    l ← select_observation(L)
    for g ∈ G do
      p ← {l, g}
      if satisfy_unary_constraints(p) then
        if satisfy_location_dependent_cnstr(L_h, p) then
          S_{h_p} ← S_h ∪ {p}
          H_v ← H_v ∪ validate_hypothesis(h_p, L \ {l}, G)
        end
      end
    end
    H_v ← H_v ∪ validate_hypothesis(h, L \ {l}, G)
  end

return H_v
end
```

Algorithm 2. *Given the hypothesis to be validated at location L_h, h, the local map L and the global map G, the function returns the set of valid hypotheses H_v.*

branch in the interpretation tree that allows correct associations in the presence of outlier observations. After validate_hypothesis has been applied for each h_i, we can distinguish the three cases verification, falsification and division:

- $|H_v| = 1$, *hypothesis verification*. The hypothesis h_i is confirmed by location dependent constraints. Given the supporting set S_{h_i}, the robot location is estimated and h_i is admitted to H.
- $|H_v| = 0$, *hypothesis falsification*. The hypothesis can not be held any more by location dependent constraints on the significance level α. It gets rejected. Hypothesis scoring could be employed here if the quality of the noise models were so poor that the true hypothesis gets discarded often. Hypotheses would be rejected if the score fell below a threshold through several falsifications and not just by a single one.
- $|H_v| > 1$, *hypothesis division*. The track of hypothesis h_i splits up into several offspring hypotheses $\{h_{i,1}, h_{i,2}, \ldots, h_{i,o}\}$ which all can be held by location dependent constraints at the predicted robot location (figure 4). The robot locations are estimated with the EIF using their respective supporting set.

3.2 Hypothesis Elimination During Tracking

When an uncertain hypothesis splits up, it can happen that *duplicate hypotheses* are produced. This is illustrated in figure 5, where two hypotheses h_1, h_2 produce each four hypotheses. If these duplicates are not eliminated, H will grow containing redundant information, and thus undermining our intent to reach $|H| = 1$.

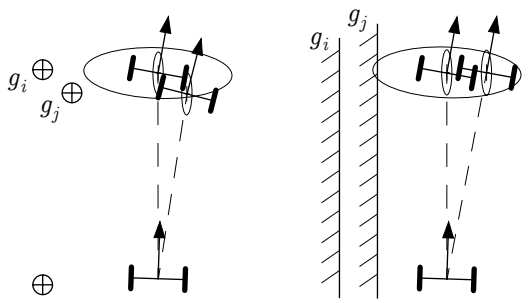

Figure 4. *Examples for hypothesis generation during tracking. Given a hypothesis, a displacement resulting in an uncertain pose and a local map with a single feature* $L = \{l_1\}$*, two hypotheses are generated. One with the supporting sets* $S_h = \{\{l_1, g_i\}\}$ *and the other with* $S_h = \{\{l_1, g_j\}\}$*. Both can be held on the specified level* α*.*

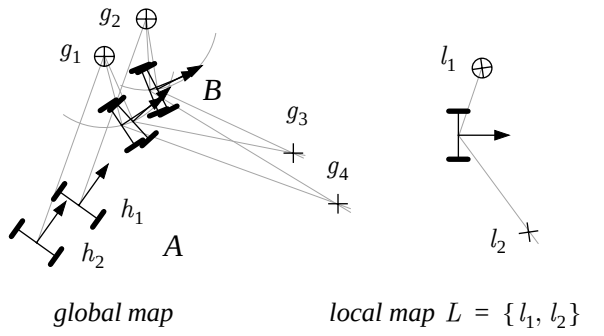

Figure 5. *An example of hypothesis duplication. Given a local map with a* x,y*–point feature* l_1*, and a* ϕ*–angle feature* l_2*, hypotheses* h_1*,* h_2 *split up each into four offsprings after an uncertain movement A to B. This results in eight hypotheses at* B*, four of them being redundant.*

3.2.1 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. Testing on positions L_h and eliminate hypotheses if they were close in some sense seems to be a straightforward approach. If however feature discriminance is low and closely located model features produce closely located position hypotheses, it would be incorrect to declare them as duplicates. They are held by different supporting sets and therefore they are different robot locations – independent on their distance.

Set-based conditions account for this distinction. They keep correctness in this sense and thereby conserve full localization accuracy. Following this direction, identical location is due to identical supporting sets.

$$h_i \equiv h_j \Leftrightarrow (S_{h_i} = S_{h_j}) \tag{9}$$

This condition can be further generalized. In case of extensive uncertainties, unexpected 'permutations' of a supporting set become statistically correct. Such associations can also yield identical locations. If we loosen condition (9) and demand only equal global supporting features F_h, we cover all these associations. It can be shown that given an F_h, it is sufficient to keep only the supporting set – among all possible supporting sets of F_h – with minimal error distance [2]. With the extended condition (10), these 'permutations' are declared as duplicates and the rejection criterion will then choose the hypothesis with the best S_h among them.

$$h_i \equiv h_j \Leftrightarrow (F_{h_i} = F_{h_j}) \tag{10}$$

Sometimes the current observation does not provide enough information to uniquely estimate a robot position (e.g. robot observes a single line or two angle-only features). 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 (10) is unable to distinguish duplicate hypotheses because several distinct hypotheses can be aligned to the same model feature (figure 6). They all would be wrongly declared as duplicates.

For non-unique pose estimates we therefore use an additional distance condition. 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

$$d_{h_i h_j} = (x_{h_i} - x_{h_j}) \cdot (P_{h_i} + P_{h_j})^{-1} \cdot (x_{h_i} - x_{h_j})^T \tag{11}$$

where $d_{h_i h_j} < \chi^2_{3,\alpha}$ must hold. $\chi^2_{3,\alpha}$ is a value chosen from a χ^2-distribution with three degrees of-freedom. The complete condition is now

$$h_i \equiv h_j \Leftrightarrow \begin{cases} (F_{h_i} = F_{h_j}) & \ell_{h_i} \\ (F_{h_i} = F_{h_j}) \wedge (d_{h_i h_j} < \chi^2_\alpha) & \neg\, \ell_{h_i} \end{cases} \tag{12}$$

where we wrote $\chi^2_{3,\alpha} = \chi^2_\alpha$. The flag ℓ_{h_i} is true if the supporting set yields a unique position and false otherwise (ℓ_{h_i} and ℓ_{h_j} are identical as $F_{h_i} = F_{h_j}$ is demanded).

3.2.2 Duplicate Rejection

Unlike Markovian and particle filter approaches, location 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.

Location estimates differ, however, in their joint Mahalanobis distance. The joint Mahalanobis distance has the same structure than equation 1 except that it applies not only to a single pairing but sums up over the whole supporting set, including correlations. It is basically the sum

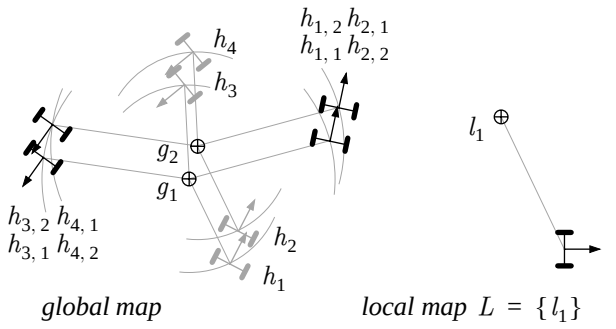

Figure 6. *With a single point feature in the local map, the EIF returns a non-unique pose in form of a circle around the respective model features g_1 and g_2. For example $h_{1,1}$, $h_{2,2}$ and $h_{3,1}$, $h_{4,2}$ are (distinct) pairs of duplicates, aligned to the same model feature g_1. Since they all have $F_h = \{g_1\}$, they must be distinguished by their distance.*

of the weighted squared error distances (or residuals). The best hypothesis is the one minimizing the distance out of a duplicate set H_d. In other words, we accept the hypothesis which, from its location, satisfies best the rigidity constraint.

4. Experiments

A simulation environment has been developed which allows to create maps and to manually guide the robot through a virtual environment. Local maps are generated by ray tracing where a 360° range finder with 1° resolution is simulated. The maximal range has been limited to two meters. Odometry employes 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 (which is known in simulation). 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 figure 7). The real robot (black in figure 7) 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 hereafter.

The current simulation employs infinite lines as features. It shall however be underlined that the presented approach to multi-hypothesis localization is completely general with respect to the feature type. We briefly summarize the relevant properties of lines for hypothesis generation: infinite lines have no unary and no extension constraints. Their only binary constraints is the angle between two lines. Rigidity and visibility are well defined. Selecting observations from local maps (function select_observation) is best done by the rule to return perpendicular lines to a given reference line. The advantage of infinite lines lies in the capacity to efficiently model man-made environments with long walls. In view of the complexity of the search problem, compact environment modeling is vital and has thus a compensating effect onto the lack of unary and extension constraints.

The simulation run of figure 7 shall test simultaneous hypothesis generation and tracking under conditions of significant odometry errors and low feature discriminance. We inject

- *Wheel space noise* accounting for uneven floors, wheel slippage or resolution artifacts. Here, the consistent model from [7] has been implemented using a piecewise linear approximation. Error growth factors have been magnified by a factor of two.
- *Cartesian space noise* accounting for collisions. A simple model with error growth proportional to the relative angular and translational displacement has been taken [6]. Error growth factors have been magnified by a factor of ten.
- *Kidnapping noise* accounting for the case of a robot clandestinely brought away from its true position. This type of noise is unmodeled.

4.1 Results

In step 1, the robot has no a priori knowledge on its position and observes two perpendicular lines. This yields 72 hypotheses (figure 8a). Steps 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 to 17 (figure 8b) 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 validate_hypothesis to generate two or more hypotheses aligned to these lines. In step 18 we kid-

nap 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 thus no evidence 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 twelve hypotheses (figure 8c) which can be falsified during the last steps up to the true one (figure 8d): 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 (at step 2 when 72 hypotheses are tracked, partially split up and eliminated).

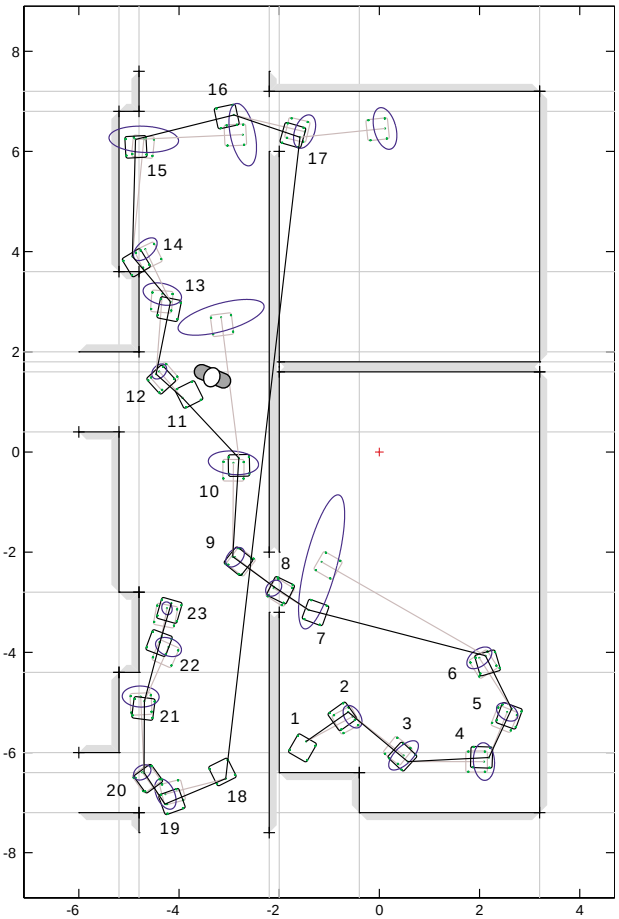

Figure 7. *The simulated test path. Besides extensive odometry uncertainties and errors, the robot collides with a person at step 11 and gets kidnapped at step 18. The ellipses denote unmagnified 95% probability levels.*

4.2 Discussion

The algorithm succeeded always in generating and tracking 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 a small amount of computational cost. Furthermore, after steps 1, 12 and 20 where the hypotheses are generated without an a priori position estimate, we can state a very fast convergence toward the true hypothesis (figure 9). This although infinite lines provide only minimalistic environment information.

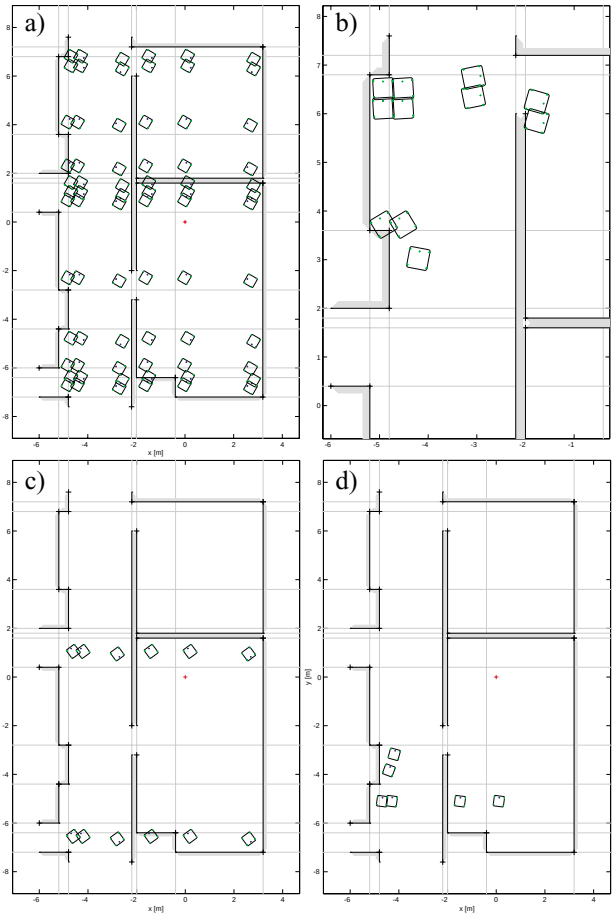

Figure 8. *Hypotheses set H at a) step 1, b) steps 13-17, c) step 20 and d) steps 21 (four hypotheses), 22 and 23.*

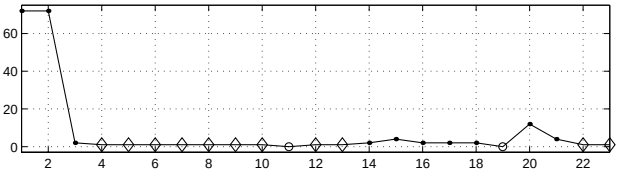

Figure 9. *Number of hypotheses. Diamonds (steps 4-10,12,13,22,23): the robot is* localized, *circles (steps 11,19): the robot is* lost, *points: the robot is* not localized.

An important observation is also that odometry error models become less important. They are liberated from the burden to be physically well grounded uncertainty models but get the character of local search regions in which validate_hypothesis looks for feasible pairings.

5. Conclusions

In this paper we presented a probabilistic feature-based approach to multi-hypothesis global localization and tracking. We addressed further the issue of stable tracking which consistently uses the same search technique as for hypothesis generation. In order to cope with data association ambiguities, hypotheses split up as soon as the correct pairing cannot be guaranteed anymore. From the experiments we conclude that the presented approach is practical and exhibits the degree of robustness which was initially required. With 58 kflops as the mean computational effort for both, hypothesis generation and tracking, the experiments further suggest that the typical efficiency of the feature-based paradigm could have been retained.

Future work will focus on the explicit mathematical representation and treatment of cases of non-unique EIF location estimates. Finally, simulations usually allow to identify only optimistic bounds of the practicability of a method. This is why an implementation on a real robot is needed.

References

[1] Arras K.O., Tomatis N., Jensen B., Siegwart R., “Multisensor On-the-Fly Localization: Precision and Reliability for Applications”, Robotics and Autonomous Systems, 34 (2-3), p.131-143, 2001.

[2] Arras K.O., “Detection and Rejection of Duplicate Hypotheses: A Step-By-Step Derivation”, Technical Report Nr. EPFL-ASL-TR-01-01R1, August 2001.

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

[4] Castellanos J.A., Tardos J.D., Neira J., “Constraint-Based Mobile Robot Localization”, 1996 Int. Workshop on Advanced Robotics and Intelligent Machines, Salford, UK.

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

[6] Chenavier F., Crowley J.L., “Position Estimation for a Mobile Robot Using Vision and Odometry”, 1992 IEEE Int. Conf. on Robotics and Automation, Nice, France.

[7] Chong K.S., Kleeman L., “Accurate Odometry and Error Modelling for a Mobile Robot”, 1997 IEEE Int. Conf. on Robotics and Automation, NM, USA.

[8] Dellaert F., Fox D., Burgard W., Thrun S., “Monte Carlo Localization for Mobile Robots”, 1999 IEEE Int. Conf. on Robotics and Automation, Detroit, USA.

[9] Drumheller M., "Mobile Robot Localization Using Sonar", IEEE Trans. Pattern Analysis and Machine Intelligence, 9(2), p.325-332, Mar. 1987.

[10] Fox D., Burgard W., Thrun S., “Markov Localization for Mobile Robots in Dynamic Environments”, Journal of Artificial Intelligence Research, vol.11, p.391-427, 1999.

[11] Grimson W.E.L., Lozano-Pérez, “Localizing Overlapping Parts by Searching the Interpretation Tree”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(4), p.469-482, 1987.

[12] Jensfelt P., Austin D., Wijk O., Andersson M., “Feature Based Condensation for Mobile Robot Localization”, 2000 Int. Conf. on Robotics and Automation, San Francisco, USA.

[13] Lim J. H. and J. Leonard J.J., "Mobile Robot Relocation from Echolocation Constraints", IEEE Trans. Pattern Analysis and Machine Intelligence, 22(9), p.1035-1041, 2000.

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

[15] Simmons R., Koenig S., “Probabilistic navigation in partially observable environments“, in Proc. of the Int. Joint Conf. on Artificial Intelligence, vol.2, p. 1660-7, 1995.