Proceedings of the 2002 IEEE
International Conference on Robotics & Automation
Washington, DC • May 2002

Feature-Based Multi-Hypothesis Localization and Tracking for Mobile Robots Using Geometric Constraints

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

[a]Autonomous Systems Lab
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
E–50015 Zaragoza, Spain

Abstract

In this paper we present a new probabilistic feature-based approach to multi-hypothesis global localization and pose 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 robot location 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 very robust localization technique which can deal with significant errors from odometry, collisions and kidnapping. Simulation experiments and first tests with a real robot 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 with several desirable properties: It operates with minimalistic environment representations, 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 approach to localization [14][13][9] 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 [7][11]. 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 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.

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 [10][8][4][12]. 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 consistently employ the same search for hypothesis generation and pose tracking.

Earlier work [6] deals with multiple hypotheses for map building. Using segments and corners from ultrasonic sensors, their hypotheses model a *typological* feature ambiguity since the features were 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.

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.

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 fail-

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.

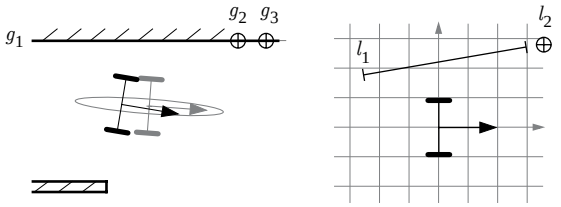

Figure 1. *A situation where the robot goes lost and where this is very difficult to detect: when the vehicle arrives at the end of a corridor with a critical amount of accumulated odometry drift (the estimated position is drawn in gray, the true one in black), the local corner feature* $\{l_2\}$ *is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

ure causes for a localization technique as follows:

- *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 spatial 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 general both problem sources, especially in simultaneous occurrence, can lead to false matchings and irrecoverable lost situations. A robust localization technique shall therefore cope with these issues and further be able to recover from lost situations caused by collisions and kidnapping.

2. Hypothesis Generation

2.1 Geometric Constraints

A pairing $p_{ij} = \{l_i, g_j\}$ is an interpretation of the measurement l_i saying that l_i and g_j denote the same physical object in the environment. l_i is the local feature, g_j the global map feature. Given two pairings p_{ij}, p_{kl}, a geometric constraint is a condition on l_i and g_j or a condition on l_i and l_k and on g_j and g_l. Geometric constraints direct the search in the space of all possible data associations and reduce enormously the complexity of the problem. Since we deal with uncertain geometric information all comparisons use the Mahalanobis distance and a significance level α. We can classify geometric constraints into two categories:

2.1.1 Location Independent Constraints

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

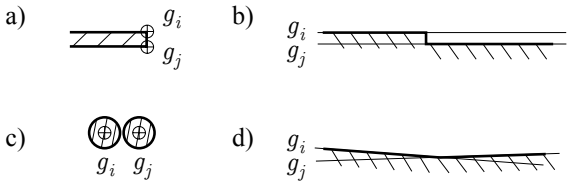

Figure 2. *Examples of feature types which are typically subject to low feature discriminance: a) angle features modeling corners, c) point features modeling columns and b) and d) line features modeling walls. Less critical are features of higher parameter dimensionality as segments or circles or features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

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 by simple preprocessing of the map.

Binary constraints always apply to the features of two pairings. Binary constraints are used to validate whether two local features are consistent with two global features (function satisfy_binary_constraints). Examples include relative measures such as distance or angle.

2.1.2 Location Dependent Constraints

Location dependent constraints come into play as soon as a robot position is available. The fundamental constraint 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).

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. In this sense, the robot can be 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. In [4] extension is satisfied if the observed segment is fully contained in the model segment (they overlap).

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

  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
          accept ← true
          for p_p ∈ S_h while accept
            accept ← satisfy_binary_constraints(p_p, p)
          end
          if accept then
            h' ← h
B           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
          H ← H ∪ generate_hypotheses(h', L\{l}, G)
        end
      end
    end
    H ← H ∪ 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 algorithm returns the set of generated location hypotheses H.*

2.2 Global Localization Using Geometric Constraints

The problem of mobile robot localization is formulated as a matching problem using geometric constraints [8][4] [12]. 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 [10] 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.

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, it is added to the supporting set $S_h = \{p_1, p_2, \ldots, p_p\}$, the location estimate is refined (function estimate_robot_location) and the function recurs in a depth-first manner (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.

2.2.1 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 υ 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:

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

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

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

$$\nabla h \cdot \xi(k+1) = \upsilon(k+1). \quad (3)$$

Assigning zero weight to the odometry-based state predic-

tion can be elegantly done by setting its inverse – the information matrix – to zero

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

By substituting equation (4) into equations (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.

$$\nabla h \cdot \hat{x}(k+1|k+1) = \upsilon(k+1) \quad (5)$$

The solution of (5) is obtained via the pseudoinverse

$$\nabla h' = (\nabla h^T \nabla h)^{-1} \nabla h^T. \quad (6)$$

where we can distinguish between $\nabla h^T \nabla h$ 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

Unable to represent location ambiguity, single hypothesis tracking can go lost as discussed in section 1.1. Even if a technique for global localization were available, we would then need a method to *detect the lost situation*. But detecting lost situations is difficult. It is a case of estimator inconsistency which loosely speaking means that the state moments make mutually incompatible statements. In figure 1 an incorrect pairing is formed, causing the robot to be wrongly aligned to model corner g_3 instead of g_2, and though, its uncertainty to collapse. Tools from estimation theory for estimator inconsistency detection (e.g. testing on innovation whiteness [3]) offer no means to discover that the robot is actually lost.

The step towards a solution of this problem is to loosen the strict distinction of being localized and being lost. With $n = |H|$, the current number of hypotheses, we introduce the following terms: the robot is *lost* if $n = 0$, the robot is *not localized* if $n > 1$, and the robot is *localized* if $n = 1$.

3.1 Hypothesis Generation During Tracking

As pointed out in [6], robot navigation deals with two types of uncertainty: uncertainty in the *values* of measurements and uncertainty in the *origin* of the measurements. The interpretation of l_2 in figure 1 is ambiguous since l_2 could be g_2 or g_3. By representing several statistically possible interpretations as multiple robot location hypotheses, we will be able to cope with the case of figure 1.

Therefore we look for an algorithm which re-generates hypotheses during tracking as soon as there is no guarantee anymore that the correct interpretation can be found (figure 3). This property has track_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. 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_t. Again, the second recursion call implements the extra branch in the interpretation tree that allows correct associations in the presence of outlier observations. After track_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. Given the supporting set S_{h_i}, the robot location is estimated and h_i is admitted to the new H.

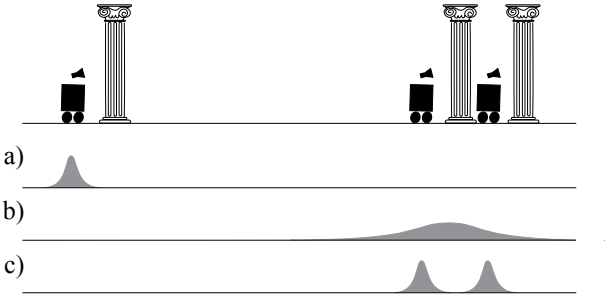

Figure 3. *The fundamental idea behind multi-hypothesis pose tracking of algorithm 2: A well localized robot in a) moves and observes a* single *feature in b) where it is impossible to say which is the correct pairing in view of the uncertainties. Instead, the hypothesis splits up in c) representing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

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

  if L = {} then
    H_t ← H_t ∪ {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_t ← H_t ∪ track_hypothesis(h_p, L \ {l}, G)
        end
      end
    end
    H_t ← H_t ∪ track_hypothesis(h, L \ {l}, G)
  end

return H_t
end
```

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

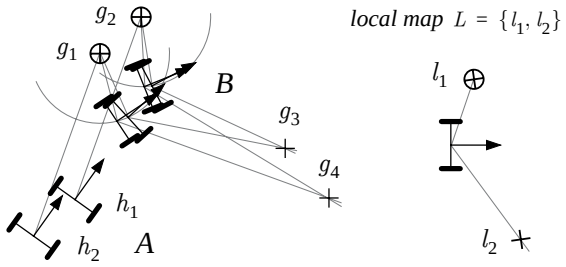

Figure 4. *An example of hypothesis duplication. Given a local map with a (x,y)–point feature l_1, and a ϕ–angle feature l_2, hypotheses h_1, h_2 split up each into four offsprings after an uncertain movement A to B. This results in eight hypotheses at B, four of them being redundant.*

- $|H_v| = 0$, *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. 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 shown in figure 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 $|H| = 1$.

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. 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

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

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

with χ^2_α a value chosen from a χ^2-distribution with three degrees of freedom (refer to [2] for more details).

3.2.2 Duplicate Rejection

Unlike Bayesian approaches to multi-hypothesis localization, 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 geometric quality. This is measured by the joint Mahalanobis distance which 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 is basically the sum of the weighted squared error distances (residuals). The best hypothesis is the one minimizing this sum 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 manually guide the robot through a virtual environment. Local maps are generated by ray tracing where a 360° range finder with 1° resolution and a maximal range of two meters is simulated. 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. 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 current simulation employs infinite lines as features. 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 5 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. Error growth factors have been magnified by a factor of two 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 ten 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.

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 6a). 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 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 obser-

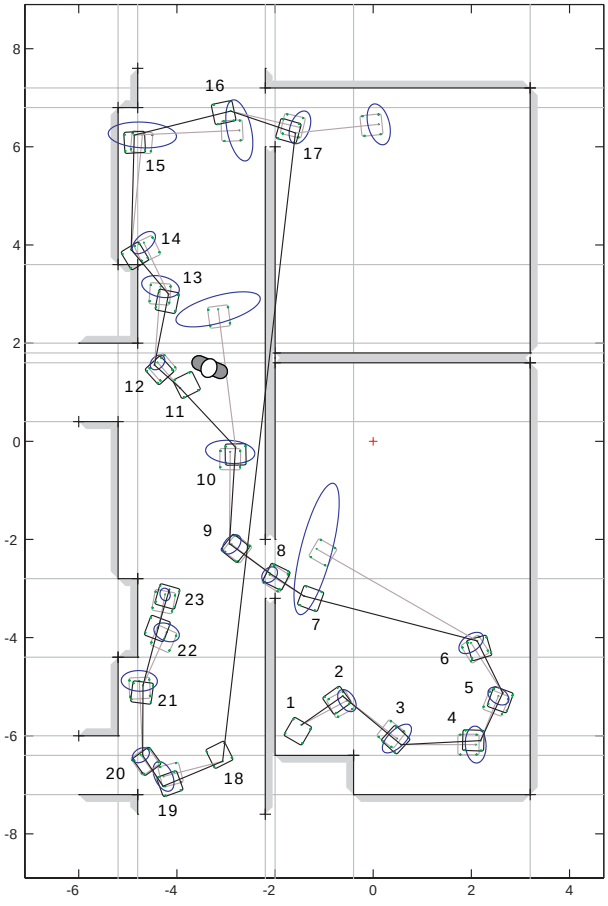

Figure 5. *The simulated test path. Besides extensive odometry uncertainties and errors, the robot collides with a person at step 11 and gets kidnapped at step 18.*

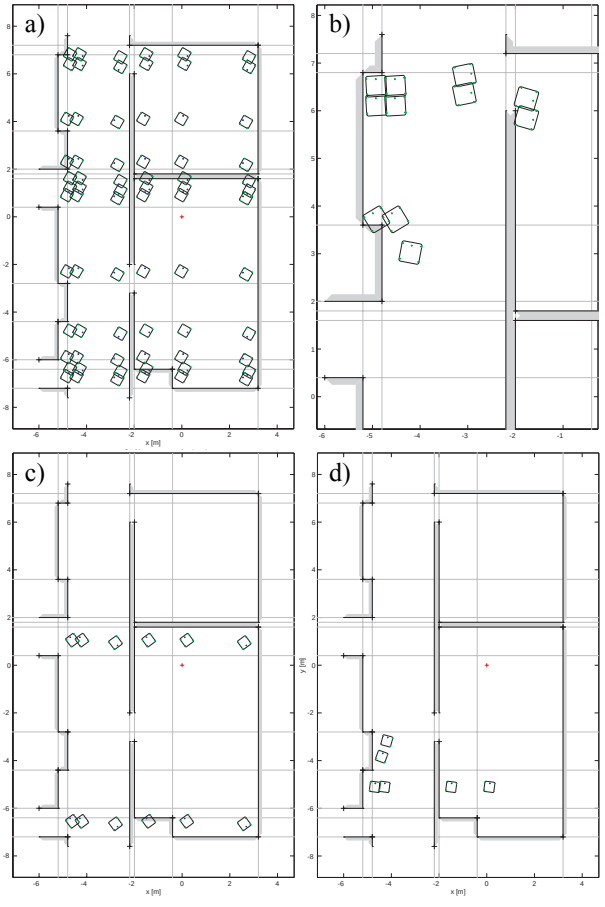

Figure 6. *Hypotheses set H at a) step 1, b) steps 13-17, c) step 20 and d) steps 21 (four hypotheses), 22 and 23. Ellipses in fig. 5 and fig. 6 denote 95% probability levels.*

vation 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 twelve hypotheses (figure 6c) which can be falsified during the last steps up to the true one (figure 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.

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. Furthermore, after steps 1, 12 and 20 where the hypotheses are generated without an a priori position estimate, we can state a fast convergence toward the true hypothesis (figure 7). This although infinite lines provide only minimalistic environment information.

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 track_hypothesis looks for feasible pairings.

First results with the implementation on an embedded system (PowerPC at 300 Mhz) in a similar environment confirm the algorithm's efficiency: around 110 ms average localization cycle time (hypotheses tracking) and rarely more than 400 ms for hypotheses generation. This was measured with the robot under full CPU load (real-time OS, obstacle avoidance, controllers, communication, etc.)

5. Conclusions

In this paper we presented a probabilistic feature-based approach to multi-hypothesis global localization and tracking using geometric constraints. We further addressed the issue of stable multi-hypothesis tracking with the same search technique. The result is a localization approach with an optimal hypothesis management – dependent on location ambiguity arising during navigation, we maintain as many hypotheses as necessary and as few as possible.

From the experiments we conclude that the presented approach is practical and exhibits the degree of robustness

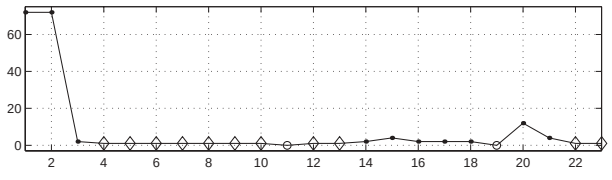

Figure 7. *Number of hypotheses. Diamonds (steps 4-10,12,13,22,23): the robot is* localized, *circles (steps 11,19): the robot is* lost, *points: the robot is* not localized.

which was initially required. With the results for the average 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 extensive real-world experiments and its application in an upcoming robot exhibition project of significant size. Further, the explicit representation and treatment of cases of non-unique EIF location estimates is an issue worth to be addressed as well.

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), 2001.

[2] Arras K.O., Castellanos J.A., Schilt M., Siegwart R., "Towards Feature-Based Multi-Hypothesis Localization and Tracking", 4th Europ. Workshop on Advanced Mobile Robots, Lund, 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, 1999.

[6] Cox I.J., Leonard J.J., "Modeling a Dynamic Environment Using a Bayesian Multiple Hypothesis Approach," Artificial Intelligence, 66(2), p. 311-44, 1994.

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

[8] Drumheller M., "Mobile Robot Localization Using Sonar", IEEE Trans. Pattern Analysis and Machine Intelligence, 9(2), p. 325-32, 1987.

[9] 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.

[10] Grimson W.E.L., Lozano-Pérez, "Localizing Overlapping Parts by Searching the Interpretation Tree", IEEE Trans. Pattern Analysis and Machine Intelligence, 9(4), p. 469-82, 1987.

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

[12] Lim J. H. and J. Leonard J.J., "Mobile Robot Relocation from Echolocation Constraints", IEEE Trans. Pattern Analysis and Machine Intelligence, 22(9), p. 1035-41, 2000.

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

[14] Simmons R., Koenig S., "Probabilistic Navigation in Partially Observable Environments", Proceedings of the Int. Joint Conf. on Artificial Intelligence, vol. 2, p. 1660-7, 1995.