Hybrid Mobile Robot Localization using Switching State-Space Models

Haris Baltzakis and Panos Trahanias

Abstract— **In this paper we focus on one of the most important issues for autonomous mobile robots: their ability to localize themselves safely and reliably within their environments. We propose a probabilistic framework for modelling the robot's state and sensory information based on a Switching State-Space Model. The proposed framework generalizes two of the most successful probabilistic model families currently used for this purpose: the Kalman filter Linear models and the Hidden Markov Models. The proposed model combines the advantages of both models, relaxing at the same time inherent assumptions made individually in each of these existing models.**

Keywords— **Localization, Kalman Filters, Hidden Markov Models, Switching State-Space Models, Hybrid Models.**

I. Introduction

BUILDING autonomous mobile robots is one of the primary goals of robotics and artificial intelligence research. In order to carry out complex navigational tasks, an autonomous robotic agent must be able to provide answers to the "Where am I?" question, that is, to localize itself within its environment.

To reduce the inherent complexity associated with this problem, adoption of appropriate geometric constrains in combination with effective modelling of related information is required. The achieved abstraction, not only makes robotic problems computationally feasible, but also provides robustness in the presence of noise, or other, often unpredictable, factors. Successful probabilistic models proposed for this purpose in the past, generally fall into two major categories: Hidden Markov Models (HMM) [1], [2], [3] and Kalman filters [4], [6], [7].

In [8] a comparison of these two major classes of approaches has been performed. The results indicate the superiority of the Kalman filter approaches with respect to computational efficiency, scalability, and accuracy. On the other hand, HMM-based approaches were proved to be more robust in the presence of noise and/or unreliable odometry information. This comparative work also encourages work in hybrid approaches, noting that combination of these techniques could lead to localization methods that inherit advantages from both.

In this paper we propose a probabilistic framework for modelling the robot's state and sensory information, based on Switching State-Space Models. The proposed approach combines the advantages of both models mentioned above, relaxing at the same time inherent assumptions made individually in each of these existing models. High-level features, consisting of sequences of line segments and corner points, extracted robustly from laser range data, are used to facilitate the implementation of the model, while, a fast dynamic programming algorithm is used in order to produce and score matches between them and an a-priori map.

Experimental results have shown the applicability of the algorithm for indoor navigation tasks where the global localization capabilities of the HMM approaches and efficiency and accuracy of the Kalman filter based approaches are required at the same time.

Institute of Computer Science, Foundation for Research and Technology – Hellas (FORTH), P.O.Box 1385, Heraklion, 711 10 Crete, Greece and Department of Computer Science, University of Crete, P.O.Box 1470, Heraklion, 714 09 Crete, Greece.

Email:{xmpalt,trahania}@ics.forth.gr

II. background

In probabilistic terms, localization is the process of determining the likelihood of finding the robot's state x_t at time instant t, $1 \leq t \leq T$, given a sequence of observations $\{y\}_1^T = y_1, ..., y_T$, that is determining the probability $P(x_t|\{y\}_1^T)$. State vector x_t is usually composed of the 2d position and orientation of the robot with respect to some arbitrary 2d coordinate system, while, the vector of observations y_t expresses sensor and odometry readings.

In practice, it is too difficult to determine the joint effect of all observations up to time T. A common assumption made in order to overcome this difficulty is that the hidden state vectors obey the Markov independence assumption (often referred also as the static world assumption), so that the joint probability for the sequences of states x_t can be factored as

$$P(\{x\}_1^T, \{y\}_1^T) = P(x_1)P(y_1|x_1)\prod_{t=2}^{T} P(x_t|x_{t-1})P(y_t|x_t) \quad (1)$$

For localizing the mobile agent at time t, observations $\{y\}_1^t$ up to time t are available and the related problem (often referred as the Inference problem) can be formulated as:

Localization Problem: Compute the probability

that the robot is at pose z at time t given all observations up to time t, i.e. compute $p(x_t = z|\{y\}_1^t)$.

In order to compute the probabilities $p(x_t = z|\{y\}_1^t)$, eq. 1 should be integrated over all possible states, which results in the recursive formula

$$P(x_t|\{y\}_1^t) = \beta \cdot \int P(x_t|x_{t-1}, y_t) P(x_{t-1}|\{y\}_1^{t-1}) dx_{t-1} \tag{2}$$

where β is a scaling factor ensuring that $P(x_t|\{y\}_1^t)$ sums up to unity over all possible states. It is obvious from eq. 2 that all information about the past history of the robot can be represented by the distribution $P(x_{t-1}|\{y\}_1^{t-1})$.

The formulation stated above dictates that practically any probabilistic localization algorithm must address the following points [8]:

1. How is the prior distribution $P(x_{t-1}|\{y\}_1^{t-1})$ represented?
2. How is the posterior distribution $P(x_t|\{y\}_1^t)$ calculated?

Depending on the method employed to tackle the first point, two major classes of approaches exist, leading to different treatments for the second point.

A. The discrete case

When the state variables x_t become discrete, the integrals in eq. 2 become sums and the prior probabilities can be computed and stored explicitly. For this purpose, grids or topological graphs are utilized to cover the space of all possible states and keep probabilities for each individual element. This leads to the Hidden Markov Model (HMM), a completely general model in terms of what the transition and the emission probabilities can be. Different variants of this scheme for localization and map building [9], [1], [3] have been developed and their characteristics have been shown experimentally [8], [10]:

- Ability (to some degree) to localize the robot even when its initial pose is unknown.
- Ability to deal with noisy measurements, such as from ultrasonic sensors.
- Ability to represent ambiguities.
- Computational time scales heavily with the number of possible states (dimensionality of the grid, size of the cells, size of the map)
- Localization accuracy is limited by the size of the grid cells.

To cope with the last two remarks, [11] has proposed a dynamic scheme for selectively updating only the belief states of the most likely parts of a fine-grained grid. More recently, [2] has proposed a method for representing the belief state via a fixed number of samples, namely Monte Carlo localization.

B. The continuous case

When the state variables x_t are continuous, probabilities must be calculated analytically and stored implicitly. For the problem to remain tractable, gaussian distributions are assumed for this purpose, and eq. 2 leads to a set of linear equations [12], [13].

$$x_t = A_t x_{t-1} + w_t \tag{3}$$
$$w_t \sim N(0, \Gamma_t) \tag{4}$$
$$y_t = C_t x_{t-1} + v_t \tag{5}$$
$$v_t \sim N(0, \Sigma_t) \tag{6}$$

In such models, often called Linear Dynamical Systems (LDSs), the Kalman filter update equations [14], [15] are used in order to propagate recursively the state probability distribution forward in time. It can be proven that they give rise to optimal solutions with respect to virtually any relevant criterion [14]. If the transition and/or the observation functions are not linear (common case in robot localization), a linearization procedure is been applied, resulting to the so-called Extended Kalman Filter (EKF).

Kalman filters and their extensions have been extensively used for robot navigation tasks [4], [6], [7] and are characterized by the following properties:

- Requirement of known initial state of the robot.
- Perform very accurately if the inputs are precise.
- Inability to recover from catastrophic failures caused by erroneous matches or incorrect error models.
- Inability to track multiple hypotheses about the location of the robot.
- Computational efficiency.

C. Comparison and hybrid approaches

In [8] a comparison of the two major classes of approaches presented above has been performed. The results indicate the superiority of the Kalman filter approaches with respect to computational efficiency, scalability, and accuracy. On the other hand, HMM-based localization approaches were proved to be more robust in the presence of noise and/or unreliable odometry information. In [10], a similar comparison between three representative mapping algorithms has been performed. The results confirmed the above stated observations.

Very recently, [16] and [17] have proposed methods that combine elements from both model families presented above. In [16] a method is proposed for localization in "network-like" environments. Kalman filtering is used for position tracking in one-dimensional straight line segments (such as roads or sewer system pipes) and Discrete HMM is used in order to describe transition of one line segment to the other (e.g. when

the robot/vehicle reaches a crossroad). In [17] a localization scheme is proposed for indoor robots based on a very coarse grid. Discrete Markovian dynamics are used in order to describe the robot transition from one cell to the other. However, within each grid cell, robot position is computed by matching sensor information to a-priory learned histograms, while robot orientation is always kept constant by the use of a compass.

III. Proposed approach

As outlined in the previous section, Kalman filter and Discrete HMM-based approaches have complementary advantages. Kalman filtering is efficient and accurate, but confined to small Gaussian sensor perturbations, while discrete Markov modelling is more robust to noise and large amounts of uncertainty.

In this paper we introduce a hybrid model that merges ideas underlying on both the Kalman filter and the HMM approaches. A central concept in our model is to let the Discrete Markov Models handle the topological aspects of the problem and Kalman filters the metric aspects. This is achieved by assuming multiple Kalman trackers assigned to multiple hypotheses about the robot's state and letting discrete Markovian dynamics handle the probabilistic relations among these hypotheses. We show how line segments and corner points extracted from laser range data can be utilized in order to dynamically (a) generate, (b) Kalman-update and (c) assign probabilities to hypotheses.

An abstract graphical representation of the proposed model is depicted in fig. 1. In such models, known as Switching State-Space Models [18], the probabilistic relation between the observations $\{y\}_1^T$ is modelled by the utilization of M real-valued state vectors, x_t^m and one discrete state vector s_t. Each real-valued state vector evolves to linear Gaussian dynamics, while, the discrete state, s_t, called "the switch", is modelled as a multinomial variable that can take on M values $S_t \epsilon \{1, ..., M\}$ and transitions according to Markovian Dynamics. The function of the the switch s_t is to choose the proper real-valued state vector that most likely represents the correct position of the robot within its environment. Such models have been successfully employed in econometrics, signal processing and other fields and efficient algorithms for solving the inference and learning tasks in such networks have been proposed [18].

To successfully apply the proposed model to the localization task, appropriate modelling of all parameters involved is very significant. In the next section we describe the modelling of the robot state and sensory information. Further, higher order, modelling of sensor information in order to extract a powerful set

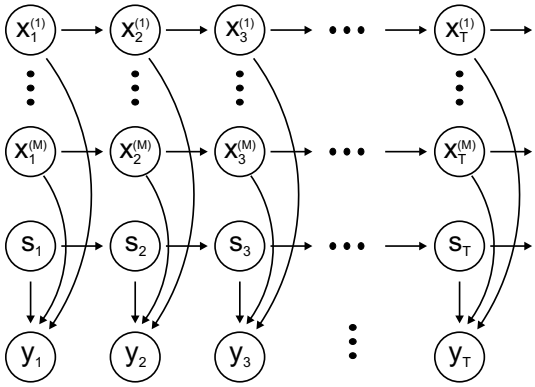

Fig. 1. A Switching State-Space Model.

of features consisting of ordered sets of line segments and corner points is described in section III.B. In sections III.C and III.D we give the Kalman filter and the HMM State-update equations.

A. State and sensor modelling

A.1 Robot state model

Robot's state at time t is modelled as a Gaussian distribution $x_t \sim N(\mu_{x_t}, \Sigma_{x_t})$, where $\mu_{x_t} = (x_t, y_t, \theta_t)^T$ is the mean value of robot's position and orientation, and Σ_{x_t}, the associated 3x3 covariance matrix.

A.2 Odometry model

A robot action α_t at time t, as measured by the odometry encoders of the robot, can also be modelled as a Gaussian distribution of a translation d_t followed by a rotation f_t. That is, $\alpha_t \sim N((f_t, d_t)^T, \Sigma_{a_t})$.

Odometry errors Σ_{a_t} are computed as a function of three constants k_r (range error), k_t (turn error) and k_d (drift error) that can be experimentally calculated.

$$\Sigma_{a_t} = \begin{bmatrix} k_r d_t & 0 \\ 0 & k_t f_t + k_d d_t \end{bmatrix} \quad (7)$$

A.3 Transition model

According to odometry measurements, α_t, the robot's state x_t can be updated using the formulas

$$\mu_{x_{t+1}} = Exp(F(\mu_{x_t}, \alpha_t)) = \begin{bmatrix} x_t - d_t \sin(\theta_t) \\ y_t + d_t \cos(\theta_t) \\ \theta_t + d_t \end{bmatrix} \quad (8)$$

$$\Sigma_{x_{t+1}} = \bigtriangledown F_x \Sigma_{x_t} \bigtriangledown F_x^T + \bigtriangledown F_\alpha \Sigma_{\alpha_t} \bigtriangledown F_\alpha^T \quad (9)$$

where Exp is the expectation operator, F the transition function and $\bigtriangledown F_x$ and $\bigtriangledown F_\alpha$ denote the Jacobians of F with respect to μ_{x_t} and α_t.

A.4 Sensor model

A range measurement r at angle ϕ can be transformed into Cartesian coordinates by a function R

$$\begin{aligned}[x,y]^T &= Exp(R(r,\phi)) \\ &= [r\cos(\phi), r\sin(\phi)]^T \end{aligned} \quad (10)$$

The rangefinder error in polar coordinates is obtained as

$$\Sigma_{polar} = \begin{bmatrix} k_\phi \phi & 0 \\ 0 & k_{\rho_0} + k_{\rho_1} r \end{bmatrix} \quad (11)$$

where k_ϕ, k_{ρ_0}, k_{ρ_1} are parameters related to the range finder angular error, range error independent of distance, and distance-dependent range error, respectively. The transformation of Σ_{polar} to Cartesian coordinates is

$$\Sigma_p = \bigtriangledown R \Sigma_{polar} \bigtriangledown R^T \quad (12)$$

where $\bigtriangledown R$ is the Jacobian of R with respect to (ϕ, r).

B. Feature extraction

Line segments and corner points extracted out of laser range measurements constitute the feature set currently utilized by the proposed methodology. Line segments are extracted directly from range data, while corner points are computed as the intersection points of subsequent line segments.

B.1 Line segments

For line segment extraction, a three-stage algorithm has been implemented. Range measurements are initially grouped to clusters of connected points according to their Sphere-of-Influence graph [19]. Clusters are then further-segmented to points belonging to the same line segment by utilization of the well-known Iterative-End-Point-Fit (IEPF) algorithm [20]. IEPF recursively splits a set of points until a distance related criterion is satisfied.

After range points have been clustered into groups of collinear points, a recursive Kalman-filter-based method is used in order to "best-fit" lines to them, minimizing the squared radial distance error from the position of the range measuring device to each point. The exact process is as follows.

Line $l = N((l_f, l_d)^T, \Sigma_l)$ is initially defined by its extreme points p_1 and p_n. Recursively, each other point p_t, $(1 < t < n)$, is used in order to update the line estimate l. A prediction of point $p_t \sim N((x_t, y_t)^T, \Sigma_{p_t})$ is computed as the trace on l of the line passing through the sensor and p_t. Defining P as the function that computes the trace, yields:

$$\begin{aligned}(x_t^-, y_t^-)^T &= Exp(P(l_f, l_d)) \\ &= \begin{bmatrix} \frac{l_d \tan(u)}{\tan(u)\cos(l_f) - \sin(l_f)} \\ \frac{-l_d}{\tan(u)\cos(l_f) - \sin(l_f)} \end{bmatrix} \end{aligned} \quad (13)$$

where $u = \tan^{-1}(y_t/x_t)$. The EKF update equations[4] can be written as

$$(l_f, l_d)'^T = (l_f, l_d)^T + K((x_t, y_t)^T - (x_t^-, y_t^-)^T) \quad (14)$$

$$\Sigma_l' = \Sigma_l - K \bigtriangledown {P_l}^- \Sigma_l \quad (15)$$

where $\bigtriangledown {P_l}^-$ is the Jacobian of P with respect to l, obtained by linearizing about the prediction point, and K, the Kalman gain given by

$$K = \Sigma_l \bigtriangledown {P_l}^{-T} (\bigtriangledown {P_l}^- \Sigma_l \bigtriangledown {P_l}^{-T} + \Sigma_{p_t})^{-1} \quad (16)$$

B.2 Corner points

Corner points are computed at the intersection points of directly adjacent line segments. A corner point is defined as $c \sim N((c_x, c_y)^T, \Sigma_c)$ where $(c_x, c_y)^T = I(f_1, d_1, f_2, d_2)$ is the intersection of lines $l_1 \sim N((f_1, d_1)^T, \Sigma_{l_1})$ and $l_2 \sim N((f_2, d_2)^T, \Sigma_{l_2})$, and Σ_c is the related covariance matrix, computed as

$$\Sigma_c = J_1 \Sigma_{l_1} J_1^T + J_2 \Sigma l_2 J_2^T \quad (17)$$

J_1, J_2 are the jacobians of I with respect to $(f_1, d_1)^T$ and $(f_2, d_2)^T$, respectively.

C. Kalman filtering

Line segments extracted by the procedure described above are matched to an a-priori known set of line segments (map). An EKF is then used, that employs sequentially each matched pair, in order to track the robots state over time.

The transition model of the Kalman filter, that is, the function used to project state estimates forward in time (prediction step) is given according to eqs. 8 and 9 as

$$\mu_{x_{t+1}^-} = F(\mu_{x_t}, \alpha_t) \quad (18)$$

$$\Sigma_{x_{t+1}^-} = \bigtriangledown F_x \Sigma_{x_t} \bigtriangledown F_x^T + \bigtriangledown F_\alpha \Sigma_{\alpha_t} \bigtriangledown F_\alpha^T \quad (19)$$

Using this predicted state, a known line segment l is predicted as

$$l_{t+1}^- = H(\mu_{x_{t+1}^-}) \quad (20)$$

where

$$H = (\theta_t - l_f, l_d - x_t \cos(l_f) - y_t \sin(l_f))^T \quad (21)$$

The difference between the predicted line segment l_{t+1}^- and the measured line segment l_{t+1} is the measurement residual (Kalman Innovation) and can be written as

$$r_{t+1} = l_{t+1} - l_{t+1}^- \quad (22)$$

$$\Sigma_{r_{t+1}} = \bigtriangledown F_{x_{t+1}^-} \Sigma_{x_{t+1}^-} \bigtriangledown F_{x_{t+1}^-}^T + \Sigma_{l_{t+1}} \quad (23)$$

where Σ_{l_t+1} is the measured line segment covariance and $\bigtriangledown F_{x_{t+1}^-}$ is the Jacobian of F, obtained by linearizing about the state prediction x_{t+1}^-.

The Kalman gain is computed as

$$K_{t+1} = \Sigma_{x_{t+1}^-} \bigtriangledown F_{x_{t+1}^-} \Sigma_{r_{t+1}^-}^{-1} \quad (24)$$

and, finally, the update to the state prediction is:

$$\mu_{x_{t+1}} = \mu_{x_{t+1}^-} + K_{t+1} r_{t+1} \quad (25)$$

$$\Sigma_{x_{t+1}} = \Sigma_{x_{t+1}^-} - K_{t+1}^T \Sigma_{r_{t+1}} K_{t+1}^T \quad (26)$$

The success of any Kalman filtering method for localization tasks relies heavily on the correct matching of features. Various methods for this purpose have been proposed in the past [4], [5].

In this paper we propose a fast and accurate method based on a dynamic programming string-search algorithm. The proposed algorithm originates from a class of dynamic programming algorithms [21] very widely used for matching, aligning and measuring the degree of structural similarity between biological sequences.

Given a pair of sequences, the dynamic programming algorithm performs a global wraparound alignment of the second sequence on the first, finding all solutions that produce a maximum score according to a scoring function. As an example, consider the character sequences: ABBCDEFG and BCEAA. The algorithm attempts to align BCEAA sequence on ABBCDEFG, wrapping around the edges of the latter if necessary. The result depends on the scoring function assumed. A typical scoring function, as the one used in out implementation, produces the following two, equally scored, alignments:

```
ABBCDEFG ABBCDEFG
  || |    |
  BC E    AA
  BC E   AA
```

For the purpose of this work, a simple scoring function was been chosen that equally rewards matches if the measurement residual of eq. 22 is bellow a large threshold, while permitting gaps in both the measured line segment sequence and the map, although slightly penalizing the former.

The dynamic programming matching method exploits information contained in the spatial ordering of the features, while, its dynamic programming implementation, furnishes it with computational efficiency.

D. Discrete Model and Switching Operation

The Kalman filtering method discussed in the previous section is unable to perform global localization. Even if correct feature matches were available, the unimodality of Gaussian distributions, used to express the belief of the robot about its state, confines the robot to a single hypothesis about its state. An obvious case where the Kalman filter fails to achieve global localization is robot "kidnapping" [3].

In this work, utilization of a supervising algorithm is used in order to dynamically generate hypotheses about the state of the robot. Kalman trackers are assigned to each hypothesis in order to dynamically update them through time. Moreover, discrete Markov dynamics are introduced in order to govern the probabilistic relation among hypotheses.

For generating hypotheses, corner points, extracted as described in section III.B, are utilized. At time instant t, the most confident corner point is picked out of the set of extracted features. For computing the confidence measure of a corner point, variances of its $x-y$ coordinates, as well as variances in its adjacent line segment orientations are considered. A fast search through the map is performed, by comparing the difference in orientations of the corner point's adjacent line segments (that is, the angle of the corner point). All matches are used in order to generate hypotheses about the robot's state.

Since corner points are characterized by their (x, y) position coordinates as well as the orientations f_1, f_2 of their adjacent line segments, the problem of determining the state of the robot given a corner point match is over-constrained (four equations with three unknowns). A weighted least squares solution yields

$$h = (\bigtriangledown C^T \Sigma_c \bigtriangledown C)^{-1} \bigtriangledown C^T \Sigma_c^{-1} c \quad (27)$$

where $c = (x, y, f_1, f_2)^T$, the observed quadruple corresponding to corner point at (x, y), Σ_c the related covariance matrix, and $\bigtriangledown C$ the jacobian of the observation function, i.e. the function mapping map corner points to predicted corner points, given the state $(c = C(h, s))$.

After new hypotheses are generated and merged with the already existing set of hypotheses, probabilities are assigned to them. For this purpose eq. 2 can be rewritten as

$$P(h_t^k) = \beta \cdot P(f_t|h_t^k) \sum_i P(h_{t-1}^i) P(h_t^k|h_{t-1}^i, \alpha_t) \quad (28)$$

where $P(f_t|h_t^k)$ is the probability to observe feature set f_t given that hypothesis h_t^k is correct, $P(h_t^k|h_{t-1}^i, \alpha_t)$ is the probability of jumping from hypothesis h_{t-1}^i to hypothesis h_t^k given the odometry measurements α_t, and β is a normalizing factor ensuring that all probabilities sum up to one. The summation index i spans all hypotheses generated at previous time step $(t-1)$.

$P(f_t|h_t^k)$ is computed for each hypothesis as the score returned by the dynamic programming algorithm at this point, normalized by the number of visible line segments. Intuitively, this heuristic gives a fast and

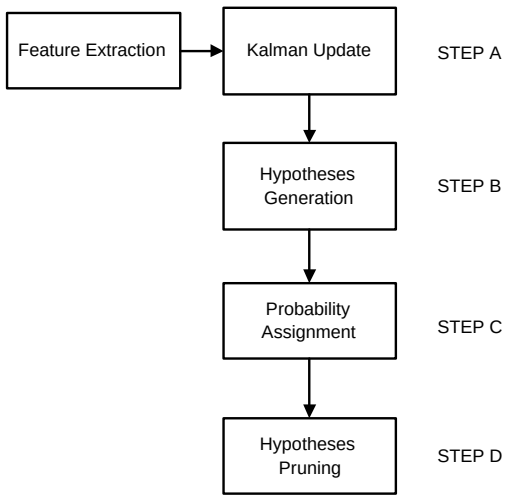

Fig. 2. Operation of the Hybrid algorithm.

precise way of determining the similarity of what is actually seen from a robot's state with what should have been seen. Alternatively, in cases that the measured line segments are detected to be of low confidence, at the cost of a slight computational overhead, row range data can be utilized in order to compute $P(f_t|h_t^k)$. This is achieved by projecting them on the map and determining the degree of overlapping.

On the other hand, $P(h_t^k|h_{t-1}^i, \alpha_t)$ is computed by a fast sampling algorithm as the value of h_{t-1}^i displaced by odometry readings α_t, integrated over a small neighborhood of h_t^k.

Figure 2 gives an overview of the operation of the complete localization algorithm. Initially (Step A), extracted features are used in order to Kalman-update existing hypotheses. In step B, new hypotheses are added to the hypotheses set by searching the map for matches of a single corner point in the observed feature set, while, in step C, hypotheses are assigned new probabilities by considering discrete Markovian dynamics. Finally, in Step D, duplicate hypotheses are merged and improbable hypotheses are removed. Evidently the procedure of generating new hypotheses need not be repeated with every iteration of the Kalman update procedure. Moreover probability assignment to hypotheses can also be performed in a lower frequency. Depending on the needs of the application, the complexity of the environment and the available computing power, the algorithm can be tuned to operate at any intermediate point in between a pure Kalman tracker and an HMM-based localizer.

IV. Experimental results

The probabilistic framework proposed in this paper has been assessed using a variety of test data acquired by a robotic platform of our laboratory, namely an iRobot-B21r, equipped with a SICK-PLS laser range finder. The range finder is capable of scanning 180 degrees of the environment, with an angle resolution of one measurement per degree and a range measuring accuracy of 5cm.

Extensive tests have also been performed with simulated data for various environments and varying odometry and range measuring resolution and accuracy.

Figure 3 demonstrates the first iterations of the algorithm, attempting to perform global localization in a simulated environment. Each of the depicted robots corresponds to a hypothesis, with the probability of the hypothesis being displayed next to it. The hypothesis corresponding to the correct state of the robot is marked with a bold arrow pointing to it, while all 95% position uncertainty ellipses depicted are magnified, for displaying purposes, by a factor of 10. Although simulated odometry errors as well as measuring errors have been set to very large values, it takes only a few iterations for the robot to become very confident about the correct state.

The effectiveness of the algorithm in performing global localization when the robot is "kidnaped" is demonstrated in fig.4. The robot is artificially moved from the actual position that it had correctly estimated in fig. 3c to a new position. It takes only a couple iterations for the robot to reset the probabilities of all hypotheses and to regain confidence about its new position.

Figure 5 demonstrates the operation of the algorithm in a corridor structure outside our lab. A handcrafted a-priori map, consisting of a dense, ordered set of line segments, has been used for this purpose. Although the corridor structure favors ambiguities about the robot position, and corner points are not visible for large operation intervals, the algorithm successfully determines the correct robot position after only a few iterations. In fig. 5a, the algorithm bootstraps by generating hypotheses according to matches of the only currently visible corner point (corner point C). By the time it reaches its position in fig. 5b, it is already confident enough about its state. However, to completely resolve all ambiguities, it has to reach the middle of the corridor (fig. 5d). Afterwards, the algorithm continues to track the robot state, while all new hypotheses generated are eventually eliminated due to low observation and/or transition probabilities.

The ability of the algorithm to rapidly localize the robot in complicated real environments is further demonstrated in fig. 6.

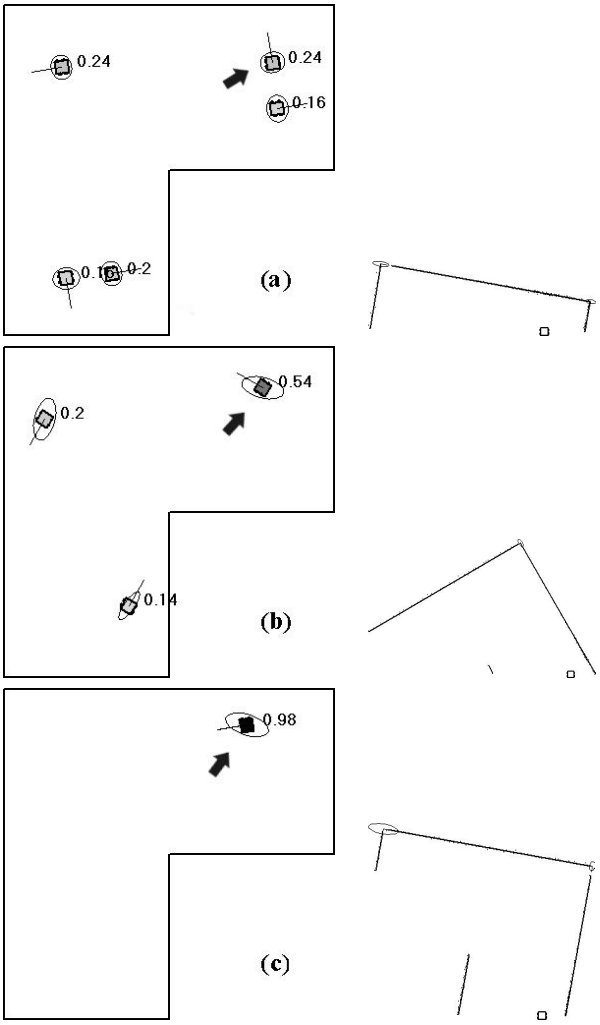

Fig. 3. Global Localization in a simulated environment. left: Position Estimation, right:Extracted Features; 95% uncertainty ellipses are magnified by 10. (a) Initial estimate, (b) estimate after 5 iterations, (c) estimate after 9 iterations.

V. Conclusions and future work

In this paper we have proposed a probabilistic framework for modelling problems related to the localization task for mobile robots. The proposed approach, based on an underlying Switching State-Space model, generalizes two of the most successful probabilistic model families used for this purpose, namely the Kalman filter linear models and HMMs.

Sequences of line segments and corner points, extracted from laser range data, in combination with a fast dynamic programming algorithm, able to produce and score matches among them, have been used in order to facilitate the implementation of the proposed methodology.

It is in our intension to further investigate utilization of other sources of sensory data, such as vision,

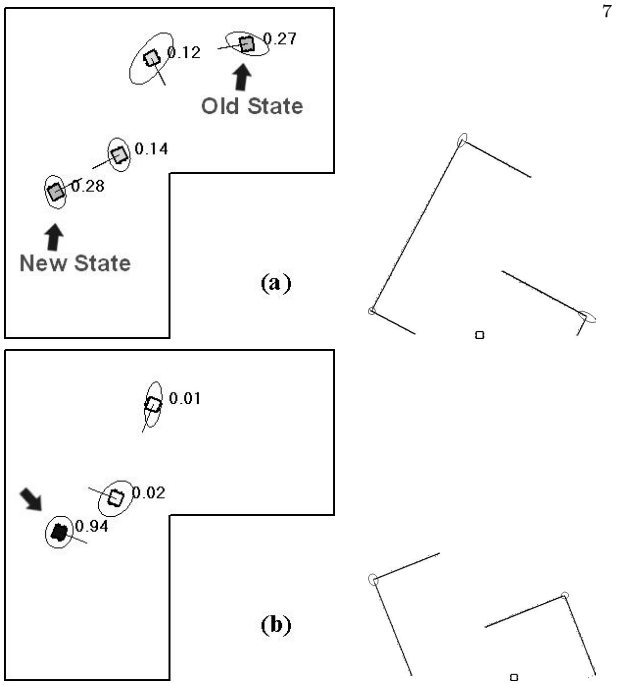

Fig. 4. Global Localization for the "kidnaped robot" case; 95% uncertainty ellipses are magnified by 10. (a) Robot is "kidnapped" to a new state, (b) estimate after 6 iterations.

in order to utilize sensor information not included in 2d range scans. Moreover, adaptation of existing mapping algorithms to the requirements of our localization model is a topic of current research.

References

[1] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien, "Acting under uncertainty: Discrete bayesian models for mobile robot navigation," in *Proc. IEEE/RSJ Intl. Conf. Intell. Robots and Systems*, 1996.

[2] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *IEEE Intl. Conf. Robotics and Automation (ICRA99)*, May 1999.

[3] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Creemers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Probabilistic algorithms and the interactive museum tour-guide robot Minerva," *Intl. J. of Robotics Research*, vol. 19, no. 11, pp. 972–999, Nov. 2000.

[4] F. Lu and E.E. Milios, "Robot pose estimation in unknown environments by matching 2d rangescans," *J. Intell. and Robotic Syst.*, vol. 18, pp. 249-275, 1997.

[5] J. S. Gutmann, T. Weigel and B. Nebel "Robot pose estimation in unknown environments by matching 2d rangescans," *Advanced Robotics*, vol.14, no.8, pp. 651–668, 2001.

[6] J.A. Castellanos, J.M. Martínez, J. Neira, and J.D. Tardós, "Simultaneous map building and localization for mobile robots: A multisensor fusion approach," in *In proc. IEEE Int. Conf. Robotics and Automation, ICRA'98*, May 1998, pp. 1244–1249.

[7] K.O. Arras, N. Tomatis, and R. Siegwart, "Multisensor on-the-fly localization using laser and vision," in *In Proc. IEEE/RSJ Intl. Conf. Intell. Robots and Systems (IROS 2000)*, Oct. 2000.

[8] J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige, "An

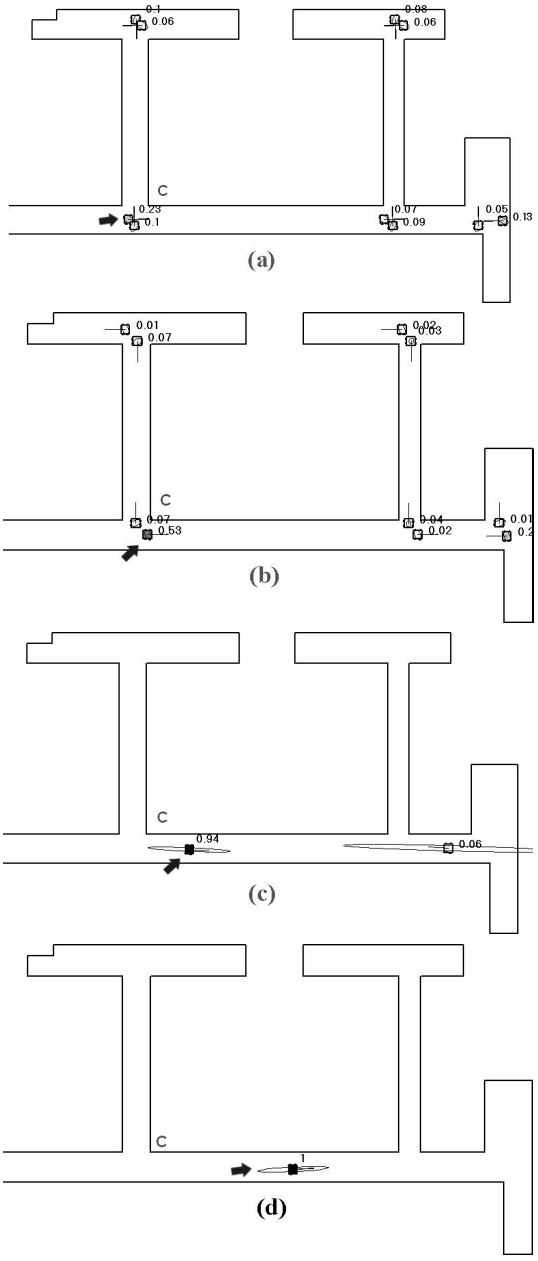

Fig. 5. Global localization in a corridor structure; 95% uncertainty ellipses are magnified by 10.

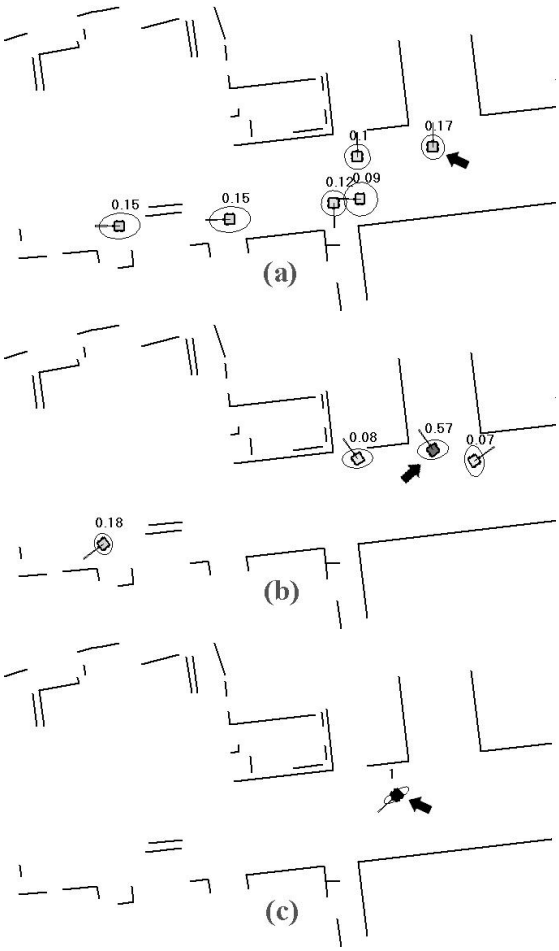

Fig. 6. Global localization in a real environment; 95% uncertainty ellipses are magnified by 10.

experimental comparison of localization methods," in *Proc. IEEE/RSJ Intl. Conf. Intell. Robots and Systems*, 1998.

[9] I. R. Nourbakhsh, R. Powers, and S. Birchfield, "DERVISH - an office-navigating robot," *AI Magazine*, vol. 16, no. 2, pp. 53–60, 1995.

[10] A. Scott, L.E. Parker, and C. Touzet, "Quantitative and qualitative comparison of three laser-range mapping algorithms using two types of laser scanner data," in *IEEE Syst., Man, and Cybern. Conf.*, Oct. 2000.

[11] W. Burgard, A. Derr, D. Fox, and A.B. Cremers, "Integrating global position estimation and position tracking for mobile robots: the Dynamic Markov Localization approach," in *Proc. IEEE/RSJ Intl. Conf. Intell. Robots and Systems*, 1998.

[12] T. P. Minka, "From hidden markov models to linear dynamical systems," Tech. Rep. TR-531, Vision and Modeling Group, MIT Media Lab, 1999.

[13] K. P. Murphy, "Learning Switching Kalman Filter Models," Tech. Rep. 98-10, Compaq Cambridge Res. Lab, 1998.

[14] P. Maybeck, *Stochastic Models, Estimation ond Control*, vol. 1, Academic, New York, 1979.

[15] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software*, Artech House Inc., Boston, MA, USA, 1993.

[16] S. Thiebaux and P. Lamb, "Combining kalman filtering and markov localization in network-like environments," in *Pacific Rim Intl. Conf. AI*, 2000, pp. 756–766.

[17] T. Duckett and U. Nehmzow, "Mobile robot self-localisation using occupancy histograms and a mixture of gaussian location hypotheses," *Robotics and Autonomous Systems*, vol. 34, no. 2–3, pp. 119–130, 2001.

[18] Z. Ghahramani and G. Hinton, "Switching state-space models," Tech. Rep. CRG-TR-96-3, Dept. of Computer Science, Univ. of Toronto, 1996.

[19] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley-Interscience, New York, 1973.

[20] T. S. Michael and Thomas Quint, "Sphere of influence graphs in general metric spaces," *Mathematical and Computer Modelling*, vol. 29, pp. 45–53, 1994.

[21] G. Benson, "Sequence alignment with tandem duplication," in *Proc. 1st Annual Intl. Conf. Computational Molecular Biology (RECOMB)*, Santa Fe, NM, 1997, pp. 27–36, ACM Press.