Combining Kalman Filtering and Markov Localization in Network-Like Environments

Sylvie Thiébaux and Peter Lamb

CSIRO Mathematical & Information Sciences
PO Box 664, Canberra 2601, Australia
`First.Last@cmis.csiro.au`

Abstract. This paper presents a hybrid localization method designed for environments having the structure of a network (road networks, sewerage networks, underground mines, *etc...*). The method, which views localization as a problem of state estimation in a switching environment, combines the flexibility and robustness of Markov localization with the accuracy and efficiency of Kalman filtering. This is achieved by letting Markov localization handle the topological aspects of the problem, and Kalman filtering the metric aspects. The two techniques are closely coupled: the Markov model determines the Kalman filters to be initiated, and statistics computed by the Kalman filters are used to define the transition and observation probabilities in the Markov model. This approach has been applied to the problem of localizing a motor vehicle traveling on an urban road network, providing robust and accurate localization at low cost.

1 Introduction

A recent comparison of localization methods [9] indicates that Kalman filtering based techniques [12, 10, 3] and Markov localization [13, 6, 8] have complementary advantages. Kalman filtering is efficient and accurate, but is confined to small Gaussian sensor perturbations. Markov localization is robust to a larger amount and a wider range of uncertainty, but is computationally much more demanding while comparatively less accurate. As suggested in [9], judiciously combining these techniques could therefore result in hybrid methods that are efficient, accurate, flexible and robust.

The present paper describes such a hybrid localization method, designed for environments having a network structure. By network, we mean a set of straight line segments connected at their extremities. The segments are to be treated as one-dimensional (lateral or vertical displacement is irrelevant), so by localizing, we mean determining the current segment and the current coordinate from the origin of this segment. An application example, used in this paper, is localizing a motor vehicle traveling on an urban road network. This is a prerequisite to a number of tasks in the field of Intelligent Transport Systems, such as on-board vehicle navigation [16]. Another example is localizing robots navigating in, inspecting and repairing a network of sewerage pipes [11].

To achieve effective localization in these environments, we propose to view localization as a problem of state estimation in a switching environment (see [2] for the seminal paper), and to approach it using methods for multiple hypothesis tracking (see [4, pp. 450-483] for a survey). This amounts to letting Markov localization handle the topological aspects of the problem and Kalman filtering the metric aspects. At the topological level, a Markov model accounts for the progression from segment to segment, and maintains a probability distribution about the path (history of segments) followed. This permits a number of concurrent hypotheses to be considered about the current segment. At the metric level, a distinct Kalman filter is assigned to each of the most likely paths, and tracks the distance traveled along the current segment of this path. The two levels are closely coupled: the Markov model determines the Kalman filters to be initiated, and statistics computed by the Kalman filters are used to define the transition and observation probabilities in the Markov model.

This combination offers the best of both underlying techniques for the type of environment considered. While delivering the required accuracy, it reduces the complexity of Markov localization if it were used alone, yet inherits its flexibility and robustness (in particular the ability to operate with little *a priori* information on the starting location). It also avoids the need for explicit map-matching which usually accompanies Kalman filtering based techniques: since each Kalman filter operates only in one spatial dimension (the dimension of a segment), the location it produces is already constrained to lie on the network.

The method is particularly well-suited to applications for which (1) Kalman filtering techniques alone are inadequate because reliable sensory information is too expensive or unavailable, and (2) pure Markov localization is also unsatisfactory because landmarks are too rare to allow for quantization up to the required accuracy. The sewerage robot and motor vehicle localization problems share these properties. For instance, GPS signals are inaccessible in sewerage pipes which are installed meters below street level, and the high slip in these pipes makes reliable odometry difficult. Although GPS receivers are available to motor vehicles, their cost contributes to making navigation systems too expensive for the non-luxury car market. In both domains, natural landmarks are only present at the extremities of the network's segments (junctions between pipes, road intersections), and not inbetween.

In this paper, we shall present the principles of our hybrid method, using a motor vehicle localization problem for illustration. The example is chosen so as to expose the key features of the method and facilitate understanding of how this latter can address other application domains and more complex situations. The next section presents a Kalman filter operating in a single spatial dimension to estimate the distance traveled on a given segment. Then, we describe the use of Markov localization to determine the current path, and explain how the two techniques are coupled. The presentation in these sections does not assume familiarity with the respective techniques. We go on to give experimental results on field and simulated data, before concluding with some notes on related work.

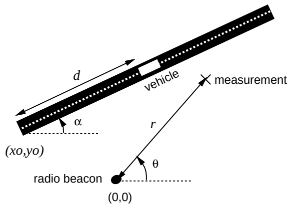


Fig. 1. Vehicle traveling on a road segment

2 One-dimensional Kalman Filter

The purpose of a discrete-time Kalman filter is to give an optimal estimate of the state of a system at discrete points of time, taking into account the dynamics of this system, the available sensor information, and the uncertainty on these which is assumed to be Gaussian.[1] The basic type of Kalman filter handles *linear* systems, that is systems for which observations and dynamics can be expressed as linear systems of equations:

$$\mathsf{z}(t) = \mathsf{C}(t)\mathsf{x}(t) + \mathsf{w}(t) \tag{1}$$

$$\mathsf{x}(t+1) = \mathsf{A}(t)\mathsf{x}(t) + \mathsf{B}(t)\mathsf{u}(t) + \mathsf{v}(t) \tag{2}$$

Let us detail the functions involved in turn, using concrete examples from a vehicle localization application.

$\mathsf{x}(t)$, the *state vector*, is a vector of n variables representing the state of the system at time t. Here in order to track a vehicle traveling on a single road segment (as in Figure 1), we model one spatial dimension, namely the distance d separating the vehicle from the segment's origin (x_0, y_0). We choose to represent the state of the vehicle by a third-order model comprising the distance d, the vehicle's speed $\dot{d}$, and its acceleration $\ddot{d}$. So $\mathsf{x}(t) = [d, \dot{d}, \ddot{d}]'_t$.

$\mathsf{z}(t)$, the *measurement vector*, is a vector of p variables representing the observations made at time t, if any. In our trials below, observations come from an experimental beacon-based positioning system[2] which measures the angle θ

---

[1] Let us recall that an n-dimensional Gaussian variable x (i.e. a vector of n Gaussian variables) is characterized by two quantities: its mean m, an n-dimensional vector, and its covariance C, a symmetric $n \times n$ matrix. This is written $\mathsf{x} \sim \mathrm{N}(\mathsf{m}, \mathsf{C})$. $\mathsf{C}_{i,i}$ is the variance of x_i and $\mathsf{C}_{i,j}$ $(i \neq j)$ indicates the extent to which x_i and x_j are correlated. The probability density at x is:

$$\Pr(\mathsf{x}) = \frac{\exp\left[-0.5(\mathsf{x}-\mathsf{m})'\mathsf{C}^{-1}(\mathsf{x}-\mathsf{m})\right]}{\sqrt{\det\left[2\pi\mathsf{C}\right]}}$$

Note that the exponent indeed evaluates to a scalar, and that $'$ refers to the transpose of a vector or a matrix.

[2] Without loss of generality, we use this positioning system which is much less reliable than GPS but also less expensive, to show that our method can cope with a large amount of noise and yield an affordable localization system.

and range r between a radio beacon and the vehicle (see Figure 1). From this, we can easily derive the x-y coordinates[3] of the vehicle relative to the segment's origin as $x = r\cos\theta - x_0$ and $y = r\sin\theta - y_0$, and take $\mathsf{z}(t) = [x, y]'_t$.

$\mathsf{C}(t)$, the *state to measurement matrix*, is a $p \times n$ matrix mapping the state vector into the measurement vector that would be obtained at time t in the absence of sensor uncertainty. Obviously here, if α is the angle between the horizontal axis and the segment, then:

$$\mathsf{C}(t) = \begin{bmatrix} \cos\alpha & 0 & 0 \\ \sin\alpha & 0 & 0 \end{bmatrix} \quad \text{for all } t$$

$\mathsf{w}(t)$, the *measurement noise vector*, is a p-dimensional Gaussian variable with mean zero and covariance matrix $\mathsf{R}(t)$, representing the uncertainty affecting the observations. For our vehicle localization example, it can be shown that if σ_r^2 and σ_θ^2 are the respective variances for the r and θ measured at time t, then (provided that σ_θ is small):[4]

$$\mathsf{R}(t) = \begin{bmatrix} r^2\sigma_\theta^2\sin^2\theta + \sigma_r^2\cos^2\theta & \cos\theta\sin\theta(\sigma_r^2 - r^2\sigma_\theta^2) \\ \cos\theta\sin\theta(\sigma_r^2 - r^2\sigma_\theta^2) & r^2\sigma_\theta^2\cos^2\theta + \sigma_r^2\sin^2\theta \end{bmatrix}_t$$

$\mathsf{A}(t)$, the *state transition matrix*, is an $n \times n$ matrix mapping the state vector at time t into the successor state vector that would be obtained at time $t+1$ in the absence of external action (or *input*) intending to control the system. Let T be the sampling interval separating two consecutive time steps t and $t+1$, then elementary kinematics yields the following transition matrix for the vehicle localization example:

$$\mathsf{A}(t) = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad \text{for all } t$$

$\mathsf{u}(t)$, the *input vector*, is a q-dimensional vector representing the input applied to the system at time t, and $\mathsf{B}(t)$, the *input gain*, is the $n \times q$ matrix translating the input into the corresponding modifications in the state vector. For a motor vehicle, the input typically corresponds to the rate of change in acceleration ($\dddot{d}$ or *jerk*) initiated by the driver, but unfortunately, this quantity is unknown to us. Another alternative which is commonly chosen when tracking robots is to use odometry information as (part of) the input. However, in the experiments below this possibility was not available to us either, since our vehicle had no interface to the odometer.[5] So for this particular application we are forced to ignore the

---

[3] A two dimensional space is sufficient for our application, however adding a third dimension is straightforward.

[4] Note that measurements (r, θ) are a *non-linear* function of the state, so we should in principle use an Extended Kalman Filter (EKF) [4, ch. 10]. Our way of computing $\mathsf{R}(t)$ is effectively an EKF approach, but which we found more stable.

[5] Odometry information would certainly improve the performance of the system. We are able to show that our method operates reliably even when such information is lacking.

input term in equation (2) and treat the jerk as the uncertainty affecting the dynamics of a vehicle operating otherwise at constant acceleration.

This leads us to $\mathsf{v}(t)$, the *process noise vector*, an n-dimensional Gaussian variable with mean zero and covariance matrix $\mathsf{Q}(t)$, which represents the uncertainty on the system's dynamics. Here $\mathsf{v}(t) = [0, 0, T\ \ddddot{d}\,]'_t$, where $\ddddot{d} \sim \mathrm{N}(0, \sigma_j)$. For our experiments, we took σ_j to be the average of the second derivative of a speed profile (speed vs time) used for fuel consumption testing [1, p.19]. Using basic kinematics, we derived the following covariance matrix for $\mathsf{v}(t)$:

$$\mathsf{Q}(t) = \sigma_j^2 \begin{bmatrix} T^6/36 & T^5/12 & T^4/6 \\ T^5/12 & T^4/4 & T^3/2 \\ T^4/6 & T^3/2 & T^2 \end{bmatrix} \quad \text{for all } t$$

How does the Kalman filter work? For a formal treatment we refer to a standard textbook, e.g. [4]. Briefly, the Kalman filter starts with an initial estimate of $\mathsf{x}(1)$ characterized by its mean $\hat{\mathsf{x}}(1)$ and its covariance matrix $\mathsf{P}(1)$. At each step, equations (1) and (2) and the previous state estimate $\hat{\mathsf{x}}(t)$ are used to calculate a prior estimate $\hat{\mathsf{x}}^-(t+1)$ of the new state, the observations $\hat{\mathsf{z}}(t+1)$ that should result, and the respective covariance matrices. When measurement $\mathsf{z}(t+1)$ is made, the filter *innovation*, i.e., the difference $\mathsf{i}(t+1) = \mathsf{z}(t+1) - \hat{\mathsf{z}}(t+1)$ between the actual and predicted observations is computed, and the prior state estimate is revised using Bayes' rule to optimally account for the discrepancy. The assumptions of linearity and Gaussian uncertainty together guarantee that the state estimate at each time step is Gaussian. This give rise to a very economical update procedure, which consists of a few elementary matrix operations.

A well known result (see [4]) is that the innovation is also a zero-mean Gaussian variable with the useful property that the probability of the observations at time $t+1$ given the past history – i.e., the observation history $Z(t) = \mathsf{z}(1)\ldots\mathsf{z}(t)$ combined with equations (1) and (2) – equals the probability density of the innovation:

$$\Pr(\mathsf{z}(t+1) \mid Z(t)) = \Pr(\mathsf{i}(t+1))$$

As the Kalman filter procedure also involves computing the covariance matrix of the innovation, the above probability, which we will need in the next section, is readily available.

3 The Markov Model

The one-dimensional Kalman filter will correctly track the distance traveled on a single segment. To handle the whole network[6] however, we need a further mechanism enabling us (a) to maintain multiple hypotheses about the current segment, (b) to initiate new filters to operate on new hypothesized segments when approaching a junction, and (c) to compute the most likely location overall.

---

[6] Curved roads can easily be accommodated as piecewise linear approximations and handled as a special case of the operations on junctions presented here.

To this end, we model segment change as an abrupt change of linear system, which we handle using methods for multiple hypothesis tracking [2, 14, 15, 4]. This amounts to the embedding of one-dimensional Kalman filters in a Markov model of the path (segment history) taken through the network, and to viewing segment change as a change of state in this Markov model.

Like Kalman filters, discrete Markov models are designed to give a Bayesian estimate of the current state of a system, given information about the dynamics of this system, and about the way it can be observed. Markov models are however much more generally applicable than Kalman filters and in return much more computationally demanding. In particular they are not limited to Gaussian noise. The uncertainty on the current state is represented by an arbitrary probability distribution on the set of states, often called the *belief state*. In lieu of the observation equation of the Kalman filter, an arbitrary *observation probability* distribution specifies the probability of making some observation given the current state, and in lieu of the the dynamics equation of the filter, an arbitrary *transition probability* distribution specifies the probability of the next state given the current one.

Here, the *states* of the Markov model at time t are the segment histories $S(t) = s(1) \dots s(t)$, a subset of which represent possible paths through the network. We denote the probability of segment history $S(t)$ in the belief state at time t as $\mathrm{b}(S(t))$. To each segment history $S(t)$ is assigned a Kalman filter which tracks the distance traveled on the current segment $s(t)$ of this history. Naturally, the state to measurement matrix of this filter is parametrized by the angle $\alpha_{s(t)}$ of that segment. So the observation equation for that filter is:

$$\mathsf{z}(t) = \mathsf{C}_{s(t)}(t)\mathsf{x}(t) + \mathsf{w}(t)$$

The *transition probabilities* $\mathrm{Tr}(S(t), S(t+1))$ for the Markov model are defined by examining the events that may affect the path: at time $t+1$ we may stay on the current segment $s(t)$, or we may have turned onto a connected segment. The first type of event does not require initiating a new Kalman filter, while the second type does. What is the probability of each of these events? If d is the distance component of the state estimate $\hat{\mathsf{x}}(t)$ tracked by the filter assigned to history $S(t)$, and l is the length of segment $s(t)$, then the probability of remaining on $s(t)$ is $\Pr(0 \leq d \leq l)$. Everything being equal, the probability of turning on any of the n_1 segments connected to the origin of $s(t)$ is $\Pr(d < 0)/n_1$, and similarly the probability of turning on any of the n_2 segments at the other extremity is $\Pr(d > l)/n_2$ (see Figure 2). d being Gaussian, these probabilities are easily calculated. Naturally, this simple scheme can be elaborated to handle more complicated situations, as required. For instance, advice to make a certain turn given by a vehicle navigation system or by a robot navigation policy should result in that turn being assigned a higher probability than others.

As for the *observation probabilities* of the Markov model, they can be easily computed by noting that the probability of observing $\mathsf{z}(t)$ when traveling along segment history $S(t)$ is simply the probability $\Pr(\mathsf{i}_S(t))$ of the innovation in the filter assigned to that history. Again, this is the default: depending on the

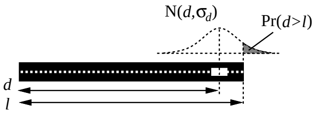


Fig. 2. Probability of exiting a segment

application, we may want to incorporate here observations of a topological nature that were irrelevant at the Kalman filter level. For example, sewer robots are able to distinguish (with some probability of error) between several types of junctions connecting sewerage pipes, and can use this information for self-localization [11].

At each time step the belief state of the Markov model is updated using a straightforward application of Bayes' rule. So when at time $t+1$ we extend segment history $S(t)$ by appending segment $s(t+1)$, then following measurement $\mathsf{z}(t+1)$, the probability $\mathrm{b}(S(t+1))$ of the resulting history is:

$$\mathrm{b}(S(t+1)) = \lambda \Pr(\mathrm{i}_S(t+1)) \mathrm{Tr}(S(t), S(t+1)) \mathrm{b}(S(t))$$

where λ is the normalizing factor.

At any time, the location returned by the system is that indicated by the Kalman filter assigned to the most probable segment history. Alternatively, we could use the expectation of the locations indicated by all the Kalman filters operating on the most probable segment. The more usual approach of using the expectation of the location estimates over all segment histories is not used here, because the resulting location will typically not lie on the network.

Even if the number of possible paths through the network is significantly lower than the total number of segment histories, the number of filters involved in the above estimation process grows exponentially with time. Since most segment histories have extremely low probabilities, it therefore makes sense to keep the computational cost bounded by pruning a large number of them. To this end, we act on two parameters. Firstly, we set a threshold p_{thresh} on the transition probabilities such that if $\mathrm{Tr}(S(t), S(t+1)) < p_{thresh}$ then the transition is simply ignored. Secondly, we follow [14] in only keeping the N most probable histories at each time step.[7] The robustness and accuracy of the localization process will depend on the settings for these two parameters, but we have observed that the algorithm operates satisfactorily over a wide range of settings. At present, we do not vary these parameters during a run, but it would be interesting to investigate ways of selecting their value according to the topology of the current junction (angle between the segments involved, connectivity).

---

[7] Further reduction of the tree size can be achieved using techniques from the field of multiple hypothesis tracking, e.g. merging histories having the same current segment and close distance estimate and pruning unlikely histories according to validation gates for the corresponding filters (see e.g. [14]). On the other hand, other well-known heuristics such as Generalised Pseudo Bayes (GPB) or Interacting Multiple Model (IMM) [4, ch. 11] do not seem to be as relevant here, as they deny the assumption that past locations are constrained to lie on the network.

4 Experiments

To validate our approach, we conducted a series of field trials and simulations with the motor vehicle localization system outlined in this paper. Our C++ implementation running under Windows 98 on a 133 Mhz Pentium with 24 Mb of memory proved sufficient to achieve real-time performance on field trials.

As our approach inherits Markov localization's ability to globally (re-)localize the vehicle, the starting location does not need to be known. At the start of each experiment and following a loss of the vehicle location,[8] the belief state was initialized by taking the projection of the first available measurement onto a large number of segments in the area, assigning to each of these segments a probability inversely proportional to the Euclidian distance between the measurement and the nearest point on the segment, and taking the projections to be the initial estimates for the respective filters. We observed that the localization system successfully resolved the ambiguity between the segments within a few seconds after the vehicle started moving.

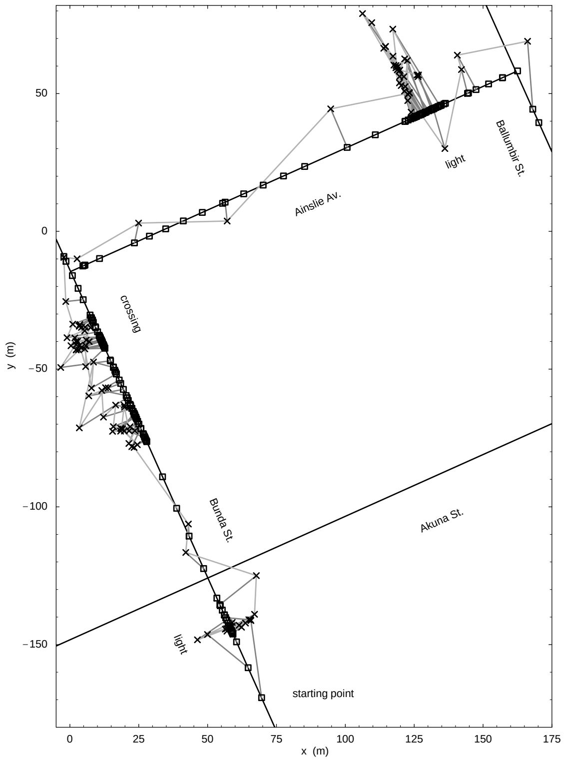
An example of the localization system operating in the center of Canberra is shown in Figure 3(a). The successive measurements are represented by crosses and joined by gray lines. Location estimates returned by the system are represented by boxes and are joined to the corresponding measurements (when available) by dark gray lines.

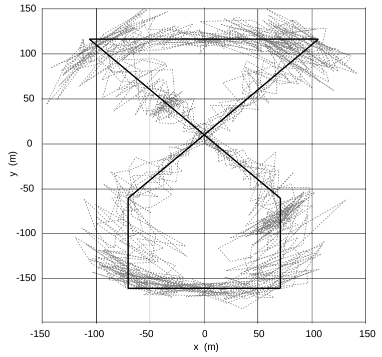
The beacon is located at the respective origins of the axes. Measurements are made at the rate of one per second. The standard deviation of the measurement is around 5.9 m for r and 11.9° for θ. If the direct path to the vehicle is blocked, the measurement may indicate the distance and angle of a reflected path. This can introduce short term bias into the measurements, e.g. see the right hand end of Ainslie. Sometimes, measurements are completely missing due to interference, see e.g. the top of Bunda and the left half of Ainslie where many locations lack corresponding measurements. The Kalman filter is particularly useful in such cases, since it still exploits the vehicle's dynamics to return an estimate in the absence of observation. At each step however, the uncertainty about the location increases. For example, this keeps the probability that the vehicle has turned from Bunda to Ainslie below that of the vehicle maintaining its track along Bunda, until a measurement takes place which forces the transition.

An additional difficulty is caused by traffic lights and pedestrian crossings placed around intersections. When the vehicle is queuing near an intersection, the lateral error of the measurement makes it quite plausible that it has turned on a perpendicular street. This happens for instance at the traffic light at the intersection between Bunda and Akuna. However, provided the transition probability threshold is set to a reasonable value (e.g. $p_{thresh} = 0.01$), good track of the vehicle is kept, even if we only maintain a few filters (e.g. $N = 3$). Figure 3(a) has been produced with these settings. Recovery following a wrong segment selection only takes a few seconds even with sparse measurements, see e.g. the

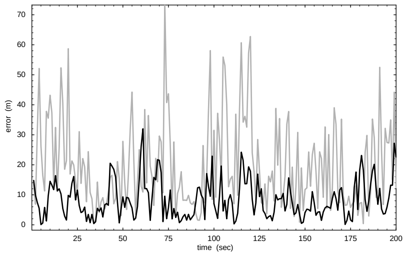
---

[8] The system considers the vehicle location to be lost when the variance of the distance estimate returned by the most probable Kalman filter exceeds a certain threshold.





(b) Synthetic network (black) and noisy measurements (gray)



(a) Field trial (locations= boxes, measurements = crosses)

(c) Error (distance) before (gray) and after processing (black)

Fig. 3. Experiments

intersection between Bunda and Ainslie. We observed that maintaining a larger number of filters improves the accuracy of the location.

The 8 segment synthetic network shown in Figure 3(b) permits an accurate comparison of the raw positioning information with the estimated vehicle location. The simulation experiment we report here is based on a 12 km long car trajectory generated with the speed of the vehicle controlled by the same drive cycle data used to estimate the variance σ_j^2 of the jerk for the Kalman filters. At the middle of the network, the vehicle makes a random turn choice. The beacon is located near this intersection. Measurements following Gaussian distributions with standard deviation 5.9 m for r and 11.9° for θ (i.e., the same as for the above field test) are generated every second. As shown by the dotted lines in Figure 3(b), this leads to a large lateral error for large r.

Because of the large lateral noise, the length of the trajectory, and especially the top acute corners, the network is quite demanding of the algorithm – note that the total number of segments is of minor importance to this algorithm; more relevant is the local connectivity of the network. In order not to lose track of the vehicle at one point or another we find that a minimum of 10 filters must be maintained, and that the transition probability threshold needs to be set to a low value ($p_{thresh} = 10^{-6}$) to ensure that transitions are generated for locations "around the corner".

Figure 3(c) shows the error (distance) between the real vehicle position and the raw positioning information (gray), resp. the estimated location (black), as a function of time for a representative subset (about 1/5) of the trajectory. The localization process improves the average error from 21m to 8m. Occasionally an incorrect segment choice is made as a corner is approached, but the choice is corrected within a few (2-3) seconds of the turn being made. Conventional approaches to motor vehicle location achieve comparable figures only by using much more accurate positioning information integrating GPS and odometry.

5 Conclusion and Related Work

This paper describes a hybrid approach to localization in network-like environments. This approach inherits the accuracy and efficiency of Kalman filtering based techniques, as well as the robustness, generality and globality of Markov localization.

As mentioned above, our way of embedding Kalman filters in a Markov model follows the principles underlying multiple hypothesis tracking for state estimation in switching environments. Other related work includes approaches improving the efficiency and accuracy of Markov localization. For instance dynamic Markov localization [5] is a fine-grained grid-based Markov localization method which selectively updates the likely parts of the belief state, and dynamically modifies the grain of the quantization to adjust to the certainty of the current location. This technique is more general and uniform than ours. However, our method is very likely to prove more efficient for the type of applications considered in this paper, because as evidenced by the vehicle localization problem, acceptable accuracy cannot be delivered by a method based solely on Markov localization without excessive quantization within a segment. Monte Carlo localization [7] is another Markov localization technique which achieves impressive performance by judiciously sampling the belief state. We believe that Monte Carlo localization could profitably be incorporated to our framework to cope with highly connected networks.

Our method has successfully been applied to motor vehicle localization in urban road networks. In this context, it avoids the error-prone map-matching step which underlies most conventional approaches to motor vehicle localization [16], and is able to produce results with comparable accuracy without requiring high performance positioning systems. Our approach is particularly attractive in that respect, since reducing the cost of localization is an important step towards making navigation systems affordable for non-luxury vehicles. We are confident that this approach will prove useful in environments sharing similar properties, for instance in sewerage or underground mine networks to provide an accurate metric localization in real-time. It remains to be seen whether our method can easily be extended to environments having a different structure while still retaining most of its benefits.

Acknowledgements

We thank Jens-Steffen Gutmann, Joachim Hertzberg, Phil Kilby, John Slaney, and a number of anonymous reviewers for their valuable comments on earlier drafts of this paper.

References

1. Methods of test for fuel consumption of motor vehicles designed to comply with Australian Design Rules 37 and 40. Publication AS 2877-1986, Standards Association of Australia, 1986.
2. G. Ackerson and K.S. Fu. On State Estimation in Switching Environments. *IEEE Trans. Aut. Control*, 15(1), 1970.
3. K. Arras and S. Vestli. Hybrid, High-Precision Localisation for the Mail Distributing Mobile Robot System MOPS. In *Proc. ICRA-98*, 1998.
4. Y. Bar-Shalom and X.-R. Li. *Estimation and Tracking: Principles, Techniques and Software.* Artech House, 1993.
5. 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. IROS-98*, 1998.
6. A.R. Cassandra, L. Kaelbling, and J.A Kurien. Acting under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation. In *Proc. IROS-96*, 1996.
7. D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In *Proc. AAAI-99*, pages 343–349, 1999.
8. D. Fox, W. Burgard, S. Thrun, and A.B Cremers. Position Estimation for Mobile Robots in Dynamic Environments. In *Proc. AAAI-98*, pages 983–988, 1998.
9. J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An Experimental Comparison of Localization Methods. In *Proc. IROS-98*, 1998.
10. J.-S. Gutmann and C. Schlegel. AMOS: Comparison of Scan Matching Approaches for Self-Localization in Indoor Environments. In *Proc. EUROBOTS-96*. IEEE Computer Society Press, 1996.
11. J. Hertzberg and F. Kirchner. Landmark-Based Autonomous Navigation in Sewerage Pipes. In *Proc. EUROBOTS-96*, pages 68–73. IEEE Computer Society Press, 1996.
12. P.S. Maybeck. The Kalman Filter: An Introduction to Concepts. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles.* Springer Verlag, 1990.
13. R. Simmons and S. Koenig. Probabilistic Robot Navigation in Partially Observable Environments. In *Proc. IJCAI-95*, pages 1080–1087, 1995.
14. J.K. Tugnait and A.H. Haddad. Detection and estimation scheme for state estimation in switching environements. *Automatica*, 15:477–481, 1982.
15. K. Watanabe and S.G. Tzafestas. Generalised pseudo-bayes estimation and detection for abruptly changing systems. *Journal of Intelligent and Robotic Systems*, 7:95–112, 1993.
16. Y. Zhao. *Vehicle Location and Navigation Systems.* Artech House, 1997.