2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

A Pure Probabilistic Approach to Range-Only SLAM

Jose-Luis Blanco, Javier González, and Juan-Antonio Fernández-Madrigal

Abstract— **Range-Only SLAM (RO-SLAM) represents a difficult problem due to the inherent ambiguity of localizing either the robot or the beacons from distance measurements only. Most previous approaches to this problem employ non-probabilistic batch optimizations or delay the initialization of new beacons within a probabilistic filter until a good estimate is available. The contribution of this work is the formulation of RO-SLAM as an online Bayesian estimation process based on a Rao-Blackwellized Particle Filter. The conditional distribution for each beacon is initialized using an additional particle filter which, eventually, is transformed into an extended Kalman filter when the uncertainty becomes sufficiently small. This approach allows the introduction of new beacons without either delay or any special non-probabilistic processing. We validate our proposal with experiments for both simulated and real datasets.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is one of the central issues required for truly autonomous mobile robots, hence the intense research effort that has been devoted to this field in the last years. One of the most widespread approach consists of using probabilistic techniques (Bayesian inference) to estimate the robot position and the map given the sequence of imperfect actions and noisy observations of the robot. Specific methods have been proposed to cope with the differences caused by using certain map representations, e.g. landmarks or occupancy grids, or robotic sensors, e.g. cameras or laser scanners. For a review of many of these methods the reader can refer to [1], [15].

This paper addresses the problem of SLAM when using range-only sensors. These devices can measure the distance to each one of a set of artificial beacons distributed throughout the environment, identifying them individually. There are two important differences between range-only SLAM (RO-SLAM) and the more common range-bearing SLAM [2]. Firstly, in RO-SLAM we can avoid the problem of data association since most practical devices used for range measurement are able of distinguishing which beacon is being detected, e.g. Ultra-Wide-Band (UWB) devices [5]. Secondly, the information provided by the measures is highly ambiguous: in general, each measurement defines a probability density for the potential positions of the sensed beacon, but for a range sensor the non-negligible part of this density has an annular shape, since the beacon is within a "ring" with radius equal to the range measurement. To illustrate how ambiguous this information can be, consider the motivating example depicted in Fig. 1, where several of these "rings" are shown for different positions of the robot. It can be observed how all the circles pass through the real position of the beacon, although they do not coincide precisely due to sensor noise. A difficult issue in RO-SLAM is the existence of multiple, apparently consistent locations for the beacon, as can be seen in the figure. Recall that, in SLAM, the estimation of the position of the beacons must be carried out simultaneously to the robot localization itself, rendering RO-SLAM even more challenging.

Several works dealing with RO-SLAM have been reported in the last years. In [14] the authors propose a geometric method for adding new beacons to a map using delayed initialization, but a partially known map is required at the beginning. Range-only localization is addressed in [6] and [7] under the classic EKF-based implementation of SLAM, where the authors propose an approximation of the sensor model inspired by the circular-shaped distributions obtained for range sensors. They also address SLAM but assuming a prior knowledge about the beacon locations. Sub-sea RO-SLAM is demonstrated in [10] with good results even with the lack of a reliable ego-motion estimation (such as odometry for ground vehicles). The main difference with the present work is the usage of a least-square error minimization procedure instead of a probabilistic filter. The work in [11] achieves RO-SLAM through a different strategy: firstly, an initial estimation of the position of each beacon is computed using a voting scheme over a 2D grid. An interesting contribution of that work is a preliminary robust filtering of outliers using a graph cut approach. Once the initial estimation converges, a standard EKF deals with the SLAM problem. A similar scheme is adopted in [3], where the authors also explore the possibility of inter-beacon range

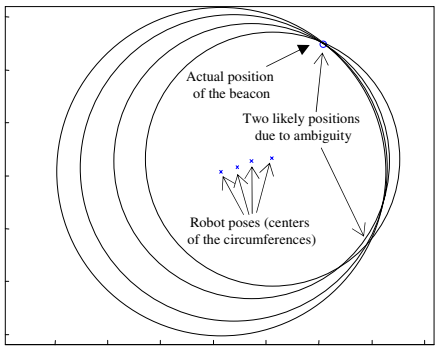


Fig. 1. Example of how ambiguous can become the localization of a beacon from range measurements only. For a valid position hypothesis it is necessary a relatively large distance between the different observations.

This work was supported by the Spanish Government under research contract DPI2005-01391 and the FPU grand program.

The authors are with the Department of System Engineering and Automation, Málaga University, Málaga, 29071, Spain {jlblanco,jgonzalez,jafma}@ctima.uma.es

measurements to improve map building.

The contribution of the present work is thus a formulation of probabilistic RO-SLAM under a pure Bayesian viewpoint and without any additional non-probabilistic step. The usage of a probabilistic framework is motivated by its well-known suitability for effectively fusing information from different sources. In contrast to many previous works, we apply a Bayesian filter from the beginning, taking advantage of a Rao-Blackwellized particle filter (RBPF) [4] to decouple the estimation of the robot poses and the map. By doing so, we can freely choose the most convenient distribution for the beacons at each time step. We derive the equations for adding and updating a beacon to the map as a set of weighted samples, and then converting it into a Gaussian only when the distribution converges to a single location. This leads to a consistent probabilistic framework for RO-SLAM where beacons are inserted the first time they are observed, independently of whether the map already contains well-localized beacons. This follows from the property of conditional independence of the mapped beacons under the RBPF approach ([4], [9]).

Another advantage of this approach is that we maintain the best estimation of each beacon at each time step, and this information is always available to improve the robot localization. In most previous works this information cannot be exploited until the knowledge about the beacon location becomes sufficiently precise.

II. RBPF-BASED SOLUTION TO RO-SLAM

The purpose of a probabilistic approach to RO-SLAM is to obtain the joint probability distribution of the robot pose (or path) and the map, given all the available data at some instant of time. This distribution represents our knowledge about the robot path, the map, and all their correlations.

Motivated by the strong non-Gaussianity of the distributions found in RO-SLAM, i.e. a circular-shaped observation likelihood, we propose to take advantage of the factorization:

$$p\left(x^t, m | z^t, u^t\right) = \underbrace{p\left(x^t | z^t, u^t\right)}_{\text{Robot path}} \underbrace{p\left(m | x^t, z^t, u^t\right)}_{\text{The map}} \tag{1}$$

to separate the representations of the robot path x^t and the map m. Robot actions and observations are denoted as u_t and z_t, respectively. Note the usage of the superscript t to designate sequences of variables from time step 1 to t.

Since we adopt a sample-based representation for the robot path, the result is a Rao-Blackwellized particle filter (RBPF) where a conditioned distribution of the map is stored for each path hypothesis [4]. An important consequence of this approach for our purposes is that, assuming independence between the errors in the measurements, the map density can be further factorized as:

$$p\left(m | x^t, z^t, u^t\right) = \prod_l p\left(m_l | x^t, z_l^t\right) \tag{2}$$

with the m_l being the different individual beacon positions in the map m, and z_l^t being the observations (i.e. ranges) of the corresponding beacons. Note that the actions u_t have been dropped since they do not provide additional information. The factorization in (2) implies the conditional independence between the individual beacons, thus their densities are stored separately and we can employ the kind of representation that is most convenient at each time step without affecting either the robot path or other beacons.

Concretely, for each beacon that is observed for the first time, we add a new auxiliary[1] particle filter (PF) to each one of the RBPF samples in order to perform the Bayesian estimation of the new beacon. As described in the next section, this auxiliary PF will eventually converge from the initial circular shape towards a small Gaussian-like shape, and at this moment it will be replaced (without loss of the estimated uncertainty) by a standard EKF which performs reliably for reduced uncertainties. The switch into an EKF is justified by the particle depletion problem that any standard implementation of a PF eventually suffers [13].

Next we describe the general procedure to iterate the RBPF with each new action and observation from the robot, while the details on how to compute some important terms are derived in the following section.

Let the set of M samples of the path be referenced as $x^{[i],t}$ for $i = 1...M$, where each pose x_t comprises a 2D location plus a heading. These samples have associated importance weights $\omega_t^{[i]}$ and are approximately distributed according to the path posterior, i.e. the left part of the product in (1). As it is common in SLAM, we initialize the filter without any prior knowledge, thus the robot starting location can be arbitrarily taken as the coordinate origin, that is, $x_0 = \mathbf{0}$.

For each time step, new particles are drawn using the robot motion model, which in our case is derived from odometry readings, that is, $x_t^{[i]} \sim p(x_t | x_{t-1}^{[i]}, u_t)$. Next, importance weights are updated as:

$$\omega_t^{[i]} \propto \omega_{t-1}^{[i]} p\left(z_t | x^{t,[i]}, z^{t-1}\right) \tag{3}$$

for which we need a probabilistic observation model, derived in the next section. If necessary, the particles may be resampled to preserve the diversity of the representation. This is typically performed whenever the effective sample size falls below a given threshold [8]. After updating the estimate of the robot path, the corresponding conditional distributions of the map must be also updated to account for the new range readings, as discussed later on.

Although it is not strictly necessary, in order to simplify the exposition we will assume coplanarity between all the beacons and also that the robot moves over a 2D surface from which all the beacons are at a fixed height. Any of these restrictions can be straightforwardly removed from our method at the cost of an increase in the computational burden.

III. IMPLEMENTATION OF THE RBPF

This section describes the two probabilistic representations of beacons in the map (PF and EKF), and how to compute

[1]We denote these filters as *auxiliary* to avoid confusion with the main RBPF. Note that this term is not related at all to the *auxiliary particle filters* introduced in [12].

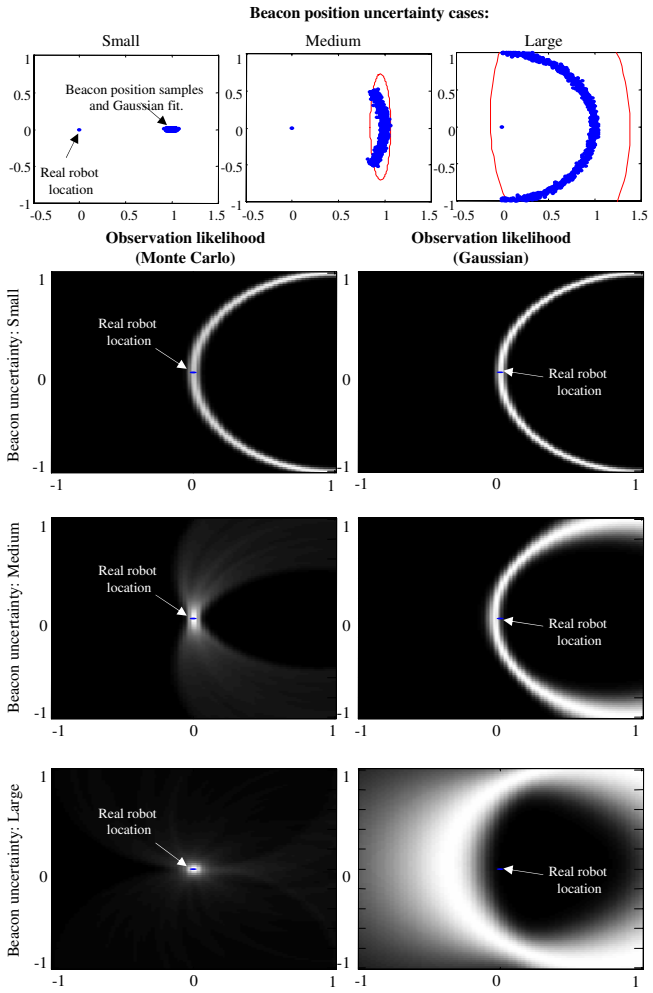


Fig. 2. A comparison of the observation model as computed using the Monte Carlo (MC) (central column) and the EKF (right column) approximations. These two columns represent the observation likelihood as an intensity image over the 2D plane of robot positions, where brighter gray levels correspond to higher values of the likelihood. Each row illustrates a situation with a different level of uncertainty in the beacon localization, represented in the left column. The probability density of the beacon position is depicted in the two forms considered in the text: as a set of samples, and as a Gaussian, in this case computed to fit the samples. Taking the MC approximation as the reference, it can be observed how the EKF approximation performs poorly when the uncertainty in the beacon position becomes excessively large, assigning low likelihood values to the real location of the robot (the origin).

the observation model and update the map for each of those two forms.

A. Observation Model for the Path Estimate

We assume that the range sensor provides measurements z_t corrupted with additive zero-mean Gaussian noise v_t with a variance σ_r^2, that is:

$$z_t = b\left(|x_t - m|\right) + v_t \tag{4}$$

where the function $b(\cdot)$ can be used to emulate transformations to the real distance between the sensor and the beacon, given by $|x_t - m|$. For example, it could be used to compensate the systematic error or *bias* of the device. In this work we assume for simplicity that $b(d) = d$. Note that we omit the beacon index l for clarity in the notation, thus the symbol m will represent a single beacon, not the whole map.

Also notice how the observation model in (4) requires a concrete value for the beacon position m, whereas we only have a distribution of its potential values (recall that at this level there is no uncertainty about the robot path, since the distributions are conditioned a given path hypothesis). In other words, the uncertainty in the location of the beacon becomes uncertainty in the expected range, and thus its density is given by:

$$p(z_t|x^{t,[i]}, z^{t-1}) = \int p(z_t|x_t^{[i]}, m)p(m|x^{t-1,[i]}, z^{t-1})dm \tag{5}$$

Observe that the second term within the integral corresponds to the map hypothesis for each RBPF particle. At this point we can find three different situations depending on the state of the beacon within the i'th particle:

(a) The beacon is not present in the map. This will happen whenever a beacon is observed for the first time. In this case the observation likelihood can be set to any arbitrary constant $p(z_t|x^{t,[i]}, \cdot) = \eta$, since it will have no effects on the estimation of the path: there are no previous references of that beacon that could improve the knowledge about the robot location.

(b) The beacon is represented by an auxiliary particle filter. Then the location of the beacon is approximated by a set of N samples $m^{[i,k]}$ with weights $\beta^{[i,k]}$ for $k = 1...N$. In this case the integral in (5) becomes a sum:

$$\begin{aligned} & p(z_t|x^{t,[i]}, z^{t-1}) = \\ = & \int p(z_t|x_t^{[i]}, m) \sum_{k=1}^{N} \beta^{[i,k]} \delta(m - m^{[i,k]})dm \\ = & \sum_{k=1}^{N} \beta^{[i,k]} p(z_t|x_t^{[i]}, m^{[i,k]}) \\ = & \sum_{k=1}^{N} \beta^{[i,k]} \mathcal{N}(z_t; \left|x_t^i - m^{[i,k]}\right|, \sigma_r^2) \end{aligned} \tag{6}$$

where the last step (replacing the observation model by a Gaussian) follows from the definition in (4) since the observation, conditioned to some known hypotheses of the robot pose and the beacon, only conserves the randomness owed to the Gaussian noise with variance σ_r^2.

(c) The beacon is already represented by a Gaussian, that is, $p(m|x^{t-1,[i]}, z^{t-1}) = \mathcal{N}(m; \hat{m}_t, P_t)$, with $\hat{m}_t$ and P_t standing for the mean and the covariance matrix, respectively. Since the Gaussian representation will be used only for reduced uncertainties in the beacon position, it is acceptable here to employ a first-order propagation of the uncertainty from the beacon to the observation z_t:

$$\begin{aligned} p(z_t|x^{t,[i]}, z^{t-1}) &= \mathcal{N}(z_t|\hat{z}_t, \sigma_t^2) \\ \hat{z}_t &= \left|x_t^i - \hat{m}_t\right| \\ \sigma_t^2 &= HP_tH^T + \sigma_r^2 \end{aligned} \tag{7}$$

where the matrix H is the Jacobian of the function in (4) with respect to the beacon coordinates.

Up to this point, we have defined how to compute the observation likelihood term for the two possible representations of a beacon in the map: a set of Monte Carlo samples, and a Gaussian. To clarify and motivate this distinction, we show in Fig. 2 some examples for the computation of the observation likelihood in three different scenarios, each one corresponding to a row in the figure. The left charts depict the real position of the robot and the current knowledge about the beacon in the two forms: a set of samples, and the Gaussian computed to fit the samples. A remarkable observation is how both representations lead to a similar observation likelihood when the uncertainty is small (first row), while the output of the Gaussian approximation degenerates as the the uncertainty becomes larger. For example, in the third row we have beacon samples distributed along a 180° arc, giving a clearly defined peak of the likelihood at the true robot position. In contrast, the Gaussian approximation assigns higher values to a wide area of the state space but a null value to the actual robot pose. This is a clear consequence of the mismatch between the actual distribution of the beacon and the fitted Gaussian, as observed in the left-bottom chart.

B. Map Update

In this section we address how to initialize and update the densities for each beacon in the map, which correspond to the map part of the factorization in (1).

By applying the Bayes rule and the definition of conditional probability, it can be shown that our estimation of the map m given the new data available at each time step (x_t and z_t) is described by:

$$\underbrace{p(m|x^t, z^t)}_{\text{Posterior}} \propto \underbrace{p(m|x^{t-1}, z^{t-1})}_{\text{Prior}} \underbrace{p(z_t|m, x^t, z^{t-1})}_{\text{Sensor model}} \tag{8}$$

To follow the evolution of the probabilistic representation of the beacon, assume that a new beacon is detected at some instant of time t (not necessarily at the first time step). Then, according to (8) we must multiply the prior belief with the observation likelihood, but due to the absence of any previous knowledge about the beacon it is reasonable to assume an uniform distribution over the whole state space of m. Thus, the first time a beacon is observed (8) reduces to computing $p(m|x^{t,[i]}, z^t) = p(z_t|m, x_t^{[i]})$ (note how m is the only free variable). Here we initialize the Monte Carlo representation of the beacon density by drawing samples $m^{[i,k]}$ along the circle centered at $x_t^{[i]}$ at any direction in the whole 360° range and at a distance of z_t plus the additive random noise – refer to Fig. 4(a) for an example. Since these samples are distributed following exactly the target distribution, we assign them equal initial weights $\beta^{[i,k]}$. The number N of particles to generate at this point is a crucial parameter of our approach: too few particles may lead to a wrong estimation, while an excessive number increases the computational burden. We have obtained good results in different scenarios using the heuristic rule $N = \alpha \cdot z_t$, where α can vary between 400 and 2000. We must remark that, if the first observation of a beacon is an outlier, our approach will fail in correcting its position, thus we assume in this work that outliers have been already discarded.

In subsequent observations of the beacon, these samples are modified to implement the recursion in (8). It can be easily shown that the Bayesian update becomes a change in the weights of these samples as:

$$\beta_t^{[i,k]} \propto \beta_{t-1}^{[i,k]} p(z_t|x_t^{[i]}, m^{[i,k]}) \tag{9}$$

Note that the rightmost term was already computed in (6), thus it does not need to be computed again. Moreover, we have observed that, if the robot is moving, a large part of the particles are quickly assigned negligible weights after a few iterations, thus they can be removed from the set to reduce the computational cost. In our implementation we drop particles with weights below 10^{-5} times the highest weight. This simple strategy reduces the computational burden of our method and leads to a practical implementation, as demonstrated in the experimental results.

As new observations are fused into the map, the beacon estimates will eventually converge towards small areas of the space where most likely the beacons should be found. The test we have applied to check whether a given distribution should be transformed into a Gaussian is to obtain the covariance matrix computed from all the samples $m^{[i,k]}$ and then check whether the major axis of the corresponding ellipsoid is below a given limit. This is implemented as a threshold for the largest eigenvalue of that covariance matrix. The threshold value should be selected to be a few times smaller than the sensor standard deviation (σ_r) in order to assure the quality of the linearized approximations assumed for the Gaussian.

Finally, the update of the beacon distribution in Gaussian form is performed through a standard EKF, linearizing over the range observations z_t.

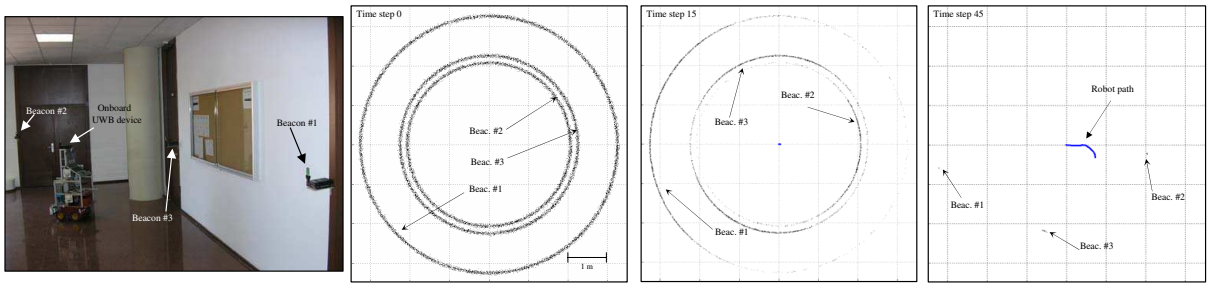
IV. Experimental Results and Simulations

In this section we will firstly validate our proposal with experimental results from a real robot equipped with range-only sensors. Next we discuss the results from simulated data to demonstrate the possibility of adding new beacons at any instant of time. They are also described the typical evolution of the RBPF and the differences in the computation burden between time steps. We encourage to also view the online video[2] for these experiments.

A. Real robot dataset

We have applied the method proposed in this work to a dataset gathered by a real robot while it moves, controlled by a human, throughout a room traversing an overall path of 30 meters. For this experiment we have installed three "PulsON" Ultra-Wide-Band (UWB) devices from TimeDomain as static beacons in the walls at a fixed height, in order to enable the coplanarity assumption. The mobile robot is equipped with a fourth device which actively requests the distance to the

[2]Available in http://www.youtube.com/watch?v=CcW2D4kN3E4.



(a) (b) (c) (d)

Fig. 3. Experimental results for the real dataset. (a) A snapshot of the environment where the data have been gathered, highlighting the location of the three UWB beacons. (c)–(e) The state of the filter at different time steps, with the beacons being labeled as #1, #2, and #3. Observe how all the beacons have been well-localized before the robot moves one meter from the beginning.

TABLE I

Distance	Ground truth	Estimation	Error (%)
$B1 \leftrightarrow B2$	9.913 m	10.282 m	3.6%
$B1 \leftrightarrow B3$	4.350 m	4.676 m	7.0%
$B2 \leftrightarrow B3$	6.346 m	6.520 m	2.7%

others in a timely fashion. Refer to Fig. 3(d) for a snapshot of the experimental setup. The range information from the UWB devices is synchronized to the robot odometry, which will be used as input to the probabilistic motion model in the RBPF.

In the first iteration of our method, the three beacons are initialized as samples distributed in a ring-like shape, as plotted in Fig. 3(e). It can be seen how these distributions quickly converge to the most likely positions of the beacons: after 45 iterations the uncertainty in the three beacons has been drastically reduced, as can be observed in Fig. 3(h). The evolution of the mean value of the 2D coordinates of each beacon and the associated uncertainties are also plotted in Fig. 3(a)–(c), respectively. Note that the maps represented in this section are always those ones associated to the particle with the highest weight in the RBPF, which in this experiment comprises of 100 samples.

These results show that the RBPF obtains an estimate of the beacon locations with the uncertainty decreasing as new measures are considered, but it should be also verified that the different distributions converge to the actual locations of the beacons. However, in practice it is difficult to obtain reliable measurements of the absolute beacon coordinates. Alternatively, we have measured the relative distances between the beacons, which are compared in Table I to the corresponding values for the final estimate of our method.

B. Simulated Data

We also present the results of our method for a simulated dataset with the purpose of demonstrating the ability of incorporating new beacons at arbitrary instants of time, and to show how the computational burden varies as the probabilistic representations of the beacons change within the filter.

For this experiment we have simulated range readings for 15 static beacons as the robot describes a circular path. A maximum detection range of 5 meters has been forced into the simulated sensor to allow the robot to discover new beacons as it moves. Measurements are also corrupted by a Gaussian noise with $\sigma_r = 0.03m$. The evolution of the filter is summarized in Fig. 4(a)–(d) at different time steps. In this case we can compute the absolute errors in the beacon localization, which are plotted in Fig. 4(e). All the beacons converge to a final error smaller than 0.1 meters.

We should remark how the beacon labeled as #10 takes a long time to converge since it comes out of the detection range of the robot, but it quickly converges after the robot approaches again (observe the abrupt decrease in the beacon error in Fig. 4(e) at step 60). As an example of a beacon added to the map some time after the beginning, consider the beacon #12, added at time step 26. This beacon still presents a clear ring-shaped distribution in Fig. 4(c) while most of the others have already converged.

The computation time required by each iteration of our method depends on how many beacons are observed and their representation in the RBPF particles: for the sample-based representation the time consumed is proportional to the number of samples, while for the Gaussian representation a fixed time is required to update the EKF. In the typical case of several samples in the auxiliary particle filters (e.g. more than 100), their update requires more time than in the case of the Gaussian representation. However, since the number of samples in the distribution of the beacons can decrease with new observations (recall section III-B), the time consumption smoothly decreases from a maximum (after inserting new beacons in the map) to the point when the auxiliary particle filter is replaced by an EKF. This pattern can be clearly observed in the graphs Fig. 4(e)–(f).

V. Conclusions

In this work we have analyzed the specific hurdles found in SLAM based solely on range measurements, as opposed to the more common case of range and bearing SLAM. We have maintained that a pure probabilistic, Bayesian solution is more desirable than other batch processing techniques due to its capability of consistently fusing information from different observations taking into account the associated uncertainty. A solution based on a RBPF has been proposed due to its ability for keeping the conditional distributions

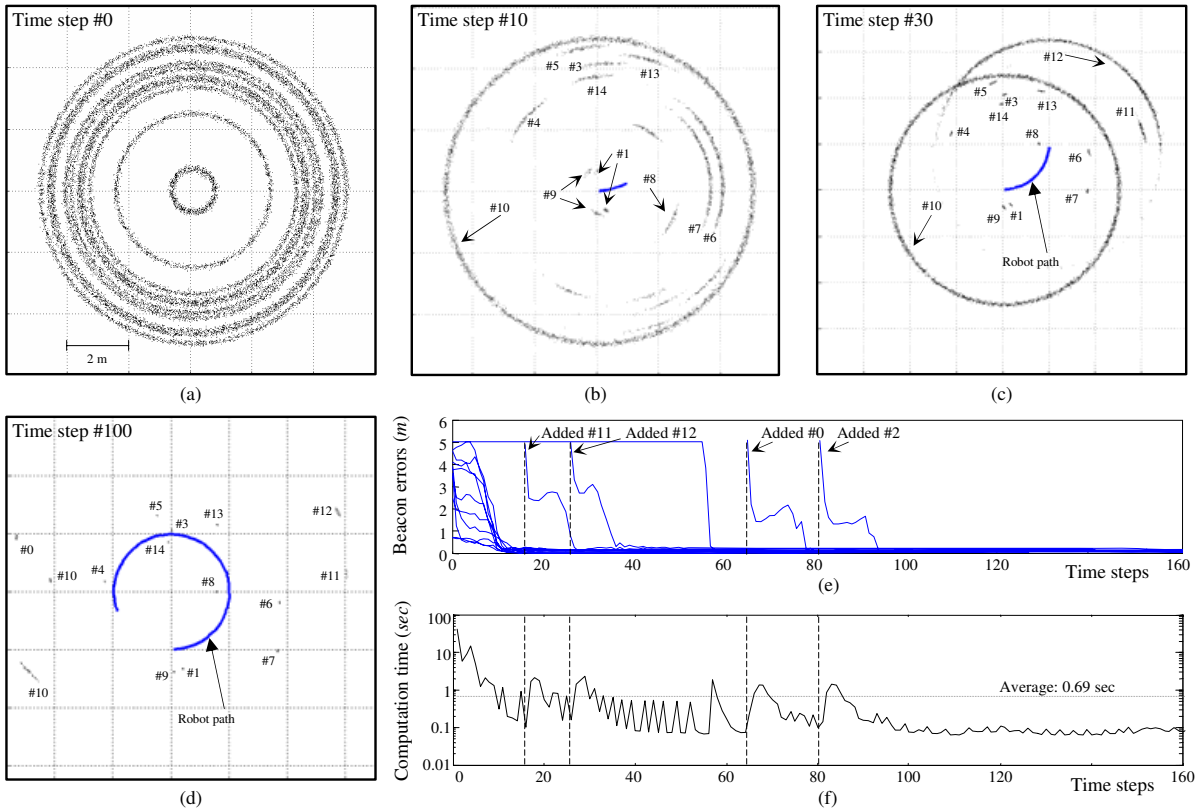


Fig. 4. Experimental results for the simulated dataset. (a)–(d) Four snapshots of the state of the RBPF at different time steps, where the beacons are labeled as #0,...#14. It can be appreciated the "ring" shape of the distribution for all the beacons initialized at the first iteration, which quickly converge towards the true beacon positions. (e) The errors from the mean estimate of each beacon and the ground truth. (f) The computation time at each time step.

separately, which is a great advantage in RO-SLAM since we can then initialize the distribution of the beacons as auxiliary PFs and convert them into EKFs when this becomes a better choice. As demonstrated with experimental results, our method has a high computational burden in the first iterations after the addition of new beacons, but it quickly becomes more efficient as the auxiliary particles are removed. As a result, we have an average execution time below 0.1s per iteration in some cases, rendering our method capable of online execution. Future research will address further improvements in efficiency and a more realistic management of outliers.

REFERENCES

[1] T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (SLAM): Part II-State of the art," *Robotics and Automation Magazine*, vol. 13, pp. 108–117, 2006.

[2] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[3] J. Djugash, S. Singh, G. Kantor, and W. Zhang, "Range-only SLAM for robots operating cooperatively with sensor networks," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006, pp. 2078–2084.

[4] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 176–183.

[5] J. Fernández-Madrigal, E. Cruz-Martin, J. Gonzalez, C. Galindo, and J. Blanco, "Application of UWB and GPS Technologies for Vehicle Localization in Combined Indoor-Outdoor Environments," *International Symposium onSignal Processing and Its Applications (ISSPA)*, 2007.

[6] G. Kantor and S. Singh, "Preliminary results in range-only localization and mapping," *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 2, 2002.

[7] D. Kurth, G. Kantor, and S. Singh, "Experimental results in range-only localization with radio," *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, 2003.

[8] J. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling," *Statistics and Computing*, vol. 6, no. 2, pp. 113–119, 1996.

[9] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," *Proceedings of the AAAI National Conference on Artificial Intelligence*, pp. 593–598, 2002.

[10] P. Newman and J. Leonard, "Pure range-only sub-sea SLAM," *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2, 2003.

[11] E. Olson, J. Leonard, and S. Teller, "Robust range-only beacon localization," *Autonomous Underwater Vehicles, 2004 IEEE/OES*, pp. 66–75, 2004.

[12] M. Pitt and N. Shephard, "Filtering Via Simulation: Auxiliary Particle Filters." *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–591, 1999.

[13] D. Rubin, "Using the SIR algorithm to simulate posterior distributions," *Bayesian Statistics*, vol. 3, pp. 395–402, 1988.

[14] S. Singh, G. Kantor, and D. Strelow, "Recent results in extensions to simultaneous localization and mapping," *International Symposium on Experimental Robotics*, 2002.

[15] S. Thrun, *Robotic Mapping: A Survey*. School of Computer Science, Carnegie Mellon University, 2002.