Using A-Priori Information to Improve the Accuracy of Indoor Dynamic Localization

Begümhan Turgut and Richard P. Martin
Department of Computer Science,
Rutgers University
110 Frelinghuysen Rd, Piscataway, NJ 08854
{bturgut, rmartin}@cs.rutgers.edu

ABSTRACT

We are considering the problem of dynamic localization of human targets in an indoor environment, such as an office building, where GPS signals are not receivable. Previous work has shown that static localization is possible through the measurement of the wireless signal strengths. Dynamic localization (tracking) can be achieved by performing periodic static localizations and filling in the gaps through an appropriate filtering technique. We are using a sampling-importance-resampling particle filter which is a probabilistic reasoning technique for this purpose.

In this paper we present approaches through which information about the environment (such as the floor plan of the building) and the target (such as the physical and social limitations of the human movement) can be incorporated in the prediction and weight update components of the particle filter. The particle filter requires information to be presented as conditional probability distributions, a format which presents both representational and computational efficiency challenges. In addition, the models need to consider both the a priori information and the operational details of the particle filter. Even technically correct models can reduce the accuracy of the localization, by inadvertently reducing the effective number of particles, which, in its turn, induces resampling errors. Through a series of experiments we show that the correct usage of a priori information can significantly improve the accuracy of dynamic localization.

Categories and Subject Descriptors

C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks

Keywords

Localization, tracking, particle filters, sampling

General Terms

Algorithms, Performance, Experimentation

1. INTRODUCTION

Dynamic localization (tracking) determines the trajectory of a target over a period of time. At its simplest, dynamic localization can be achieved using a discrete series of static localization steps. As the target is moving, its current location needs to be extrapolated from the latest localization, with the help of additional information including previous locations as well as knowledge about the environment and the target. Unfortunately, these pieces of information are frequently noisy or uncertain, and many of them are snapshots of dynamically changing phenomena. Particle filters, a probabilistic reasoning technique have been shown to be an efficient approach for integrating the various sources of information.

One area in which particle filters have been found useful is the self-localization of mobile robots. The approach proposed in this paper shares many common features with the approaches used in robotics. The most important difference is that the robots are localizing *themselves*. Although environmental factors, odometry errors and so on can introduce uncertainty in the location of the robot, there is no uncertainty about the intention of the robot, which forms the basis of the deterministic component of the prediction step of the particle filter. In our case, however, we do not know the intentions of the human targets. Although some assumptions can be made (for instance, of inertial movement), we simply do not know which direction will a human turn at an intersection, or whether it might decide to suddenly stop or turn around.

While the particle filter itself is a well tested theoretical tool, the practical expression of the available information in a format usable for the PF is a complex challenge, which led many researchers to advocate a "less is more" approach, claiming that by keeping the data model simple and the quantity of the processed information low they can achieve a better performance than complex models which try to exploit every available information piece.

In this paper, we describe an indoor dynamic localization system which utilizes a sampling-importance-resampling particle filter which integrates observations received from a static localization technique based on wireless signal strengths. We are using the GRAIL system [3] to measure the signal strength and perform the static localization using the algorithm based on minimum intersection areas of iso-RSS lines described in [21]. The advantage of the latter is that it provides both an estimate of location, as well as an estimate of the localization error, which allows us to convert the output into a probability distribution which represent external observations for the particle filter. Then, we describe ways in which a priori information about the environment can be integrated in the prediction and weight update components of the particle filter.

We are considering two types of a priori information: (a) environmental information and (b) target information. Environmental information includes the floor plan of the building, including walls, unreachable or forbidden areas. This information will be represented as a prior probability distribution (efficiently represented in a sum of rectangles probability model called the *ziggurat model*), and used in the weight update step of the particle filter. Target models contain information about the likely movement patterns of the target. In case of a human target this includes both physical constraints (humans have limitations on the speed and acceleration they can achieve) as well as social constraints (certain movement, although physically possible, are not likely to be performed on the corridors of an office building). Technically, this information should also be presented in form of posterior probabilities in the form $p(X_t|X_{t-1})$, where X_t is the location of the target at time t. In practice, however, we are describing this in the form of a *mobility model*, which accounts for the uncertainty of the location through a built in random component. As we will see, the nature of this randomness makes a significant difference in the performance of the overall system.

The remainder of the paper is organized as follows. Section 2 discusses the related work. Section 3 describes the general flow of the particle filter localization. The prediction model is described in Section 4 while the weight update step is described in Section 5. In Section 6, through an experimental study with various scenarios we compare the performance of particle filter localization with and without incorporating a priori information. Section 7 concludes the paper.

2. RELATED WORK

In the past decade, there has been an extensive research done both in indoor [2] and outdoor [11]. Most of the studies were emphasized based on its (a) applications such as location dependent services, resource allocation and control, and (b) the techniques used in positioning. As a result, various novel methods have emerged such as measurements of Time of Arrival (ToA) [11], Received Signal Strength (RSS) [2,20], Time Difference of Arrival (TDoA) [19] [13], Angle of Arrival (AoA) [8,16,18]. Other techniques use Distance Vectors (DV) [17] and Bayesian networks [14].

Particle filters [1, 4, 6, 7] is another well-known solution usually applied to dynamic localization. The main goal remains to be accuracy improvement. This approach has often been applied to indoor robot localization through data collection with visual cameras, as well as infrared, and ultrasound sensors of the moving target. From the literature, it is clear that there exists many ways of data collection methods, how to chose the subset of collected data to be fed into the particle filter method as well as their representation within the filter approach. In this section, we summarize several of these approaches.

Fox et al. [10] describe a Bayesian filter technique to estimate the location of multiple targets from wireless signal strengths in the presence of attenuation. This approach tracks each particle in the system as a separate hypothesis. The goal is to improve on approaches based on Kalman filters which are restricted to unimodal distributions. Experimental validation was performed in an indoor environment.

Gustafsson et al. [12] present a particle filter algorithm for static and dynamic localization in outdoor environments. The simulation studies of localizing and tracking cars and aircrafts show improved accuracy compared to existing Kalman filter techniques.

Montemerlo et al. [15] present a probabilistic particle filter method to track multiple human targets that are within a certain vicinity of a mobile robot equipped with 2D laser range finder in indoors. The conditional particle filter approach is able to handle different interpretation of sensor readings based on robot's poses over the predefined map. The experimental study shows that the algorithm is able to accurately localize a mobile robot and track multiple people simultaneously, even in the presence of global uncertainty in the robot poses.

Fox [9] describes a method in which it varies the number of particles in a particle filter used for localization. He uses the Kullback-Leibler distance (KLD) to decide on the minimum number of particles needed to achieve sufficient accuracy in a given moment. The application considered is mobile indoor robot localization. Experimental results show that the presented approach reduces the number of samples to 6% compared to fixed sampling and to 9% compared to likelihood sampling, while achieving higher accuracy.

Crisan and Doucet [4] gather a summary of several convergence results on particle filter methods. The authors show that with a set of specific restrictive assumptions, the distributions generated by particle filters converge to the accurate distribution.

Dellaert et al. [5] present a Monte Carlo (particle filter) method for indoor localization with a novel representation of the probability distribution of the target's state space. The approach concentrates the particles on the parts of the floor perceived as relevant. Experimental results using a robot with a laser sensor as a target show better accuracy and lower memory usage compared to grid-based approaches.

Vlassis et al. [23] apply to indoor localization an auxiliary particle filter previously proposed by Pitt and Shephard. They introduce an inverted nonparametric observation model based on nearest neighbor conditional density estimation. The experimental study uses a robot equipped with an omni directional vision system on multiple grid placements over the floor. This system feeds a set of images into a database to be used as a positioning input. The results show the algorithm to be robust and able to handle scenarios when the robot was manually relocated forcing it to re-localize based on the new sensor readings.

3. PARTICLE FILTER WORKFLOW

3.1 General Architecture

The general architecture of our system, together with the flow of control and flow of information is described in Figure 1.

The central part of our solution is the particle filter loop composed of four steps a) prediction, b) weight update, c) estimation and d) resample and restart. We will discuss them one by one.

Prediction. The prediction step applies a mobility model consisting of a deterministic and a random component to all the particles in the particle cloud. The mobility model

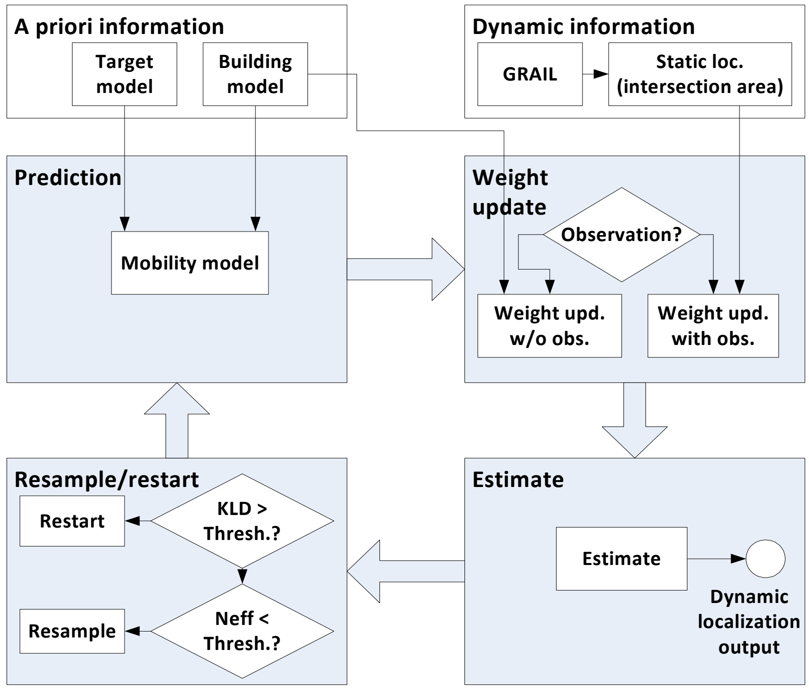


Figure 1: The workflow of the dynamic localization

depends on the a priori information about the target, and the particle history. However, as we will see in Section 4, the overall performance of the particle filter can be improved if we integrate environmental information in the prediction step as well.

Weight update. The weight update step is different depending on whether a new external observation is available or not. If a new observation is available, the corresponding probability is used to change the weight of the particles:

$$w_p^{'} = w_p \cdot prob(p|obs) \tag{1}$$

If no new observation is available, the weight update is done based on the a priori probability from the environmental information. In both cases, the resulting weights are normalized such that they sum to 1.0 for all particles.

Estimation. The estimation step determines the output of the particle filter. There are several choices we can use here - the simplest one involves a weighted sum of the particles. Various robust estimation techniques can be applied as well.

Beyond extracting the output of the localization, the estimate is also used internally in the particle filter to update the deterministic component of the prediction model. If the prediction is inertial, as in our case, we need a list of previous locations to calculate the inertial speed. These locations are extracted from the global estimate.

Resample or restart. During the operation of the particle filter, a natural occurrence is that the weight of certain particles becomes very small or zero. These particles will not contribute to the estimate of the location, thus they are excluded from the prediction and weight update step as well. Over time, the effective number of particles are reduced, and the performance of the particle filter suffers.

The resampling step replaces the existing particles with a new set which has equal weights. The resampling algorithm is designed such that the probability distribution implied by the new particle set approximates the one implied by the old one. The resampling technique we are using is multinomial resampling.

Unfortunately, resampling introduces its own errors, thus it needs to be deployed as rarely as possible. In our current system we resample the particles, when the effective number of particles becomes less than $\frac{2}{3}$ of the original number of particles.

The last step of the particle filter loop concerns the merging of the particle set. Under certain conditions, it is advantageous to replace the particles with a new set of particles.

The first such situation is the restart decision we mentioned above. If we notice that there is a major, irreconcilable difference between the current observation and the particle set, and we trust the observation more than our particles, we might decide to discard all the particles and replace them with a new set of particles created from the sampling of the probability distribution obtained from the most recent observation. Details about this technique are available in [22].

3.2 Comparison versus the particle filters used in robotics

Particle filter based techniques have been frequently used in localization for mobile robot applications. Our system has many common points with those systems. The major difference is that when a robot performs self-localization, the deterministic component of the prediction is based on the known intentions of the robot. The uncertainties are only in the random factors, such as slippery floors, odometry errors, imperfections in the actuators and so on.

When localizing a human target, however, we do not have access to his or her intentions, thus the deterministic component will also have uncertainties. We can, for instance, assume an inertial motion in long corridors, however, in the intersections between corridors we have at best a statistics based guess in which direction will the target turn. Human targets might also surprise us by suddenly stopping or turning around. In conclusion, our model needs to operate under a much larger uncertainty in its prediction model, and we need to handle the occasional wrong guesses. One approach towards this direction is the automatic restart method we proposed in [22]. This approach bases the restart decision on the threshold of the Kullback-Leibler divergence between the probability surface induced by the particle cloud and the probability surface corresponding to the latest observation.

4. PREDICTION STEP: MOBILITY MODELS

The mobility models of the particles describe the way in which the particles are evolving in time. The mobility model is a more convenient way to express the probability distribution of the location of the target conditional of the previous location $P(x_t|x_{t-1})$. Thus the mobility model needs to include the uncertainty which otherwise would be present in the probability distribution.

Thus the mobility model can be seen as having two components: a purposeful component which reflects the purposeful movement of the target, and a random component which reflects the uncertainties in the execution of the movement, observation errors and uncertainties in the modeling of the persistent components.

4.1 The deterministic component: inertial estimation

In particle filter based localization systems used in mobile robotics, the deterministic component of the movement is simply the desired movement pattern of the mobile robot. In our case, we do not have access to the desired movement pattern of the human observer, thus we need to estimate it based on the limitations of the target.

One of the relatively reliable approaches is based on the inertial estimate: we assume that the target, in absence of new decisions, will move on an inertial trajectory, maintaining its current speed of movement. The decisions we need to make for the calculation of the inertial speed concern the time span over which the decisions are made and the source of locations we are considering. Over the source of the considered locations we have essentially three choices: we can consider the locations of the particles, the locations of the observation and the estimated location over the whole particle filter.

Using the historical estimate over the particles historical location would allow each particle to make a different inertial prediction. However, the problem is that this way inertial prediction will be completely independent from the observations. Remember that the observations do not affect the location of the particle, only its weight. Essentially, such a system would only introduce an inertial movement over the random component of the prediction model.

Using the location implied by the observations would make the deterministic component of the mobility model identical over the set of particles (which has performance advantages). A problem, however, is the quality of these observations. If we have frequent observations such that the space traversed by the target between two observations is smaller than the sum of localization errors, we can have a situation where the inferred inertial movement vector is pointing exactly in the opposite direction. We can, naturally improve on this by performing a filtering over the history of observations. However, this second filter, for speed values, would be in addition to the existing, location based filter which is our main goal. The third choice, which we used in our system, uses the already filtered location values of the current particle filter to perform the inertial estimate. That is, the velocity estimate will be based on the output of the particle filter and will be computed using the following formula.

$$v^{'} = \alpha v + (1-\alpha)[L_{est}(t) - L_{est}(t-1)] \tag{2}$$

where $t-1$ is the value of the previous observation.

One of the problems with the inertial estimate is that in an indoor environment inertial movement is limited by obstacles, the shapes of the corridors etc. In a corridor turn, the inertial estimate is actually the worst choice, because it will throw all particles outside the building (which the case according to our floor map).

Naturally, a high level prediction model might be able to account for traffic models on the corridors, human decision models etc. We can, however, do a simple modification, which resolves most of these problems: if the predicted future location of the estimate is unfeasible, restore the inertial estimate to zero and reset the fast start algorithm.

The formula 2 perform an exponential smoothing over the estimates over a time. Unfortunately, the disadvantage of exponential smoothing is that it starts up slowly at the beginning of the simulation, or after a visualized reset. If the localization process is visualized, this phenomenon can be clearly seen as the estimate falls behind the observation at the beginning of the target, then catches up, only to fall behind again when the target changes direction in the corners.

To solve this problem, we introduce a slightly modified formula by setting:

$$\alpha^{'} = \min(\alpha, 1/n) \tag{3}$$

where n is the number of observations since the inertial speed estimate was restarted.

Gaussian dispersion

The random component of the mobility model accounts for the observation errors, uncertainties and imperfections in the mobility of the target, as well as uncertainties in the deterministic component of the movement model.

The random component is important because while the persistent component is responsible for the movement of the weight center of the posterior probability surface, the random component is responsible for its actual shape.

The simplest choice for the random component, frequently used in particle filters, is Gaussian dispersion. The new position of the particle is determined by the deterministic speed and Gaussian noise added to the x and y coordinates:

$$x^{'} = x + v_{det} \cdot \Delta t \tag{4}$$

$$x(t+\Delta t) = x(t) + v_x \Delta t + \varepsilon_x \tag{5}$$

$$y(t+\Delta t) = y(t) + v_y \Delta t + \varepsilon_y \tag{6}$$

where ε_x and ε_y are drawn from a normal distribution $N(0, \Gamma)$.

The problem with the Gaussian dispersion model is that while it is an accurate representation of observation error, it is typically not a good representation of uncertainties and the imperfections in the mobility of the targets. Gaussian noise is historyless, it assumes that the divergence from the deterministic component can change in every iteration of the particle filter. Figure 2-a traces the movement of five independent particles using the Gaussian dispersion over the span of 1000 time steps of 1ms. The "jagged" trajectories are characteristic of the model.

Our knowledge about the nature of the target (a moving human), tells us that this model is unrealistic. A human target, or a robot with an equivalent weight can not physically enact the accelerations necessary for this trajectory. Note that this model would be feasible for the tracing of the red dot of a laser pointer.

Another problem is that the Gaussian model is dependent on the update rate of the particle filter. The trace in the figure was obtained with a relatively fine tracing rate of 1ms. Using a larger, 100ms sampling rate, we would obtain an equally jagged, but significantly different trajectory in which the small "shake-like" movements are replaced by large "jumps". Neither of these are realistic models of human targets moving in an office environment.

4.2 Random wandering

Let us now try to develop a model which can generate trajectories which can be interpreted as a realistic hypothesis of the movement of a human target. To achieve such a model, we need to satisfy several conditions. We need to keep the movement of the individual particles within the bounds of the physical possibilities of a target with the weight of a human. This implies that the random component of the movement needs to be also subject to the laws of inertia. We also prefer a model which does not give significantly different trajectories at different sampling rates.

The proposed *random wandering* model assumes that the deterministic component of the partical movement is summed with a $\bar{v}_{random}$ random component, which is carried over between timesteps, but is gradually changing through two separate, small acceleration noise components: the *angle noise* ε_{angle} and the *magnitude noise* $\varepsilon_{magnitude}$. The separation of these two components is justified by the fact that the heading and the propulsion system is relatively well separated both in humans and in autonomous robots.

In conclusion, the random component of the prediction is described by the following formulas:

$$v_r(t + \Delta t) = v_r(t) + \varepsilon_\Theta \tag{7}$$

$$v_\Theta(t + \Delta t) = v_r(\Theta) + \varepsilon_\Theta \tag{8}$$

$$v_x = v_r + \sin(v_\Theta) \tag{9}$$

$$v_y = v_r + \cos(v_\Theta) \tag{10}$$

Figure 2-b traces the movement of five independent particles using the random wandering movement over the span of 1000 time steps of 1ms. We note that the model generates trajectories which are physically feasible for the human target.

4.3 Exploiting a priori information in mobility models: random wandering with sliding

One of the problems with the mobility models is that they do not take into account the a priori information about the environment, such as the map of the building. In our current setup, the building information is taken into account in the weight update step of the particle filter. The mobility model generates movement patterns without considering the priori probability of the destination location, thus particles might end up traversing walls, moving into forbidden rooms and so on. In the weight update step the weights of these particles will be adjusted accordingly.

Ignoring the a priori information at the prediction step, however, can create problems even if it is subsequently considered at the weight update step. Particles which had an unfeasible prediction, will have their weights reduced to zero or near zero. These particles will be *lost*, eliminated in the resampling step. While they do not introduce errors by themselves, their potential contribution to the accuracy of the filter will be lost. In addition, every resampling step introduces resampling errors.

A more subtle effect of keeping a priori information only in the weight update step is that it introduces an *adverse selection* over the particles. For instance, when moving in narrow corridors, fast particles are more likely to "stray" into forbidden areas. These particles will be lost, and the diversity of the particle population reduced.

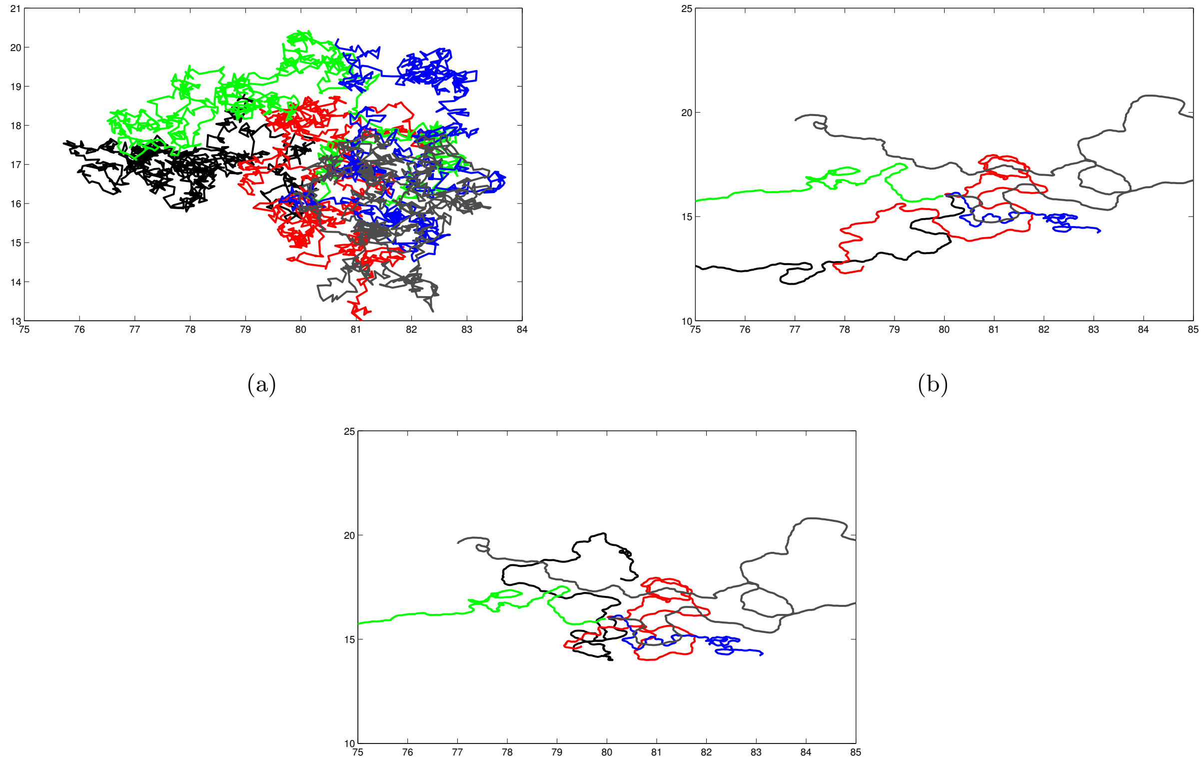
It is desirable to adjust the mobility model in such a way as to avoid generating particles moving into unfeasible locations. In our particular case, the main problem relates to the map of the building: how should a particle behave if the mobility model generates a prediction which requires the particle to cross a wall?

There are several immediate possible solutions. The particle can stop, that is, predict the next position as the current position. The particle can bounce back from the wall in an elastic collision. These model are physically feasible (as long as they do not involve large accelerations), but they are socially unfeasible: human targets normally do not bounce back from walls, or stop in the middle of movement. The alternative approach we propose is *sliding* - if a predicted movement would make the target cross a wall, instead it would loose the movement along the axis perpendicular to the wall, and retain the movement along the axis parallel to the wall. This corresponds to the socially feasible movement pattern of a target avoiding an obstacle, but continuing along the length of the corridor at a path closer to the wall.

Figure 2-c traces the movement of five independent particles using the random wandering with sliding, with the wall structure used being the map of the CoRE building at Rutgers University, see Figure 4. The trace is very similar to trace (b), for the simple random wandering approach, as the same random generators were used with identical seeds. One of the five particles was about to enter the forbidden area, its movement was modified with the sliding approach.

5. THE WEIGHT UPDATE STEP

In the weight update step, the particle filter updates the weight of the particles, based on external information of their likelihood. If an external observation is available at the given time point, the update step uses it together with any



(a) (b)

(c)

Figure 2: The traces of five particles over 1000 timesteps of 1ms each using (a) Gaussian diffusion (b) Random wandering and (c) Random wandering with sliding

available a prior information. If no observation is available, the update is based exclusively on the a priori information.

The main challenge of the weight update step is the representation of the observations. The canonical representation of the observation for the particle filter is a $P(X|E)$ conditional probability, describing the probability that the target is the location X given observation E. For many static localization algorithms, however, the output is only a single "most likely" location X_{ML}.

Converting this output into a conditional probability involves some sensitive choices. Representing as a single, "stick" probability is a bad choice which would eliminate all or all-but-one particle. A better choice would be the assumption of a bivariate Gaussian distribution centered in X_{ML}. Gaussian distributions are frequently used to represent probability surfaces, because of *(a)* the theoretical reason that many real world error distributions are actually good approximations, and *(b)* the pragmatic reason, that this allows us to represent the probability surface with a small number of parameters, in the bivariate case the median point *M(x,y)* and the *2x2* covariance matrix.

In our case, however, the Gaussian representation is not as advantageous, as the a priori information contains hard transitions from feasible to unfeasible areas, which are difficult to represent in the form of Gaussians or mixture of Gaussians. We choose to use a computationally efficient representation which can represent probabilities with sharp transitions accurately, and can approximate smooth transitions to an arbitrary accuracy. This representation, the *ziggurat model*, represents the probability distribution as a series of rectangular areas with specific probabilities. The probabilities multiplied with the area of the rectangles are summing to 1.0. Thus, the probability at a given point is equal with the sum of probabilities of the rectangles which contain the given point.

The a priori location probability of a target is the probability, independent of any measurement, to be at a specific location at a specific moment in time. We can represent it as a 2D probability surface $p_{\text{apriori}}(target, t, x, y)$. An example of the feasible and unfeasible locations for the 2nd floor of the CoRE building is shown in Figure 3. This representation, determined from the building's floor plan, assumes that the feasible locations are only the corridors (that is, all the labs and offices are closed).

6. SIMULATION STUDY

For this study, we have considered a scenario involving a person moving at a normal walking pace around the corridors of the CoRE building at Rutgers University. Static localization using the intersection area algorithm is performed 24 times during this experiment. The results are fed into the particle filter workflow, which then provides the results at a much higher temporal resolution.

This scenario is a relatively good match for the particle

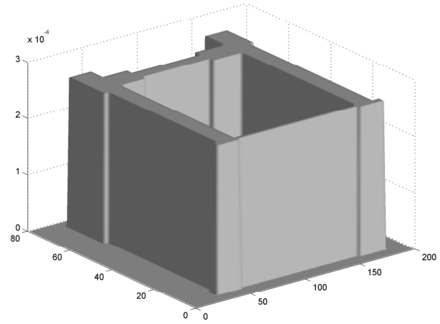


Figure 3: The a priori probabilities for possible and impossible locations on the 2nd floor of the CoRE building, using the ziggurat probability model.

filter model: the movement is relatively predictable, and the narrowness of the corridors used as a priori information constraints the possible errors. There are however three 90 ° turns in the trajectory where the inertial prediction of the movement would fail. These, together with the starting point are the cases where the particles might make a bad prediction stranding the particle filter. Whether this will indeed happen depends on the actual series of observations, their errors and precision. The particle filter can usually handle relatively large errors in the longer straight sequences, but one or more large errors in the turns might make the particle filter "miss the turn".

The original data was collected from a real time experiment using the GRAIL system. However, in order to have a larger dataset, we have generated artificial scenarios, in which we have generated the error of the static localization from a distribution which matches the cumulative distribution function of the actual static localization.

6.1 Prediction and weight update models considered. Visual observation.

For our experiments, we can considered four combinations of prediction models and weight update models, with various levels of use of a priori information. In the following, we succinctly describe these combinations. For a better understanding of the differences between these models, we have taken screenshots of the distribution of the particles at timepoint t=8000ms in the movement of the target (see Figure 4). The visualization represents the relative weight of the particles with their colors, thus particles represented by white rectangles have zero or near zero weight, while particles represented by black rectangles have weights equal or higher than $1/n$, where n is the number of particles.

For the deterministic component, all the models are using the inertial model described in Section 4.1.

Gaussian diffusion with no a priori information. In this case the prediction model is based on the Gaussian diffusion model described in Subsection 4.1. Neither the prediction nor the weight update step uses any a priori information about the building or the target. The update step only uses information from the observations (the static localization steps). The screenshot in Figure 4-a shows that the particles are distributed over a very large area, although the actual estimate is reasonably close to the correct ground truth.

Random wandering without a priori information. In this setup, the prediction model uses the random wandering model described in Subsection 4.2. Again, no a priori information was used in the prediction or the weight update step. The static snapshot of this setup in Figure 4-b shows relatively little difference from the Gaussian diffusion setup, although from Figure 2 we know that the *dynamic* behavior is radically different.

Random wandering with a priori information. In this setup we are using a priori information in the weight update step, but not in the prediction step. Figure 4-c shows a very different picture from the previous ones: particles are clustered more tightly around the estimate. Still, a relatively large number of particles are outside the feasible area, with very low or zero weight, these are the particles which moved outside the feasible area since the last resampling. In addition, we have a relatively large number of particles with zero weight inside the feasible area, these being mostly particles whose trajectory took them out of the feasible area, and then back again.

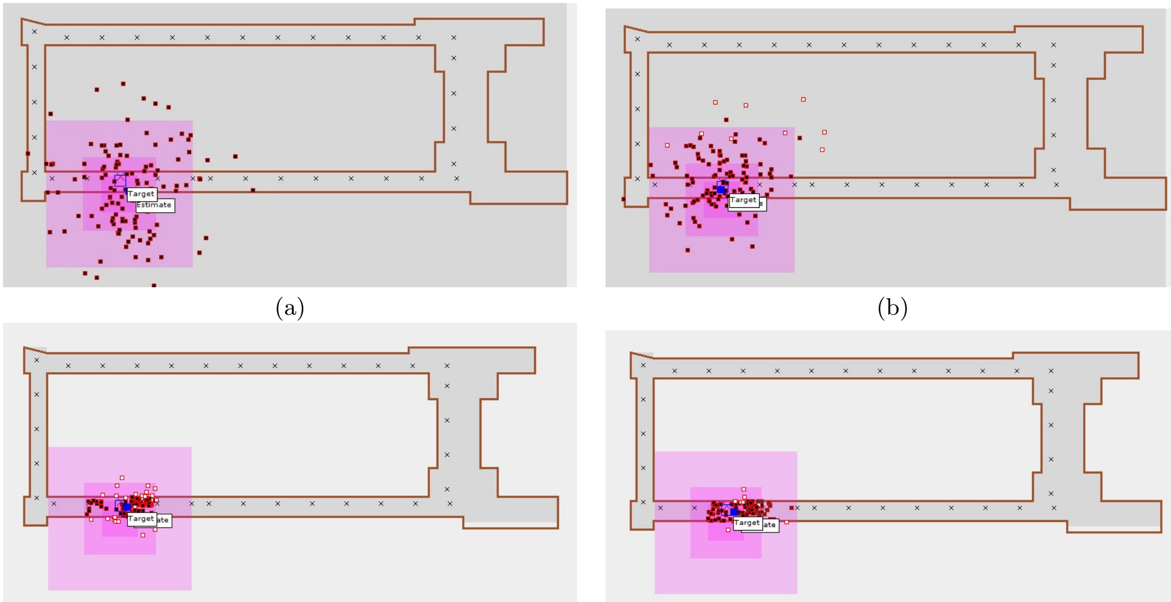
Random wandering with sliding. This setup uses a priori information both in the prediction model (by implementing the sliding mechanism described in Subsection 4.3, and in the weight update step. Naturally, the sliding method eliminates most cases of particles entering the unfeasible area (although such events can still happen, for instance if a particle heads straight to the wall, or in corners). A screenshot of this setup is shown in Figure 4-d. As expected, we see a more tight packing of the particles in the feasible area (the corridor), only a small number of particles straying outside.

6.2 Experimental results

We have run the localization workflow over the 100 scenarios generated as described in the previous section. We used $n = 140$ particles. The deterministic component was a smoothed inertial predictor based on the running average of estimated speed, with the weight of the new observation being $w = 0.2$. The stochastic component was based on a diffusion model, which assumes a level of inertia in the randomness of the movement. We used a diffusion speed of 0.03 m/s, with an angle noise $\epsilon_{angle} = 0.05$ and speed noise $\epsilon_{speed} = 0.05$.

Figure 5 shows the average distance from the ground truth of the estimated location function of the number of particles for the four different prediction models. To put the performance of the particle filter in perspective, we draw the average accuracy of the extrapolated static localization (8.75ft in our experiments) as a thick grey line. If a particle filter setup does not improve on this baseline, its deployment is not justified.

As an overall assessment, all the four methods show a decrease in the localization error with the increase in the number of particles. The methods without a priori information are performing the worst. The random wandering method performs somewhat better performance for a low number of particles, but for larger number of particles the two approaches converge to the same values. The relative improvement in the Gaussian diffusion approach happens because the "jagged" movement of the particles becomes smoothed out if summed over a sufficiently large number of particles.



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

Figure 4: Screenshots of the distributions of the particles at the same time point t = 8000ms. (a) Gaussian diffusion (b) Random wandering without a priori information (c) Random wandering with a priori information and (d) Random wandering with a priori information and slide

Overall, however, the results show only a minor increase in accuracy over the extrapolation of the static localization, an increase which does not justify the computational cost of the particle filter. Note, though, that in our scenario we have assumed a very dense arrival time of static localization observations - when the observations are move rare, the improvements will be more significant.

The remaining two approaches, random wandering with a priori and sliding respectively, form a separate group, both of them achieving an average accuracy of about 5.5 ft, significantly better than the 8.75 ft accuracy of the extrapolated static localization. The difference between these two approaches is on the speed of the convergence: while the slide approach achieves this accuracy with about 30 particles, the approach not utilizing sliding requires 60 particles, with the associated memory and computational cost.

The reasons behind this is explained by the Figure 6 which shows the number of particles which were “lost” through the resampling. The sliding approach shows a smaller particle loss compared to random wandering with a priori information. This was expected, because a every sliding particle would have otherwise crossed into the unfeasible areas, and become lost. The smaller number of lost particles allows the sliding approach to achieve its best accuracy with a lower number of particles. However, lowering the number of lost particles can not be a goal in itself: the approaches without a priori information are having a much lower number of particle loss, because they only loose particles through the incompatibility with observations. Nevertheless, the smaller number of lost particles does not translate in higher accuracy.

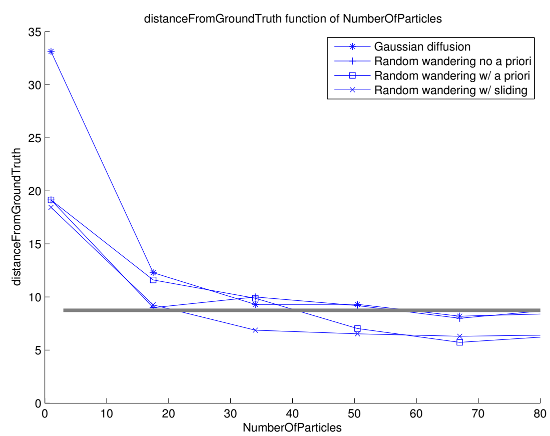


Figure 5: The average distance from ground truth of the estimated location function of the number of particles for the four considered setups. The horizontal gray line shows the accuracy of the extrapolated static localization.

7. CONCLUSIONS

In this paper we described ways in which a particle filter based localization approach can use a priori information about the environment and the target. We found that the use of a priori information during the weight update step can significantly increase the accuracy of the localization. Furthermore, we found that the use of the a priori information in the prediction step as well can reduce the number

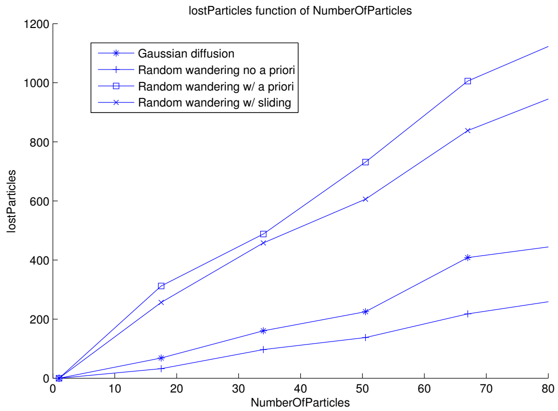


Figure 6: The number of lost particles function of the number of particles for the four considered setups.

of particles necessary to achieve a given level of accuracy, thus reducing the computational and memory requirements of the localization.

8. REFERENCES

[1] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

[2] P. Bahl and V. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proceedings of IEEE InfoCom*, pages 775–784, March 2000.

[3] Y. Chen, G. Chandrasekaran, E. Elnahrawy, J.-A. Francisco, K. Kleisouris, X. Li, R. P. Martin, R. S. Moore, and B. Turgut. GRAIL: A general purpose localization system. *Sensor Review, special edition, Localization Systems*, 28:115–124, 2008.

[4] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50:736–746, 2002.

[5] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1322–1328, 1999.

[6] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer Verlag, 2001.

[7] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[8] E. Elnahrawy, J.-A. Francisco, and R. P. Martin. Bayesian localization in wireless networks using angle of arrival. In *Proceedings of ACM SenSys*, pages 272–273, November 2005.

[9] D. Fox. Adapting the sample size in particle filters through KLD-sampling. *I. J. Robotic Res.*, 22(12):985–1004, 2003.

[10] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, 2:24–33, 2003.

[11] I. Getting. The global positioning system. *IEEE Spectrum*, 30(12):36–47, December 1993.

[12] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50:425–437, 2002.

[13] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Proceedings of 5th Annual International Conference on Mobile Computing and Networking*, pages 59–68, August 1999.

[14] D. Madigan, E. Elnahrawy, R. Martin, W.-H. Ju, P. Krishnan, and A. Krishnakumar. Bayesian indoor positioning systems. In *Proceedings of Infocom*, volume 2, pages 1217–1227, March 2005.

[15] M. Montemerlo, S. Thrun, and W. Whittaker. Conditional particle filters for simultaneous mobile robot localization and people-tracking. In *IEEE Conference on Robotics and Automation*, volume 1, pages 695–701, May 2002.

[16] D. Niculescu and B. Nath. Ad hoc positioning system (APS). In *Proceedings of Globecom*, pages 2926–2931, November 2001.

[17] D. Niculescu and B. Nath. DV based positioning in ad hoc networks. In *Proceedings of Telecommunication Systems*, pages 267–280, 2003.

[18] D. Niculescu and B. Nath. VOR base stations for indoor 802.11 positioning. In *Proceedings of the 10th Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 2926–2931, September 2004.

[19] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The CRICKET location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 32–43, August 2000.

[20] S. Saha, K. Chaudhuri, D. Sanghi, and P. Bhagwat. Location determination of a mobile device using IEEE 802.11 access point signals. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, volume 3, pages 1987–1992, March 2003.

[21] B. Turgut and R. P. Martin. Localization for indoor wireless networks using minimum intersection areas of iso-RSS lines. In *Proceedings of the IEEE Conference on Local Computer Networks (LCN-07)*, pages 962–972, 2007.

[22] B. Turgut and R. P. Martin. Restarting particle filters: an approach to improve the performance of dynamic indoor localization. In *to be presented at IEEE Global Communications Conference (Globecom)*, 2009.

[23] N. Vlassis, B. Terwijn, and B. Krose. Auxiliary particle filter robot localization from high-dimensional sensor observations. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 7–12, May 2002.