Improving Unreliable Mobile GIS with Swarm-based Particle Filters

Fatma Hrizi, Jérôme Härri, Christian Bonnet
EURECOM*, Mobile Communications Department
Campus SophiaTech, 450 Route des Chappes
Biot, France
{hrizi, haerri, bonnet}@eurecom.fr

ABSTRACT

Accurate Mobile Geographic Information System (GIS) is a major building block of many applications, particularly in Intelligent Transportation Systems (ITS). In this context, GPS provides position information of each vehicle, while immediate surrounding information is gathered through the exchange of beacons. Yet, the ITS environment is characterized by frequent losses of GPS signal and beacons. Estimation/tracking based on Kalman or Particle filters could be an alternative to support the precision of the Mobile GIS, but both approaches are equally sensitive to missing and unreliable data. In this paper, we propose GSF, a Glowworm Swarm Optimization to particle filters, adding the bio-inspired capabilities of Glowworms to converge to multiple potential estimates, when unreliable mobile GIS lack precise updates. We first analyze the performance of GSF by considering perfect conditions. Second by considering GPS signal loss, packet loss and positioning errors. Simulation results show that our approach achieves its design goal of improving the precision of the mobile GIS. GSF performs better than standard particle filter scheme in terms of position accuracy, and this at a reduced complexity and fair convergence time.

Categories and Subject Descriptors: H.2.8 [Information Systems]: Database Management: Database applications: *Spatial databases and GIS*;
G.3 [Mathematics of Computing]: Probability and Statistics: *Probabilistic algorithms (including Monte Carlo)*;
C.2.1 [Computer-Communication Networks]: *Network Architecture and Design- Wireless Communication*;
I.6.6 [Simulation and Modeling]: *Simulation Output Analysis*.

Keywords: Mobile GIS, GPS, Particle Filter, Swarm Intelligence, ITS.

*EURECOM industrial members: BMW Group, Cisco, Monaco Telecom, Orange, SAP, SFR, STEricsson, Swisscom, Symantec, Thales. The authors would further acknowledge the support of the French DGCIS/OSEO for the FOT Project SCOREF.

ACM SIGSPATIAL MobiGIS '12, November 6, 2012, Redondo Beach, CA, USA.

1. INTRODUCTION

Mobile Geographical Information System (GIS) is an emerging technology combining GIS and the capabilities of mobile communication to access, retrieve, fusion and alter remotely stored data. Geographic data can be obtained not only by means of traditional GIS methods such as acquisition from GPS, digital road maps and local databases, but also via data coming from wireless technologies. This first allows remote access to large size databases, but also enables data collection and storage from mobile GIS applications to remote databases.

In ITS, GPS data and beacon messages represent the major sources of geographic information for the mobile GIS. GPS provides local position information of each vehicle, while geographic information of the surrounding environment (called commonly *"awareness"*) is collected from the beacons exchanged between nodes. A high GIS precision is strongly required by many applications, notably ITS traffic safety. For instance, in case of collision avoidance application, accurate geographic information of each vehicle and their neighbors has to be provided by the mobile GIS in order to efficiently evaluate the risk of collision with potential vehicles. Yet, GPS signals and wireless communication are known to be unreliable and uncertain. Beacon losses due to high fading and interferences may occur frequently, and GPS signals may be missed or received with large errors. In such cases, extrapolating unreliable or missing data using position tracking represents a solution for the Mobile GIS.

Tracking has been studied extensively in the last decades and several tracking approaches have been proposed. Bayesian filters i.e. Kalman filters[2] and particle filters [8] are the most known ones. The limitations in these approaches are that they rely on reliable and constant position updates either from GPS or from beacons, as well as on assumptions that future motions not varying much from the previous ones. Yet, the unreliable characteristics of the wireless channel and GPS signals, as well as unexpected sudden motion changes typically found in vehicular motions, make those filters lose the precise location of estimated vehicles. Observing that vehicular mobility is jointly governed by physical and social laws, e.g. clusters are formed on the road as a result of social needs and behaviors, and depicts comparable patterns to swarm behaviors, we suggest to rely on artificial swarm intelligence to enhance tracking algorithms.

In this work, we propose glow-worm swarm filter (GSF), a swarm-based SIR particle filtering based on multiple hypotheses tracking. The proposed solution is to consider not only a single potential future location but also to consider various other potential locations modeling the eventual loss of GPS signal or packets in addition to the unpredictable motion changes. A glow-worm swarm optimization (GSO) algorithm has been used because of its capabilities to find multiple local optima and to cluster the search space into various multiple hypotheses. This might implicitly improve the functionality of particle filter by augmenting the diversity of particles and avoiding the degeneracy problem. Our approach can be applied in several GIS domain but in this paper we investigate the case study of ITS. We evaluate the performance of GSF with a basic SIR particle filter (later referred as PF) scheme in terms of precision and convergence speed. Using the iTETRIS [1] simulation platform and calibrated realistic vehicular scenarios, we illustrate how GSF provides better tracking accuracy requiring a lower number of particles compared to the standard PF. Moreover, GSF showed to achieve its design goal to ensure a trade-off between high tracking precision and fast convergence.

The rest of this paper is structured as follows. Section 2 analyses the tracking problem of unreliable GIS: the standard particle filter is discussed with its challenging problems then we introduce our tracking approach GSF. Afterwards, in Section 3, a simulation study is performed evaluating the performance of GSF compared to the basic PF. In Section 4, we provide related works of position tracking. Finally, Section 5 reports the conclusions and provides directions for further research.

2. TRACKING UNRELIABLE GIS

2.1 Assumptions

In the scope of this work, we assume that each node (later referred as ego node) manages and maintains a GIS including the self location information gathered from Global Positioning System (GPS). GPS has become one of the most important data resources of GIS. It provides 3-dimensional position information [x, y, z], velocity vector [Vx, Vy] and time information in real time. In ITS and related cooperative applications, an up-to-date knowledge of the surrounding context is required as well. Therefore, each node handles a GIS for other neighboring nodes. This is obtained by the exchange of periodic beacons, i.e. broadcast messages sent at fixed or variable rate over a single hop, and including location and mobility data e.g. geographic position, speed and direction.

2.2 Problem statement

Accurate location information is becoming increasingly important for many applications. In particular, ITS cooperative safety applications require precise and up-to-date GIS, i.e. provided by local GPS data. Moreover, an accurate context-aware knowledge, i.e. GIS of neighboring nodes, is needed. These two kind of information are unfortunately exposed to frequent loss of precision. In some specific zones in the road, the GPS reception might be obstructed leading to a wrong estimation of the self state. Moreover, due to the high fading of wireless channel in the vehicular environment, beacon messages sent from neighboring nodes are highly influenced by channel losses. These unreliable characteristics suggest position tracking to support and to enhance the precision of the GIS.

Various estimators have been proposed over the past years. Bayesian filters i.e. Kalman filters[2] and particle filters [8] are the most popular ones. The common idea is to create a posterior distribution of the inner motion state considering all collected data i.e. from GPS or beacons. Kalman Filters have been widely used for tracking in several research fields. Their key drawback comes from their implicitly assumption that vehicular motions follow a linear model, and that collected data is subject to Gaussian noise. Extended Kalman filters (EKF) represent a possible alternative for the estimation of nonlinear systems, although the posterior distribution remains approximated by a Gaussian distribution. When dealing with the challenging environment of a mobile GIS, both Gaussian approximations and motion linearizations are not accurate assumptions and might yield to low tracking performance. Particle filters instead do not make any assumptions in the motion or the posterior distributions, and have been shown to fit well in GIS. They approximate the posterior distribution by a set of weighted particles corresponding to potential estimates of the unknown state.

Conceptually speaking, in a Bayesian framework, the tracking problem is composed of three main blocks:

1. *Internal Motion Model* - the internal representation of the evolution of the position. It is described by a mobility model, which is supposed to match closely the movement of the target vehicles. Mathematically, it is represented as $p(x_t|x_{t-1})$
2. *Estimation Model* - the estimation of the internal model from observations (gathered from GPS or beacon messages). Particle filters could be employed to solve the likelihood function $p(z_t|x_t)$.
3. *Decision Making Model* - the output of the tracking. The posterior probability conditioned to the observations obtained by the decision model. Mathematically, it is represented as $p(x_t|z_t, z_{t-1}...z_0)$, and solved by functions such as Maximum Mean Square Error (MMSE) or Root Mean Square Error (RMSE).

The effectiveness of position tracking depends extremely on two basic elements. First, the accessibility of the external observation (coming from beacons) is required for the good performance of the tracking system. When a GPS signal is missing or when packet exchanged between nodes are lost due to bad channel conditions, a sudden large deviation is expected to occur. Mathematically, if z_t are missed for several t values, then the likelihood function and the posterior function cannot be evaluated properly. Second, the reliability of the internal motion model. If it is known, predefined and well controlled (as the case of most of the Bayesian tracking approaches), the tracking problem is not complex. Unfortunately, highly dynamic environment is characterized by frequent and sudden changes in dynamic patterns. Accordingly, the evolution of the position $p(z_t|x_t)$ deviates significantly from the real state. Both aspects are regularly experienced in mobile GIS for ITS environments. In this work, our main focus is to improve the accuracy of the GIS and provide an efficient estimation of the state of the ego vehicle and other moving neighboring nodes under such circumstances. We address in details the first major problem consisting of missing or erroneous GPS signals, or losing packets exchanged between nodes due to a bad channel conditions. We discuss partially as well the second issue of motion model which is subject to sudden and unexpected movement changes.

2.3 Particle Filtering

Particle filters approximate the posterior distribution, describing the state of the system, by a finite set of weighted samples (or *particles*) $\{(x_t^{(i)}, w_t^{(i)})$: i=1,..,n$\}$. Recursive Monte Carlo sampling is performed guided by a dynamic motion model. Several implementations have been proposed for Particle Filtering. Their main differences lie in the used resampling method and/or importance distribution. Sampling importance resampling (SIR) PF [8] is the most used implementation in tracking systems. In order to avoid the problem of degeneracy of the PF algorithm where all but one of the importance weights are close to zero, resampling is used at each time step. The posterior distribution is approximated by the importance weights $w_t^{(i)}$ as follows:

$$\int f(x_t)p(x_t|y_{0:t})dx_t \approx \sum_{i=1}^{n} w_t^{(i)} f(x_t^{(i)})$$

$$\sum_{i=1}^{n} w_t^{(i)} = 1$$

The weight update is given by:

$$w_t^{(i)} = w_{t-1}^{(i)} \times (\frac{p(y_t|x_t)p(x_t|x_{t-1})}{\pi(x_t|x_{0:t-1}, y_{1:t})})$$

where the importance distribution $\pi(x_t|x_{0:t-1}, y_{1:t})$ is approximated as $p(x_t|x_{t-1})$.

Algorithm 1 gives an overview of the SIR particle filter operation. Mainly, SIR consists in three steps, the state prediction where the posterior probability is given based on the probabilistic system transition model $p(x_t|x_{t-1})$. The second step is the update which is performed based on the likelihood $p(z_t|z_{t-1}^{(i)})$. The particles are then resampled to generate an unweighted particle set. Resampling is performed by drawing n particles from the current set with probabilities proportional to their weights and then assigning to them equal weights $1/n$.

Algorithm 1 Pseudo-code of the SIR particle filter

1: Initialization:
   Draw N particles $x_t^{(i)} \sim p_0(x_t|x_{t-1})$ with equal weights
2: State prediction:
   $x_t^{(i)} \sim p(x_t|x_{t-1})$
3: Weights update:
   For each particle: Calculate $w_t^{(i)} = p(z_t|x_t^{(i)})$
4: Normalization:
   For each particle: Calculate $w_t^{(i)} = w_t^{(i)} \ / \ \sum_{i=1}^{n}(w_t^{(i)})$
5: Resampling

Figure 1 depicts a detailed graphical representation of the evolution of particles of the SIR algorithm with only $n = 9$ samples. First, particles are initialized or distributed according to the probability distribution $p(x_t|x_{t-1})$. At the reception of a new observation, weights are computed for the different samples according to the likelihood of the observation given the current state $p(z_t|x_t^{(i)})$. Normalization and resampling are then performed to generate new particles according to their weights with more particles in areas with high weights and fewer particles in areas with low weights. The state transition generates new particles states given the current states and the algorithm is restarted.

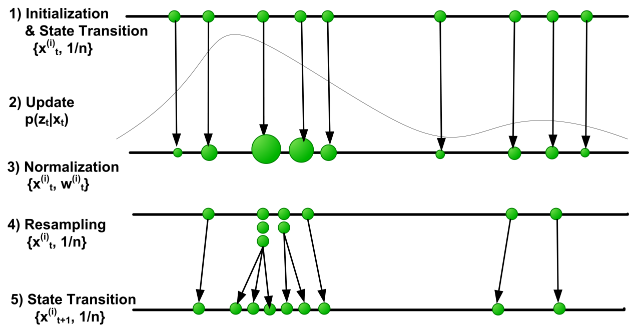

Figure 1: A representation of the evolution of the particles following the SIR algorithm.

A weak point of particle filters is particles degeneracy. Even with a high number of particles, it may happen that the set of particles loses its diversity and deviates from the real state. This is emphasized by the lack of external observations due to GPS signals or packets losses and/or to the unexpected and sudden change in motion patterns, as mentioned in Section 2.2. As depicted in Figure 1, in such situations, the regions of likelihood are far away from the prior distribution. Thus, most of the particles are generated with low weights. The reason behind this degeneracy problem is that PF maintains one single hypothesis which is strongly connected to the motion model's assumptions. An example of a vehicular scenario is illustrated in Figure 2, based on this unique hypothesis, PF fails to track accurately the neighboring vehicle after an update loss. The particles lose the diversity yielding to severe divergence from the real position.

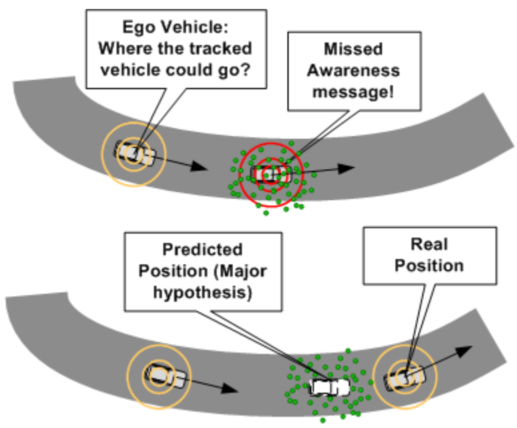

Figure 2: An illustration of a tracking scenario where the dots illustrate the generated PF particles. The tracked vehicle is estimated by the PF of the ego vehicle to be in the major hypothesis. After missing the beacon, PF fails to predict properly the real position.

The proposed solution we describe next section is not only to consider a single potential future location (illustrated as *Major Hypothesis*) but also to consider various other eventual locations due to the loss of GPS signals or observations (illustrated as *Minor Hypothesis*), and/or to unpredictable motion changes.

2.4 Glow-worm Swarm Filter (GSF)

Following observed human behaviors, when we lack clues on where something disappeared, we look in various *"potential"* locations as function of the context. Our filtering proposal follows the same approach. Consequently, typical situations resulting from GPS signals or packets losses, or sudden change in mobility are modeled as alternative hypotheses apart from the expected single major hypothesis. Maintaining multiple hypotheses allows the filter to handle sudden changes and recover from temporary diversions. The idea is to cluster the set of particles into sub-groups representing each a potential hypothesis. Therefore, we extend the SIR PF with the bio-inspired pattern, namely the Glow-worm Swarm Optimization algorithm (GSO) [5]. Before the resampling phase, the GSO algorithm is applied. Accordingly, the PF particles are mapped onto the glow-worms of the GSO. The algorithm is able to divide the particles into clusters that can converge simultaneously to multiple optima. The set of particles is enriched with not only high but also low probable hypotheses. The glow-worms behavior applies well to the idea of having multiple tracking hypotheses.

Figure 3 reproduces the same scenario of Figure 2 but this time GSF is applied, particles are sub-divided into 3 groups, one major hypothesis and two minor hypotheses. Considering these different hypotheses, the filter, consequently, is able to manage the loss of the beacon and maintain efficiently the tracking process stable.

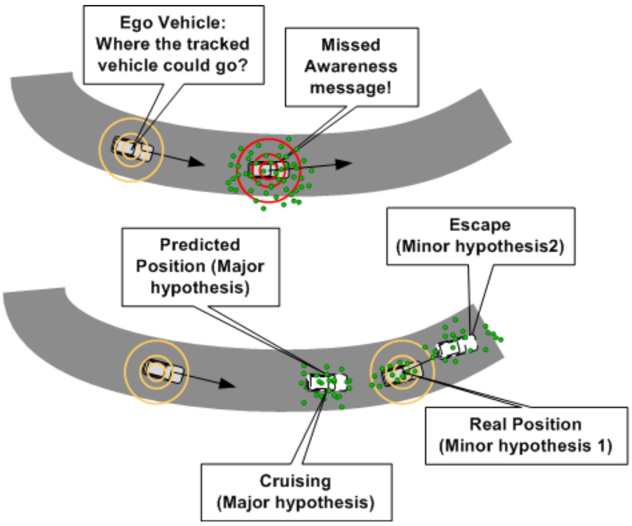

Figure 3: An illustration of a tracking scenario. The dots in the figure represent the particles corresponding to the position estimate of the vehicle. The tracked vehicle is estimated by GSF to be in the minor hypothesis. Considering unexpected beacons losses, GSF is able to ensure good tracking performance.

2.4.1 *Glow-worm swarm optimization (GSO)*

In this section, we give more details about the GSO algorithm. Basically, Glow-worm swarm optimization (GSO) [5] is a swarm intelligence based algorithm inspired by the behavior of the glow-worms in nature where the female glows to attract a male for mating. In the algorithm, a probabilistic approach is used to select neighbors with brighter glow. A luciferin value proportional to the intensity of the glow is associated with each agent. GSO starts with a random deployment of an initial population of glow-worms in the search space with equal quantity of luciferin l_0 and with the same neighborhood decision r_0. Each iteration consists of a luciferin update phase followed by a movement phase based on a transition rule. The luciferin value is updated based on previous luciferin value and a function $J(x)$ that evaluates the fitness of the previous position of the glow-worms. The particularity of GSO is its decentralized decision making aspect. The movement phase depends on the position of neighbor where the glow-worm will move. The algorithm of GSO is further explained in Algorithm 2.

Algorithm 2 Pseudo-code of the GSO

Parameters: n, l_0, r_0, γ, ρ, β, s, r_s, n_t
Deploy N glow-worms randomly with equal luciferin $l_i(0) = l_0$
while t<itermax **do**
  for $i = 1 \rightarrow N$ **do**
    Update phase
    $l_i(t) = (1-\rho)l_i(t-1) + \gamma\ J(x_i(t))$
  end for
  for $i = 1 \rightarrow N$ **do**
    Movement phase
    Determine the list of neighbors for glow-worm i
    for $j = 1 \rightarrow N$ **do**
      Calculate moving probability
    end for
    Select Glow-worm j according to the probability: $P_{ij} = (l_j(t) - l_i(t)) / (\sum_{k=1}^{n}(l_k(t) - l_i(t)))$
    Move Glow-worm i toward j: $x_i(t+1) = x_i(t) + s*((x_j(t) - x_i(t)) / (||x_j(t) - x_i(t)||))$
    Update neighborhood range: $r_d^i(t+1) = \min(rs, \max(0, r_d^i(t)+\beta(n_t - |N_i(t)|)))$
  end for
end while

2.4.2 *GSF algorithm*

In GSF, as shown in Algorithm 3, GSO is applied to particles before the resampling phase. Particles, considered as glow-worms, move towards neighbors with high weights. This results in a creation of several sub-groups with various weights in the solution space. The particles resampling is then performed for each sub-group. Only particles with highest weights in each sub-group will survive.

Figure 4 depicts a representation of an example of the evolution of particles according to GSF algorithm. The GSO algorithm is performed before resampling phase. Particles move towards neighbors with high weights. Accordingly, different groups are built corresponding to multiple hypotheses. The resampling process is then applied to each group with more particles in areas with high weights and less particles in areas with low weights. The state transition predicts new particle states given the current particle states and the algorithm is restarted.

3. EVALUATIONS

In this section, we evaluate the performance of our new tracking system aiming at enhancing the GIS accuracy. We attempt to examine the effectiveness of our GSF algorithm to track efficiently the state of both the ego vehicle and the moving neighboring nodes considering the two cases with

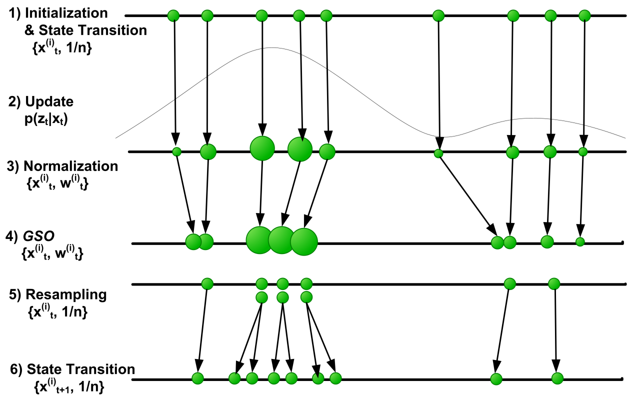

Figure 4: A representation of the evolution of particles according to GSF algorithm.

Algorithm 3 Pseudo-code of the GSF

Initialization: Deploy particles randomly
State update: Apply mobility model
Weights update
Normalization
Move particles according to GSO algorithm
Resampling taking into account the several sub-groups created by GSO

and without GPS (or packets) losses and under dynamic constraints. Moreover, GPS positioning errors resulting from the precision of the navigation system are considered.

We perform a comparison of GSF with the standard particle filter scheme. We consider vehicular environment as a case study to model dynamic systems. In the following, we introduce the simulation setup and the configuration of mobility and network scenarios. We present then the set of performance metrics we have measured, and finally the results of our experiments.

3.1 Simulation setup

We have carried out a set of simulations to examine the performance of our proposed system under various realistic conditions. We have used iTETRIS [1], the integrated ITS simulation platform. iTETRIS enables the simulation of V2X communications and the modeling of vehicular mobility patterns and traffic conditions.

Simulations have been carried out on the basis of one ego vehicle that runs GSF and the basic PF to compare both filters performances. Accordingly, simulations of GPS signals and beacons losses are performed using the same scenario. In the first case, by considering the scenario as self tracking of the ego vehicle. In the second case, by running the same algorithm of the neighboring node on the ego node. To simulate the loss, we proceed by suppressing the beacons transmissions in some given time steps. We apply three different schemes of beacon update loss (or GPS signal loss): one loss out of ten updates per second, two successive losses and three successive losses.

3.1.1 *Mobility scenarios*

We have considered both urban and highway traffic environments. The former models the unpredictable and the uncertain mobility due to the brutal change that can occur in the vehicle trajectory in such environment. The latter represents the high dynamic aspect of vehicular environment. The simulation experiments are based on four scenarios. We have used two calibrated realistic scenarios from the city of "Bologna". Furthermore, two artificial scenarios: urban and highway have been designed. The first realistic scenario called "Acosta Pasubio joined" models an urban environment composed of multiple intersections. The second one, consists in a highway. The Figures 5 and 6 illustrate the "Acosta Pasubio joined" and the highway iTETRIS scenarios. The configuration parameters of our simulations are shown in Table 1.

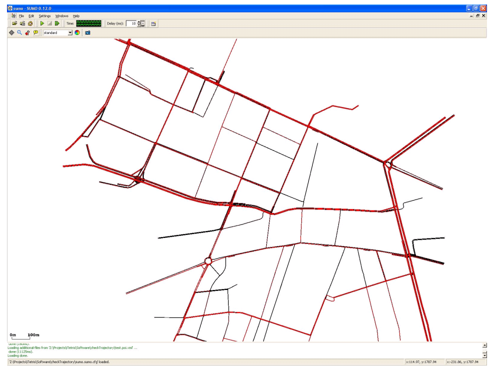

Figure 5: iTETRIS Acosta Pasubio joined mobility scenarios.

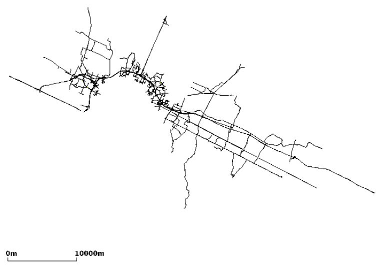

Figure 6: iTETRIS Highway mobility scenarios.

Scenario	Size	Mean Speed [m/s]	Max Speed [m/s]	Max Accel [m/s2]	Max Deccel [m/s2]
Artificial-Urban	x:2000m y:100m	12.2	13.5	1	4.5
Artificial-Highway	x:10000m y:0m	18.47	36	1	4.5
iTETRIS-Urban	x:2126m y:2117m	8.01	13.89	3	4.5
iTETRIS-Highway	x:69000m y:53000m	11.31	36.11	3	4.5

Table 1: Configuration parameters of the mobility scenarios.

3.1.2 *Network scenario*

In our communication scenario, we consider that beacons are sent at the network layer periodically. Table 2 gives an

Parameter	Value
802.11p Channel	CCH Control channel
Simulation Time	100s for each run
Number of simulation runs	5 times for each scenario
Propagation model	Logarithmic Distance
Transmission power	20 dBm
V2V transmission range	400m

Table 2: Configuration parameters of the network scenario.

overview of the configuration parameters for the communication scenario.

Regarding the configuration of the GSO algorithm, the parameters' values resulting from [5] have been used. They are illustrated in Table 3.

ρ	γ	β	n_t	s	l_0
0.4	0.6	0.08	5	0.03	5

Table 3: Configuration parameters of the GSO algorithm.

3.1.3 Performance metrics

Performances of GSF and the basic particle filter have been examined in terms of:

1. Error distance (ED) which is defined as the Cartesian distance between the estimate obtained from filtering algorithms and the real position: ED $= \sqrt{(D_x^2) + (D_y^2)}$ where D_x^2 and D_y^2 denote respectively the error on X and Y position. This metric represents a measure of the level of accuracy of the filters.

2. Convergence time that denotes the filter run-time reflecting its real-time capability and its convergence speed.

3.2 GSF performance evaluation

In this section, we aim to evaluate the performance of our GSF algorithm and compare it with the generic particle filter scheme considering several traffic environments. Mainly, we examine in this section the level of accuracy of the filters in urban and highway scenarios. As a first step, we assess the performance of both filters in loss free channel. Then, we consider beacon messages or GPS signals losses with various ratios. Finally, the impact of positioning errors on both tracking schemes is studied. Moreover, the results of the convergence time for both schemes are represented. Error distance and convergence time are considered as the evaluation metrics.

3.2.1 Tracking Precision

Urban Scenario.

Table 4 shows the average error distance obtained for both GSF and the basic PF in case of both realistic iTETRIS and artificial urban scenarios. We deduce that the performance of both filters is enhanced when increasing the number of particles. For urban traffic scenarios, GSF gives better estimation results less than 1.6 m. However, the basic particle filter exceeds 2 m of position error. An improvement on the tracking error (54% than the basic PF) is provided by GSF in case of urban traffic. A slight decrease in the distance error is observed for realistic iTETRIS scenario which can be explained by the fact that the average speed is more important for artificial scenario. We conclude that the speed is an influencing factor on the tracking model which will be more shown in next section.

Scenario	10	100	500
GSF(A-Urban)	1.53 m	1.49 m	1.55 m
PF(A-Urban)	2.73 m	2.44 m	2.24 m
GSF(iTETRIS-Urban)	1.36 m	1.31 m	1.27 m
PF(iTETRIS-Urban)	2.42 m	1.79 m	1.69 m

Table 4: Error distance of GSF and the basic PF in case of urban scenarios for different particles' numbers.

Scenario	10	100	500
GSF(A-Highway)	2.10 m	2.11 m	2.10 m
PF(A-Highway)	2.68 m	2.52 m	2.19 m
GSF(iTETRIS-Highway)	1.41 m	1.39 m	1.35 m
PF(iTETRIS-Highway)	2.14 m	1.58 m	1.60 m

Table 5: Error distance of GSF and the basic PF in case of highway scenarios.

Highway Scenario.

The first observation that can be taken from Table 5 is that the distance error increases when the velocity of the vehicle becomes important. As expected, when the number of particles grows, the accuracy of both filters is enhanced. Moreover, GSF outperforms the standard particle filter scheme which even with 500 particles can not perform the same way as GSF with 10 particles. GSF is capable of enhancing the tracking error of 12% only with 10 particles compared to 500 particles for PF.

3.2.2 Impact of packet/GPS loss

ITS environment is constrained to high fading channels and congested network which lead to a serious problem of packet loss. Moreover, it may be common in ITS environment to miss GPS signals. The impact of this aspect on tracking algorithms is worthy to investigate. In the following, we study this aspect in both urban and highway traffic environments.

Urban Scenario.

From Table 6, we observe that in all the cases the distance error from real position information grows when the packet loss ratio increases. The distance error does not exceed around 2.5 m for GSF scheme however it goes up to 4.7 m in case of the basic PF. The particles in the basic PF lose their importance. However, in GSF they are spread in all possible directions to augment the space search.

Highway Scenario.

Table 7 shows the behavior of both tracking algorithms in high speed environments. From the first sight, we can deduce that for both filters the distance error increases com-

# Packet Tx	1/1	1/2	1/3	1/4
GSF	1.53 m	1.69 m	1.88 m	2.50 m
PF	2.73 m	3.23 m	4.26 m	4.72 m

Table 6: Impact of packet loss on the error distance of GSF and the basic PF in case of urban scenario.

# Packet Tx	1/1	1/2	1/3	1/4
GSF	2.10 m	2.82 m	3.68 m	4.64 m
PF	2.68 m	3.99 m	4.94 m	5.78 m

Table 7: Impact of packet loss on the error distance of GSF and the basic PF for highway traffic scenario.

pared to urban scenario. Moreover, when increasing the update loss ratio the error becomes more and more important.

Figure 7 depicts the evolution of the distance error in urban scenario during simulation time. Both results with and without updates loss (first scheme) are plotted. We can observe that the distance error increases in case of update loss for the basic PF and the GSF. Moreover, we distinguish two zones in the curve, the former where both filters perform approximately the same way. The latter is the zone where the basic particle filter deviates from the real position and gives more than 2m of position error which is due to the augmentation of vehicle speed. In contrast to particle filter, the performance of GSF remains almost stable when varying the vehicle velocity. This is due to the intelligence in our tracking scheme and its reliability to provide the highest position accuracy. Regarding highway scenario illustrated in Figure 8, three zones can be defined. The first corresponds to an equivalent performance for both filters. It is worthy to note that the performance of our tracking model can be shown in the second zone where GSF outperforms PF and the last one, corresponding to high velocity, where both filters deviate from the real position.

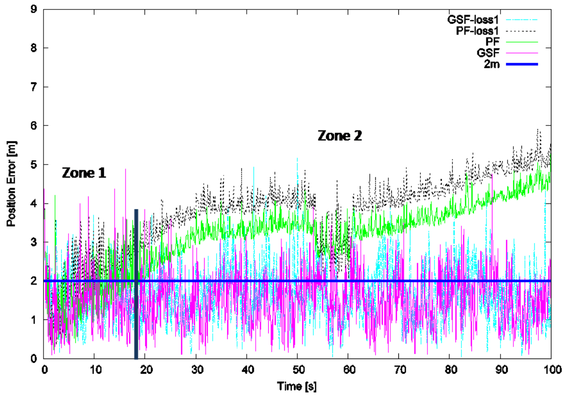

Figure 7: The evolution of the Error distance of GSF and PF with packet loss ratio (1/1) in case of urban scenario.

3.2.3 *Impact of sudden changes in motion model*

Sudden and unexpected movement changes influences dramatically the performance of the tracking scheme. It is the subject of investigation of this section. Figure 9 depicts a

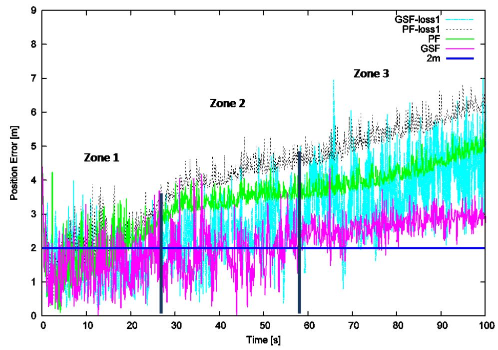

Figure 8: The evolution of the Error distance of GSF and PF with packet loss ratio (1/1) in case of highway scenario.

tracking case taken from iTETRIS Acosta scenario where the vehicle starts moving from second 32. At seconds 40 and 42 the position of the vehicle deviates abruptly due to two successive lane changes. This leads to a severe deviation of the particle filter where the position error reaches 4m. However, GSF succeed to track well the vehicle giving an average position error of around 1.3m.

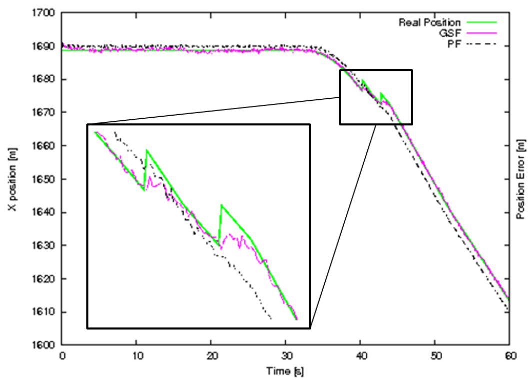

Figure 9: Tracking error of GSF and PF in case of lane change.

3.2.4 *Effect of positioning errors on tracking performance*

A part from GPS signal loss, another source of error more related to the precision of the navigation system can be introduced to the positioning information. In this section, we examine this aspect and we evaluate the behavior of GSF and the basic SIR PF when we consider the error of positioning devices that can be introduced to the real position. Table 8 summarizes the obtained results for all of the different traffic scenarios. We can conclude that the performance of GSF remains stable however a degradation of PF performance can be observed. For instance the position error goes from 2.14m to 3.26 in case of realistic highway scenario.

The obtained simulation results reveal that GSF achieves its design goal of providing a good and sufficient level of accuracy for position as compared to the basic particle filter scheme. GPS signals and awareness updates losses as well as errors on positioning have shown to be important and influencing factors for the precision of filters. In the next

Scenario	PF	GSF
Artificial-Urban	3.08 m	1.58 m
Artificial-Highway	3.18 m	2.08 m
iTETRIS-Urban	2.31 m	1.38 m
iTETRIS-Highway	3.26 m	1.44 m

Table 8: Impact of positioning error on the error distance for PF and GSF and considering 10 particles.

Scenario	10	100	500
GSF(A-Highway)	1.6 s	40.7 s	1715.4 s
PF(A-Highway)	0.6 s	8.2 s	158.7 s
GSF(A-Urban)	1.3 s	33.7 s	934.6 s
PF(A-Urban)	0.6 s	8.2 s	158.2 s

Table 9: Convergence time of GSF compared to the basic PF.

section, we study the performance of the filters in terms of convergence time.

3.2.5 Convergence time

In order to evaluate the real-time performance of the tracking algorithms, the execution time has been measured for different numbers of particles. Table 9 illustrates the real execution time in seconds of 100s of simulation in ns-3 for some scenarios. The basic PF ensures the lowest run time compared to GSF for the different scenarios which is due to the extra computation that GSF algorithm introduces. However, in order to respect real-time requirements of ITS active safety applications and at the same time preserve a high level of accuracy, a trade-off between fast convergence and high precision must be taken into account. The performance of GSF with 10 particles showed to ensure this trade-off.

4. RELATED WORKS

Tracking has been extensively studied in many research domains. Several mechanisms have been proposed to enhance the accuracy of the state estimation. The most relevant techniques of mobility prediction that we found in the literature can be classified in two categories: Bayesian filters such as Kalman [2] and particle filter [8][4], and heuristic approaches based on neural networks [3][7] and genetic algorithms [9]. Generally, the issue with heuristic schemes is the memory usage. For instance, genetic algorithm is characterized by a long processing time and there is no guarantee for convergence. The performance of neural networks depends on the learning phase which is inappropriate for dynamic environments.

Kalman filter is an optimal Bayesian filtering approach whose advantage is its simplicity in implementation and computation. Nevertheless, it has been designed mostly to suit linear and Gaussian problems considering simplistic assumptions. This does not cope with the dynamic nature of ITS systems. Particle filters, on the other hand, consider non-linear and non-Gaussian systems but also does not require very complex computation resources. In spite of these advantages, particle filters have the drawback of particles degeneracy. Even with a high number of particles, it may happen that the set of particles loses its diversity and deviates from the real state. Many optimization algorithms have been proposed to tackle this limitation. For instance, the authors in [6] propose to introduce genetic algorithm in the basic particle filter approach. As mentioned above, this may reduce the convergence rate due to the heavy computation time that genetic algorithms require.

5. CONCLUSIONS AND FUTURE WORKS

In this work, we proposed to improve mobile Geographic Information Systems (GIS) for uncertain and unreliable environments by integrating an swarm-inspired tracking mechanism. We have presented GSF (Glow-worm Swarm Filter), an improved particle filter designed to deal with unreliable wireless channel resulting in beacon messages and GPS signal loss on the one hand, and with the unpredictable internal dynamic models on the other hand. The main feature of GSF is the glow-worm abilities to track multiple hypotheses (i.e. potential location estimates), typically required when the previously described challenges add a large unknown on the location estimates. Simulation results show that GSF outperforms the basic particle filter and ensures a good trade-of between high precision in position estimate and fast convergence. In future works, we plan to improve our tracking system and particularly the decision making model. We believe that it could be the key to further reduce the error to fall below that of the GPS precision requirements in ITS (1.5m).

6. REFERENCES

[1] iTETRIS project. `http://wiki.ict-itetris.eu/index.php/Main_Page`.

[2] B. D. O. Anderson and J. B. Moore. Optimal filtering. *Prentice-Hall, Englewood Cliffs, NJ.*

[3] C. M. H. C. Y. Kong and M. Y. Mashor. Radar tracking system using neural networks. *in Proc. of the International journal of the computer, the internet and management, Vol. 6, No. 2.*, 1998.

[4] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. *in Proc. of the IEEE Transactions on in Signal Processing*, 2002.

[5] K. Krishnanand and D. Ghose. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *in Proc. of the Swarm Intelligence Journal*, 2009.

[6] S. Park, J. Hwang, K. Rou, and E. Kim. A New Particle Filter Inspired by Biological Evolution: Genetic Filter. *World Academy of Science, Engineering and Technology*, September 2007.

[7] L. I. Perlovsky and R. W. Deming. Neural networks for improved tracking. *in Proc. of the IEEE transactions on neural networks, Vol. 18, No. 6.*, 2007.

[8] B. Ristic, S. Arulampalam, and N. Gordon. Beyond the kalman filter: Particle filters for tracking applications. *Artech House Boston London*, 2004.

[9] P. J. Shea, K. Alexander, and J. Peterson. Group tracking using genetic algorithms. *in Proc. of the International Society Information Fusion*, 2003.