14th International Conference on Information Fusion
Chicago, Illinois, USA, July 5-8, 2011

Robust regularized particle filter for terrain navigation

Achille Murangira, Christian Musso, Karim Dahia, Jean-Michel Allard
Onera - The French Aerospace Lab
F-91761 Palaiseau, France
achille.murangira@onera.fr, musso@onera.fr, karim.dahia@onera.fr, jean-michel.allard@onera.fr

***Abstract*—In this paper, we address the multimodality of the filtering distribution in the context of ambiguous measurements. In terrain navigation, the similarities in the elevation profiles can lead to multiple modes in the posterior distribution. Typical experiments based on standard particle filters show that the filter tends to lose the true mode as time goes on. Using a clustering algorithm, a new filter based on a mixture representation with local regularization is introduced and displays good robustness properties.**

Keywords: Particle filtering, Mixture filters, Multimodality, Regularized particle filters, Terrain navigation, Clustering.

I. Introduction

Particle filters have recently proved to be the most efficient solution to non-linear/non gaussian filtering in a wide range of applications. They were introduced in a Bayesian setting where recursive computation of the approximate posterior distribution is made tractable once a dynamic markovian model and a likelihood model are defined.

Besides its ability to deal with non linearities in both the dynamics and the observation model, particle filtering is suited to multimodality in the posterior distribution. However, it has been observed that usual particle filters are not always successful at keeping more than one mode in the posterior distribution [1]. This can be critical in applications where the measurements are ambiguous for a long stretch. Another issue, specific to the standard regularized particle filter is that the resampling procedure often fails when the modes in the filtering distribution are well separated.

In this paper we propose to adapt mixture particle filters to the problem of terrain navigation. After underlining the relationship between mode loss and resampling in section II, we recall the basics of mixture filters in section III and show how it can be applied to regularized particle filters [6]. Simulations show a gain in robustness especially when the resampling rate is high.

II. Resampling and loss of multimodality

Consider the following popular state/space model, analyzed in particular in [2]:

$$\begin{cases} X_k = & X_{k-1} + W_k \\ Y_k = & X_k^2 + V_k \end{cases} \tag{1}$$

where $W_k \sim \mathcal{N}(0, \sigma_w^2)$ and $V_k \sim \mathcal{N}(0, \sigma_v^2)$. The observation's ambiguity leads to a multimodal filtering distribution with symmetric modes around zero. In [2], it was observed that the classical particle filter loses track of one of the modes after a few iterations.

To get an insight into why resampling can lead to a loss of mode, we introduce a simple probabilistic experiment. We evidence mathematically that repeated resampling leads to a a loss of mode in a multimodal setting. Let $\{X_1^0, \cdots, X_N^0\}$ be a sample set generated according to a bimodal distribution with symmetric modes around zero. In practice, X_i^0 can be drawn from a gaussian mixture $\rho\mathcal{N}(-1, 0.1) + (1-\rho)\mathcal{N}(1, 0.1)$, where $0 < \rho < 1$. A simple classification rule is to assign X_i^0 to the first mode if $X_i^0 < 0$ and to the second one otherwise.

Now by drawing N samples uniformly with replacement from the initial set, we obtain a new collection $\{X_1^1, \cdots, X_N^1\}$. Denote $\{X_1^k, \cdots, X_N^k\}$ the sample set at time k drawn from $\{X_1^{k-1}, \cdots, X_N^{k-1}\}$ according to the same procedure. This is analogous to the Wright-Fisher model [13] widely studied in population genetics. Let $A = (A_k)_{k \geq 0}$ be the number of X_k^i's belonging to the first mode. $A_k \in \{0, 1, \cdots, N\}$. it is easy to see that A_k is a markov chain with binomial transition probability $\mathcal{B}(N, \frac{A_{k-1}}{N})$ ie.:

$$\mathbb{P}(A_k = j | A_{k-1} = m) = \binom{N}{j} \left(\frac{m}{N}\right)^j \left(1 - \frac{m}{N}\right)^{N-j} = p_{mj} \tag{2}$$

The states $\{0, N\}$ are said to be absorbent since once the Markov chain A reaches either one it remains attached to that state. Note that because A is also a bounded martingale it converges a.s to X_∞. Since 0 and N are the only absorbing states, $X_\infty \in \{0, N\}$. This ensures that the time of absorption $\tau = \inf\{n | A_n \in \{0, N\}\}$ is almost surely finite. Let $t_i = \mathbb{E}_i[\tau] = \mathbb{E}[\tau | A_0 = i]$ be the mean time of mode loss. The Markov property yields the following equation [13]:

$$1 + \textstyle\sum_{j=0}^{N} p_{ij} t_j = t_i \quad \forall i = 1 \cdots N-1 \tag{3}$$

where p_{ij} is the transition matrix defined in (2). This is a linear equation involving $(t_1, ..., t_N)$ so a solution can be obtained through matrix inversion. For large N, the following simple approximation [13] is often preferred:

$$\mathbb{E}[\tau | A_0 = \lfloor nx \rfloor] \underset{N \to \infty}{\sim} -2N\left(x \ln x + (1-x)\ln(1-x)\right) \tag{4}$$

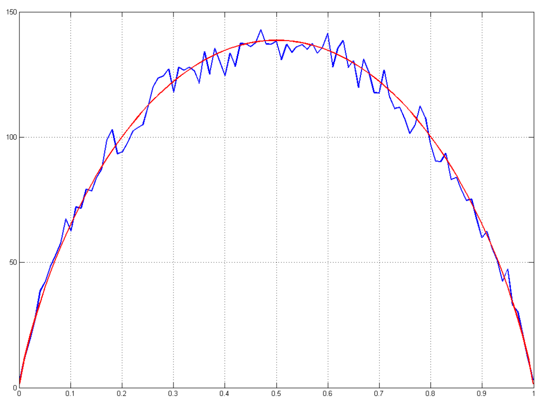

Figure 1. Mean Time of Absorption versus initial proportion x. Red Line : theoretical approximation, Blue Line : empirical mean time

Table I
SIR ALGORITHM WITH TRANSITION PRIOR PROPOSAL

Initialization	Draw N i.i.d samples $x_0^i \sim p(dx_0)$
	Set $w_0^i = \frac{1}{N}$
for $k \geq 1$	Draw N i.i.d $x_k^i \sim p(dx_k \vert x_{k-1}^i)$
	Update the weights $\omega_k^i = \omega_{k-1}^i g(y_k \vert x_k^i)$
	Normalize $\omega_k^i = \frac{\omega_k^i}{\sum_{j=1}^{N} \omega_k^j}$
	Compute $N_{eff} = \frac{1}{\sum_{i=1}^{N} {\omega_k^i}^2}$
	Resampling
	if Neff is below a given threshold
	Resample the set of particles
	end if

x being the initial proportion of samples belonging to the first mode. Note that the mean time of absorption is linear with respect to N which is, in the filtering context, the number of particles.

Figure 1 shows the adequation of the theoretical mean time of loss (4) and the empirical one computed with 500 runs. The theoretical mean time of mode loss is around 138 iterations for $N = 100$ and $x = 1/2$. Note that this estimate has a high empirical standard deviation (100) which means that the probability of losing a mode can be high after only a few iterations. This experiment illustrates that resampling automatically leads to a mode loss in a finite time.
For comparison's sake we ran a bootstrap filter [4] with 100 particles for the state/space model defined in (1). The true state was simulated according to the equations state in (1). x_0 was drawn from a mixture of symmetric uniform distribution $\frac{1}{2}U(-6.01, -5.99) + \frac{1}{2}U(5.99, 6.01)$. The process and measurement noise variance where $\sigma_w^2 = 0.01$ and $\sigma_v^2 = 0.01$.
The mean time of mode loss was found to be around 11 iterations. This lower number can be accounted for by the fact that the resampling is done according to the normalized weights which evolve through time. In the next section, we describe a general framework for limiting mode loss through the use of a mixture representation and derive a new algorithm for regularized particle filters.

III. MIXTURE FILTERING

A. General model

Let X_k be the state of an object partially observed through a series of measurements Y_k. We assume the following state-space model :

$$\begin{cases} X_k = & f_k(X_k, w_k) \\ Y_k = & h_k(X_k) + v_k \end{cases} \tag{5}$$

where the process noise w_k and the measurement noise v_k are mutually independent, i.i.d sequences. We also assume that $v_k \sim p_v(x)dx$, where p_v is a density w.r.t to the Lebesgue measure.

Our aim is to recursively approximate the filtering distribution $p_k = p(X_k|Y_{0:k})$. One may iteratively update the filter through the following steps :

- *Prediction* Assume $p_{k-1} = p(X_{k-1}|Y_{0:k-1})$ is available. Then the predictive distribution can be computed through the following equation :

$$p(X_k|Y_{0:k-1}) = \int p(X_k|x_{k-1})p(x_{k-1}|Y_{0:k-1})\,\mathrm{d}x_{k-1} \tag{6}$$

- *Measurement Update* The filtering distribution is obtained through Bayes rule.

$$p(X_k|Y_{0:k}) = \frac{g(Y_k|x_k)p(x_k|Y_{0:k-1})}{\int g(Y_k|x_k)p(x_k|Y_{0:k-1})\,\mathrm{d}x_k} \tag{7}$$

g denotes the likelihood of the observation given the actual state and can obtained easily since we assume additive measurement noise. We also assume some prior distribution $p(X_0)$ on the initial state. The classical SIR algorithm approximates p_k by the empirical weighed sum

$$\hat{p}_k^N = \sum_{i=1}^{N} \omega_k^i \delta_{x_k^i} \tag{8}$$

where $\delta_{x_k^i}$ denotes the dirac measure at x_k^i.
The filtering problem is then reduced to a weight and particle update (see table I).

It is often necessary to resample the set of particles when weight degeneracy occurs. Indeed, after a few time steps, the empirical distribution $\hat{p}_k^N$ tends to have few particles with significant weight, meaning that $\hat{p}_k^N$ will be a poor approximation of the filtering distribution. The main resampling procedures usually consists in duplicating the particles with high weights and discarding those with low weights any time the variance of the normalized importance weights is too high. To monitor this variance, an empirical criteria called effective sample size (N_{eff}) is often used.

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} {\omega_k^i}^2} \tag{9}$$

Resampling occurs anytime N_{eff} falls below a given threshold N_{thresh}.

As explained in section 2, we may lose the multimodality of the underlying distribution if we resample often. To mitigate this issue the use of a mixture filter representation has often been advocated in visual tracking [2] and robot localization applications [3], the idea being that each mixture component should ideally track a mode. Such a representation can be written as follows :

$$p(X_k|Y_{0:k}) = \sum_{l=1}^{L} \alpha_{l,k} p_l(X_k|Y_{0:k}) \tag{10}$$

where the mixture weights satisfy $\sum_{l=1}^{L} \alpha_{l,k} = 1$ and L is the number of components.

Let us recall briefly how the mixture representation can be updated. Let $p(X_{k-1}|Y_{0:k-1})$ be the current filter representation.

- *Prediction Step* The predictive distribution is obtained as in (6), i.e. :

$$p(X_k|Y_{0:k-1}) = \sum_{l=1}^{L} \alpha_{l,k-1} p_l(X_k|Y_{0:k-1}) \tag{11}$$

  where

$$p_l(X_k|Y_{0:k-1}) = \int p(X_k|x_{k-1}) p_l(x_{k-1}|Y_{0:k-1})\, \mathrm{d}x_{k-1}$$

- *Measurement update* One may show using (7) that :

$$p(X_k|Y_{0:k}) = \sum_{l=1}^{L} \alpha_{l,k} p_l(X_k|Y_{0:k}) \tag{12}$$

  where

$$p_l(X_k|Y_{0:k}) = \frac{g(Y_k|X_k) p_l(X_k|Y_{k-1})}{\int g(Y_k|x_k) p_l(x_k|Y_{0:k-1})\, \mathrm{d}x_k}$$

  and

$$\alpha_{l,k} = \frac{\alpha_{l,k-1} p_l(Y_k|Y_{0:k-1})}{\sum_{j=1}^{L} \alpha_{j,k-1} p_j(Y_k|Y_{0:k-1})}$$

The second equation shows that the mixture weights can be updated recursively provided that an estimate of the predictive likelihood is available for each component.

B. Mixture Particle Filtering

The previous section explained how to update recursively the mixture representation in the Bayesian framework. We shall now address how particle filtering can yield a practical implementation of mixture filters. In their most general setting, particle filters can approximate the filtering distribution without constraining the dynamic and observation models to be linear and/or gaussian. Furthermore, they are particularly suited when the filtering distribution is multimodal.

Similarly to III-A, let L be the number of mixture components and $\alpha_{l,k}$ the mixture weights at time k. Each component p_l is tracked by a particle filter with a set of N_l particles i.e.:

$$p_l \approx \sum_{i=1}^{N_l} \omega_{l,k}^i \delta_{x_{l,k}^i} \tag{13}$$

From (11), it is clear that the prediction step will be done independently for each mixture component using the dynamic model. The same goes for the intra-component weight update which can be done as outlined in I. More generally, if $q(dx_k|x_{k-1}^i, y_k)$ is the proposal distribution used for the particle set propagation, the update rule is :

$$\omega_{l,k}^{*,i} = \omega_{l,k-1}^i \frac{g(y_k|x_{l,k}^i) p(x_{l,k}^i|x_{l,k-1}^i)}{q(x_{l,k}^i|x_{l,k-1}^i, y_k)} \tag{14}$$

$$\omega_{l,k}^i = \frac{\omega_{l,k}^{*,i}}{\sum_{j=1}^{N_l} \omega_{l,k}^{*,i}} \tag{15}$$

Then the particle approximation of our mixture filter is as follows :

$$\hat{p}^N = \sum_{l=1}^{L} \alpha_{l,k} \sum_{i=1}^{N_l} \omega_{l,k}^i \delta_{x_{l,k}^i} \tag{16}$$

where $\sum_{l=1}^{L} N_l = N$, N being the total number of particles.

Note that in this setting, resampling is done locally for each mixture component. This way we can expect to maintain multimodality over time. The mixture weights $\alpha_{l,k}$ reflect the likelihood of each mode and can be evaluated recursively by working out the predictive likelihood of each component.

$$L_l(y) = p_l(y_k|y_{0:k-1}) = \int p_l(y_k, x_k|y_{0:k-1}) \mathrm{d}x_k =$$

$$\int\int g_l(y_k|x_k) p(x_k|x_{k-1}) p_l(x_{k-1}|y_{k-1}) \mathrm{d}x_k \mathrm{d}x_{k-1}$$

Replacing $p_l(x_{k-1}|y_{k-1})$ by its particle approximation yields

$$L_l(y) \approx \iint g_l(y_k|x_k) p(x_k|x_{k-1}) \sum_{i=1}^{N_l} \omega_{l,k-1}^i \delta_{x_{l,k-1}^i} \mathrm{d}x_{k-1} \mathrm{d}x_k$$

$$\approx \int g_l(y_k|x_k) \sum_{i=1}^{N_l} \omega_{l,k-1}^i p(dx_k|x_{l,k-1}^i)$$

For bootstrap filtering, the latter integral can be evaluated since $\{x^i_{l,k},\ i = 1 \cdots N_l\}$ is an i.i.d sequence drawn from the mixture $\sum_{i=1}^{N_l} \omega^i_{l,k-1} p(dx_k|x^i_{l,k-1})$. Thus,

$$p_l(y_k|y_{0:k-1}) \approx \sum_{i=1}^{N} g_l(y_k|x^i_{l,k}) = w^+_{l,k} \tag{17}$$

and

$$\alpha_{l,k} \approx \frac{\alpha_{l,k-1} w^+_{l,k}}{\sum_{j=1}^{L} \alpha_{j,k-1} w^+_{j,k}} \tag{18}$$

Note that in this section, $\{\omega^i_{l,k}\}_i$ denotes the set of weights before resampling.

Recall that throughout this paper, we have implicitly supposed that the number of mixture components is known beforehand and remains fixed over time. Ideally, we wish for each mixture component to represent a mode of the target filtering distribution. Although, in some applications this makes sense (for instance when tracking a fixed number of objects in a video sequence), in terrain navigation, the number of modes evolves over time so an additional step is needed in order to maintain the mixture representation.

At time $k-1$, the filter outputs $\hat{p}_{k-1} = \sum_{l=1}^{L_{k-1}} \alpha_{l,k-1} \hat{p}_l(X_{k-1}|Y_{0:k-1})$. At time k, after the correction step $\hat{p}_k = \sum_{l=1}^{L_{k-1}} \alpha_{l,k} \sum_{i \in I_l} \omega^i_k \delta_{x^i_k}$ may be a mixture where each component overlaps or contains several modes. If this is the case, we need to compute another mixture representation with as many unimodal components as possible.

$$\sum_{l=1}^{L_k} \beta_{l,k} \sum_{j \in J_l} \nu^j_k \delta_{x^j_k} \tag{19}$$

where L_k is the number of new components. $\hat{p}_k$ can be rewritten as

$$\hat{p}_k = \sum_{i=1}^{N} \alpha_{c_1(i),k} \omega^i_k \delta_{x^i_k}$$

we also consider the following expression obtained using the new representation,

$$\hat{p}_k = \sum_{i=1}^{N} \beta_{c_2(i),k} \nu^i_k \delta_{x^i_k}$$

where $c_1(i)$ (resp. $c_2(i)$) is the component number for particle i in the old mixture representation (resp. new mixture representation). In this case (see [2]), we have

$$\beta_{l,k} = \sum_{i \in J_l} \alpha_{c_1(i),k} \omega^i_k \tag{20}$$

J_l being the set of sample indices pertaining to the l^{th} mixture component in the new representation defined in (19).
The new particle weights satisfy :

$$\nu^i_k = \frac{\alpha_{c_1(i),k} \omega^i_k}{\beta_{c_2(i),k}} \tag{21}$$

Note that it's unnecessary to maintain a mode if the corresponding weight is too low i.e. less than a given threshold. To maintain a constant number of particles, an additional step consisting in sampling from the mixture composed by the remaining components is required.
In our terrain navigation application, we used mean shift clustering in order to maintain and update the mixture representation. Introduced in [9], it is a non parametric technique that does not require the number L of clusters to be known in advance : this is useful since L changes with time. Due to similarities in terrain profiles, there is ambiguity on the X^x and X^y coordinates of the mobile's position : in short, a measure of the ground clearance yields multiple locations in the horizontal plane, causing multimodality of the posterior distribution. Hence, it's sufficient to perform $2D$ spatial clustering to determine the components of our mixture representation.
The mean shift algorithm is based on the determination of the modes of the underlying spatial density through gradient ascent. Given N particles x^i, the following kernel density estimate is used for mode finding :

$$\hat{f}(x) = \frac{1}{NB^d} \sum_{i=1}^{N} K_B(x - x^i) \tag{22}$$

where K_B is a gaussian or Epanechnikov kernel and B is a bandwidth parameter.
A cluster is defined as a set of points which converge to the same local maximum. The bandwidth parameter can be typically chosen so as to have a reasonable degree of separation between the clusters. In terrain aided navigation, we found that the use of a fixed bandwidth parameter over time did not degrade the clustering performance.

C. Locally regularized particle filtering

In the previous subsection we have presented the mixture particle filtering in a *Sequential Importance Resampling (SIR)* framework. As a general rule, repeated resampling introduces a loss of diversity in the set of particles. In some cases, the consequences can be extremely severe. For instance, when the process noise is low with respect to the initial uncertainty, the particles tend to accumulate in one zone of the state space. The same can also be observed in the case of a highly peaked likelihood function : this typically happens when the observation noise is additive and very low. This can lead to a high rate of divergence of the classical particle filter if nothing is done to recreate some diversity. The *Regularized Particle Filter (RPF)* [6] alleviates the loss of diversity by replacing the discrete approximation $\hat{p}^N_k$ by a continuous one via a regularization scheme. One of the key steps in the *RPF* algorithm is the whitening procedure that occurs in the resampling step : by adding noise in a controlled fashion, the particles regain diversity. However when the filtering distribution is multimodal, with modes that are dispersed, it was observed that whitening often leads to unstable filter behavior. This can be explained as follows : whitening transforms the sample uncertainty from an ellipsoid to a

sphere. Then after whitening, noise is added in this sphere. In the case where we have 2 or more modes, whitening deforms the true shape of the underlying density.
In the following, we present a new procedure aimed at robustifying the *RPF* through local regularization performed via the use of a mixture representation.

In density estimation, a regularization kernel is a density function $K : \mathbb{R}^d \to \mathbb{R}_+$ satisfying the following conditions :

$$\int K(x)\,\mathrm{d}x = 1 \quad \int xK(x)\,\mathrm{d}x = 0 \quad \int \|x\|^2 K(x)\,\mathrm{d}x < \infty$$

for $h \geq 0$ (bandwidth), the scaled kernel is

$$K_h(x) = \frac{det(S^{-1/2})}{h^d} K(\frac{S^{-\frac{1}{2}}x}{h}) \tag{23}$$

where S is the sample covariance.
Let $\hat{\mu}^N = \sum_{i=1}^N \omega^i \delta_{x^i}$ be a discrete probability measure approximating a certain continous distribution μ with density $f(x)dx$. Then the regularized measure of $\hat{\mu}^N$ has a density

$$(K_h * \hat{\mu}^N)(x) = \sum_{i=1}^N \omega^i K_h(x - x^i) \tag{24}$$

where $*$ is the convolution operator.

The choice of the parameters h and K is done by minimizing the *MISE* criterion *(Mean Integrated Square Error)* between the regularized distribution and the true distribution, i.e. $\mathbb{E}\left\|K_h * \hat{\mu}^N - \mu\right\|_2^2$. The optimal Kernel is the Epanechnikov kernel (see [10]). In the case where μ represents a multivariate Gaussian distribution with unit covariance, the optimal bandwidth for the Epanechnikov kernel is

$$h_{opt} = D(K)N^{-\frac{1}{d+4}} \tag{25}$$

where $D(K) = (8{c_d}^{-1}(d+4)(2\sqrt{\pi})^d)^{(1/d+4)}$ and c_d is the volume of the unit sphere.
In the standard *RPF*, particles are resampled from the continous distribution $(K_h * \hat{p}_k^N)(dx)$. We briefly recall the regularized particle filter algorithm hereafter :

Resampling step in the Regularized Particle Filter

---

at time k, compute N_{eff} according to (9)
if $N_{eff} \leq N_{thresh}$

- compute the empirical covariance matrix S of $\{x_k^i\}_i$
- compute A_k such that $A_k A_k^T = S$
- resample $\{x_k^i\}_i$ according to their weights to obtain a new set of particles $\{\tilde{x}_k^i\}_i$
- draw N iid samples η^i from the Epanechnikov kernel K, $i = 1 \cdots N$
- set $x_k^i = \tilde{x}_k^i + h_{opt} A_k \eta^i$ where h_{opt} is given in (25).

end if

In its basic form, the *RPF* uses a fixed bandwidth parameter even when multivariate distribution are involved. By adding a whitening procedure, we can avoid using d bandwidth parameters for each dimension of our state model. One only needs to work with the whitened samples $\{S^{-1/2}x_k^i\}_i$. However this can yield very bad results when $p(X_k|Y_{0:k})$ is multimodal, especially when the modes are well separated.

To circumvent this, we propose the use of the mixture representation. Assume the filtering distribution admits the following representation :

$$p(X_k|Y_{0:k}) = \sum_{l=1}^L \alpha_{l,k} p_l(X_k|Y_{0:k}) \tag{26}$$

Each component p_l is estimated via the *RPF*. In the same way resampling was done independently for each mixture components in the *SIR* framework, we shall regularize each p_l individually during the resampling step. The idea is that if the above representation is well chosen, each p_l should be unimodal or have undistinguishable modes.
Hence we expect the whitening procedure to maintain the shape of the density of each component. Moreover, in the computation of the optimal bandwidth parameter h_{opt}, the filtering distribution is assumed to be gaussian. This, of course, isn't necessarily the case. By using the gaussian assumption for each mixture component, we expect a more accurate resampling procedure.

Let $\hat{p}_l = \sum_{i=i}^{N_l} \omega_{l,k}^i \delta_{x_{l,k}^i}$ be the empirical density of the l^{th} mixture component at time k, prior to the regularization step. In the local regularization procedure, we first compute the empirical covariance matrix of $\{x_{l,k}^i\}_{i=1\cdots N_l}$, denoted by $S_{l,k}$. We then obtain $A_{l,k}$ such that $A_{l,k}A_{l,k}^T = S_{l,k}$. The choice of the bandwidth now also depends on our mixture components p_l.

$$h_{opt}^l = D(K)N_l^{\frac{-1}{d+4}} \tag{27}$$

where N_l denotes the number of particles for the l^{th} component.
The new set of resampled particles can be worked out as follows :

$$x_{l,k}^i = \tilde{x}_{l,k}^i + h_{opt}^l A_{l,k} \eta^i \tag{28}$$

where $\{\eta^i\}_i$ is an iid sample of size N_l drawn from the Epanechnikov kernel and $\{\tilde{x}_{k,l}^i\}_i$ denote the particles resampled multinomially according to their weights. The corresponding algorithm is summarized below :

- $k = 0$
  - sample $x_0^i \sim p_0(dx_0)$
  - compute the normalized weights w_0^i
  - determine a mixture representation $\{\hat{p}_{0,l}\}$ with component weights $\{\alpha_{0,l}\}$, $l = 1 \cdots L_0$ using the mean shift clustering
- $k \geq 1$
  for each of mixture component $\hat{p}_{k-1,l}$, $l \in 1 \cdots L_{k-1}$
  - sample $x_{l,k}^i \sim p(dx_k|x_{l,k-1}^i)$ $i = 1 \cdots N_l$

- compute the intra component normalized weights $w^i_{l,k}$ according to (15)
- compute mixture weight $\alpha_{l,k}$ according to (18)

end for
update the mixture representation according to (20) and (21), obtain L_k components
<u>*Local regularization step*</u>
for each of mixture component $\hat{p}_{k,l}$

- compute N^l_{eff}
- if $N^l_{eff} \leq N^l_{thresh}$, resample according to (28), where the resampling threshold is defined as a fraction of the total number of particles in the component $N^l_{thresh} = \gamma N_l$.

end for

IV. Application of mixture particle filters to terrain navigation

We consider an aircraft equipped with an INS (inertial navigation system). Such systems provide estimates of the mobile's position, speed and its attitude thanks to internal sensors such as accelerometers and gyrometers : accelerometers measure the specific force whilst gyrometers yield the inertial rotational angular velocity. The integration of the navigation equations leads to an estimate of the aircraft's position.
It is well known that the INS position, speed and attitude estimates drift over time and a regular update is necessary in order to maintain reasonable knowledge of the mobile's position.
Several particle filters ([7], [12]) relying on numerical terrain models have been tested since the development of sequential Monte Carlo method. After recalling the dynamic and measurement model, we show how terrain ambiguity can lead to severe filter divergence and how we can mitigate this issue when using locally regularized particle filters.

A. Dynamic and measurement model

In this section we will consider the problem of estimating the absolute position and speed $X_k = \left(X^x_k, X^y_k, X^z_k, \dot{X}^x_k, \dot{X}^y_k, \dot{X}^z_k\right)$ of an aircraft. For the sake of simplicity, inertial measurements are not taken into account. The dynamical model considered is the following first-order equation :

$$X_k = F X_{k-1} + W_k \tag{29}$$

where

$$F = \begin{pmatrix} 1 & 0 & 0 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{30}$$

A radioaltimeter onboard measures the relative height along the path and the aircraft. Thanks to a digital terrain database, it is possible to compare the measured profile and all those contained in the database. In this setting, the digital terrain elevation is a function of coordinates X^x and X^y.
Thus the measure provided by the radioaltimeter is modelled as follows:

$$Y_k = X^z_k - h_{TE}(X^x_k, X^y_k) + V_k \tag{31}$$

where h_{TE} is the the terrain elevation data and V_k is an additive gaussian noise accounting for the radio-altimeter measurement error. We assume the error between the real terrain profile and the terrain elevation data to be negligible with respect to the measurement noise. Equations (29) and (31) provide a bayesian setting in which particle filtering methods are suited.

One of the main difficulties of terrain aided navigation via radioaltimeter measurements is the ambiguity that may arise due to similar terrain profiles in the immediate surroundings of the aircraft. This can be further aggravated when the measurement noise has a high variance. This leads to a multimodal filtering distribution and the issues of mode loss due to resampling arise (see section II). The other important drawback in a highly ambiguous scenario is the increase of divergences of the *RPF* when frequent resampling occurs.

To maintain the modes throughout the path of the aircraft until a portion of unambigous terrain data appears, we propose the use a mixture of regularized particle filters. Furthermore robustness issues specific to the regularization procedure shall be analyzed.

V. Simulation results

This section presents the performance of the mixture regularized particle filters in the context of important terrain ambiguity. We are interested in maintaining multimodality until the terrain is unambiguous as well as quantifying a possible gain in robustness induced by the local regularization procedure detailed in section III-C.

We consider an aircraft moving along a straight path at a constant speed of $v_0 = 156\ m.s^{-1}$ during approximately four and half minutes.
The mean of the initial particle cloud is drawn from a gaussian centered around the true position with covariance matrix $P_0 = diag(1000m, 1000m, 100m, 5m.s^{-1}, 5m.s^{-1}, 1m.s^{-1})^2$. The time step Δ is $0.7s$. The radio-altimeter measurement noise is modeled as a zero mean gaussian random variable with standard deviation $\sigma_v = 15\ m$.

In the numerical studies, we worked with $N = 1000$, 2000 and 3000 particles. Since the resampling criterion is crucial we also included it in our analysis through the use of the effective sample size defined in (9). Resampling scenarios went from sparse resampling to systematic resampling (Bootstrap filter).
First, the overall performance was assessed by evaluating the rate at which each filter converges. We consider a filter run to be a case of convergence if the final position was within the 99% confidence ellipse centered around the true position.

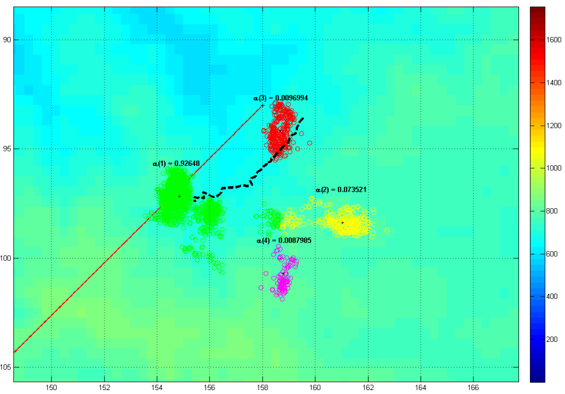

Figure 2. The regularized particle filter with a mean shift procedure : each coloured cloud represents a mode of the filtering distribution. Dashed line : horizontal position estimate, Solid line : true position

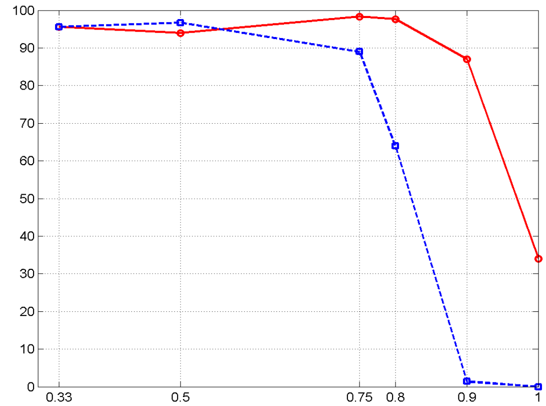

Figure 5. Convergence rate of the Mixture RPF (solid line) and standard RPF (dashed line) plotted against γ such that $N_{thresh} = \gamma N$ (3000 particles)

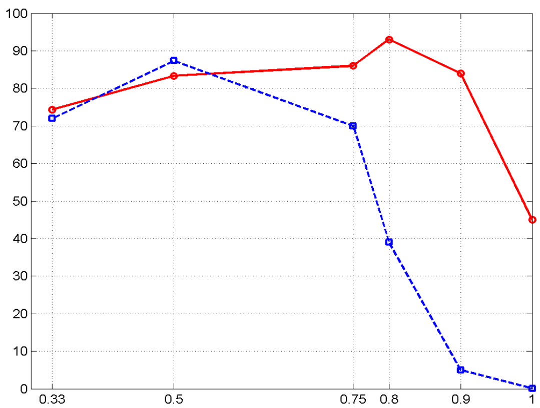

Figure 3. Convergence rate of the Mixture RPF (solid line) and standard RPF (dashed line) plotted against γ such that $N_{thresh} = \gamma N$ (1000 particles)

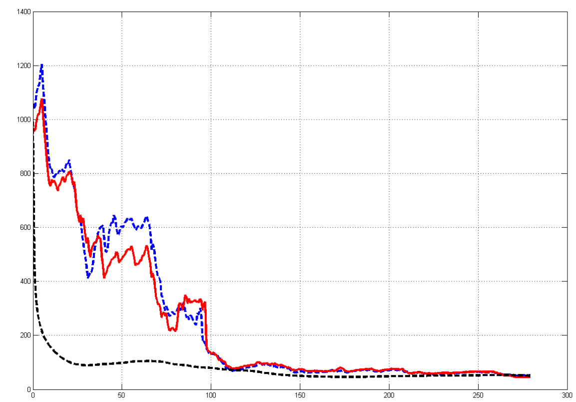

Figure 6. RMSE evolution for the x coordinate of the Mixture RPF (solid line) and standard RPF (dashed line) (3000 particles)

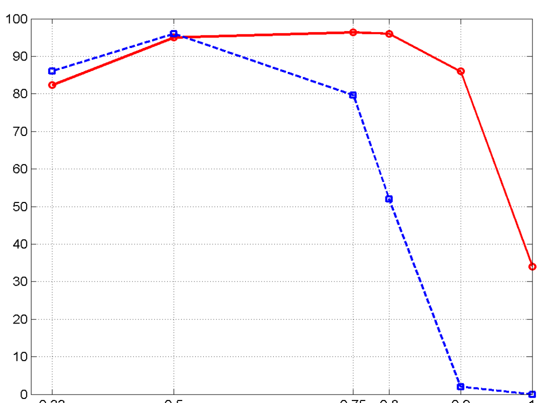

Figure 4. Convergence rate of the Mixture RPF (solid line) and standard RPF (dashed line) plotted against γ such that $N_{thresh} = \gamma N$ (2000 particles)

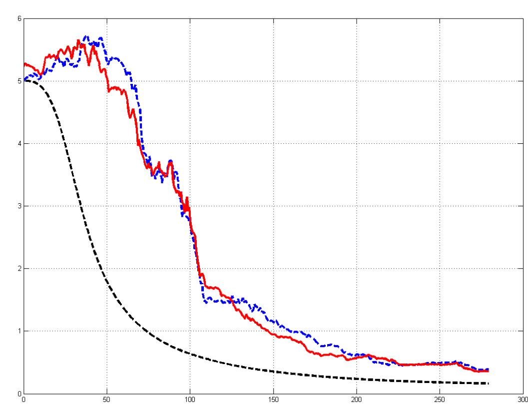

Figure 7. RMSE evolution for the x speed coordinate of the Mixture RPF (solid line) and standard RPF (dashed line) (3000 particles)

This ellipse is given by the Posterior Cramer-Rao Bound (PCRB) [11]. 300 Monte Carlo runs were used in order to derive these rates.

Figures 3, 4 and 5 show the convergence rate as a function of a parameter γ such that the threshold $N_{thresh} = \gamma N$ when the resampling criterion is the effective sample size. A small value of γ indicates that we resample sparingly whereas a higher value will increase the resampling rate. In the study case, the average resampling rate was around 6% for the standard RPF when setting $N_{thresh} = N/3$ and 30% for $N_{thresh} = 3N/4$. According to our results, the mixture *RPF* outperforms the standard *RPF* in terms of convergence whenever the effective sample size threshold is set such that $N_{eff} > \frac{N}{2}$. In particular, the algorithm fares significantly better when we resample often. For instance in the extreme case of systematic resampling, the standard regularized particle filter fails to converge when using 2000 or 3000 particles whereas its mixture version keeps track of the true state in 34% of the runs. This is an interesting point since often times one can only set the resampling threshold once and for all, which may turn out to be suboptimal depending on the terrain. Hence, its desirable to have an algorithm which is less sensitive to the choice of this resampling parameter.
Additionally, in the standard RPF, the empirical covariance of the resampled particles tends to be significantly greater than what we would obtain by using local regularization. By locally whitening the data, we avoid this unstable behavior and are able to keep modes in the posterior density.
Finally, the comparison of the Root Mean Squared Error (RMSE) of both algorithms (figures 6 and 7) did not reveal any significant difference, thus underlining that the main aspect lies in the greater stability of the mixture version of the *RPF*. Note that the all the comparisons were carried out by using the same number of particles for both algorithms and that the RMSE was computed by taking into account only the convergent trials. Finally, the overhead cost of the clustering algorithm used to maintain the mixture representation is moderate.

VI. Conclusions

We have proposed a particle filter adapted to maintaining the modes of the filtering distribution in terrain navigation. In this context, the terrain ambiguities lead to multimodality in the posterior density. The novel algorithm is based on a mixture representation of the *RPF* where resampling is performed for each mixture component. A clustering algorithm ensures that each mixture component corresponds to a mode of the filtering distribution. The algorithm is compared with the classical *RPF* and proves to be more robust in that it is less sensitive to resampling and is able to keep track of the true state at a higher rate.

Acknowledgement

This work was partially supported by DGA (Délégation Générale de l'Armement). The authors gratefully acknowledge Gilles Clerc-Renaud (DGA) for his interest and support.

References

[1] Z. Liu, Z. Shi, M. Zhao and W. Xu, "Mobile Robots Global Localization Using Adaptive Dynamic Clustered Particle Filters," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego CA, USA Oct 29 - Nov 2, 2007.

[2] J. Vermaak, A. Doucet and P. Perez, "Maintaining multimodality through mixture tracking," in *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003)*, vol. 2, 2003, pp. 1110–1116.

[3] A. Milstein, J.N. Sanchez and E.T. Williamson, "Robust global localization using clustered particle filtering," in *Eighteenth national conference on Artificial intelligence*, Edmonton, Alberta, Canada, 2002, pp. 581–586.

[4] N.J. Gordon, D.J. Salmond and A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *Proceedings of the IEE F Radar and Signal Processing*, vol. 140, pp. 107–113, 1993.

[5] M.S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," in *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[6] C. Musso, N. Oudjane, and F. Le Gland, "Improving regularised particle filters," in *Sequential Monte Carlo methods in practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. Springer, 2001.

[7] K. Dahia, C. Musso, D.T. Pham and J-P. Guibert, "Application of the Kalman Particle Kernel Filter to the updated inertial navigation system," in *14th European Signal Processing Conference*, Vienna, Austria, September 6-10, 2004.

[8] A. Giremus, A. Doucet, V. Calmettes, and J.-Y. Tourneret," Rao-Blackwellized particle filter for INS/GPS integration," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04).*, 17-21 May 2004.

[9] D. Comaniciu and P. Meer., "Mean shift: a robust approach toward feature space analysis," in *IEEE Trans. Pattern Anal. Machine Intell.* vol. 24, no. 1, pp. 603-619, 2002.

[10] B.W. Silvermann, *Density Estimation*. Chapman and Hall, London, 2007.

[11] P. Tichavsky, C.H. Muravchik and A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," in *Signal Processing, IEEE Transactions on*, vol. 46, pp. 1386 -1396, 1998.

[12] N. Bergman, *Recursive Bayesian Estimation: Navigation and Tracking Applications*. PhD. Thesis, Linköping University, 1999.

[13] W. J. Ewens, *Mathematical population genetics*. Springer-Verlag, 1979.