COHESIVE PARTICLE FILTERING FOR SOUND SOURCE LOCALIZATION

Jonathan Fillion-Deneault, Jean Rouat

Département de génie électrique et de génie informatique
Université de Sherbrooke, Québec, Canada

ABSTRACT

We present a novel resampling technique for particle filtering algorithms based on a model of attractive and repulsive forces. The new approach avoids the common degeneracy problem found in traditional resampling algorithms that prevents precise tracking of a source. The algorithm is applied to an 8 microphone source localization problem and is shown to accurately localize a single speaker.

1. INTRODUCTION

Particle filter theory, also known as Bayesian filtering, is becoming increasingly popular as an alternative method to discrete filtering for detection, tracking and identification in complex systems. The method promises good accuracy and performance in various domains such as mobile robotics, computer vision, economics and network communications where it is often desired to find the true value of a parameter based on its noisy observations.

One crucial step in the implementation of particle filtering algorithms is the choice of the resampling method used between each predict-update iteration of the filter. As is often reported in literature [1] [2], the principal problem with particle resampling is degeneracy or otherwise known as particle impoverishment. Particles of considerable weights are continually resampled over those of lower weights, up to the point where they become cluttered and confined in a small space, thus preventing the filter from detecting anything else from that point onwards.

We propose a new approach to resampling where the particles are regarded as having attractive and repulsive forces between them, proportional to their weight and respective to their mutual proximity. The particles are still updated according to the system model, but the suggested resampling approach ensures a certain cohesion (attractive forces) and coverage (repulsive forces) between the particles. The new method named *cohesive particle filtering* is applied to a classic acoustic source localization problem, extending on the works of [3].

The remainder of this paper is organized as follows. Section 2 presents the basic theory behind particle filtering. Section 3 then exposes the degeneracy problem and its common solutions. We then introduce our *cohesive particle filtering* in Section 4, followed by results of its implementation on a source localization problem in Section 5. Further ideas and avenues are finally discussed in Section 6.

2. THEORY

Particle filtering is an effective solution to the problem of estimating the true value of a variable (a position, an orientation, a speed, etc.) when presented with its noisy observations. The general idea of particle filtering is to enhance iteratively the probabilities that the real value be x given all past observations y where $y = f(x)$. This iterative process is ran at time index k such that we are looking for a measure of $p(x_k|y_{1:k})$. The derivation of the predict-update formula follows from Baye's chain rule [4], and is given by:

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \tag{1}$$

This is the most encountered form of the predict-update equation that describes the iterations of a particle filter. Looking at the right hand side of (1), we note that :

1. $p(y_k|x_k)$ is the likelihood of observing y given the current estimation x; this is called the **update step**.
2. $p(x_k|y_{1:k-1})$ equals $\int p(x_k|x_{k-1}) \cdot p(x_{k-1}|y_{1:k-1})dx_{k-1}$
   (a) $p(x_k|x_{k-1})$ is called the **predict step** and describes the probability that a transition occurs from x_{k-1} to x_k (2 different positions, for example)
   (b) $p(x_{k-1}|y_{1:k-1})$ was found on the last iteration
3. $p(y_k|y_{1:k-1})$ is a **normalizing constant** such that the sums of $p(x_k|y_{1:k})$ for all possible x_k's add up to unity

In a practical implementation of (1), one needs the update step, the predict step and a discrete set of x's onto which the iteration will be evaluated. The update and predict step are usually easily obtained (or approximated). What is problematic is the choice of discrete x's onto which the filter will be applied. In a high dimensional space, the number of x's can grow very large. For example, pinpointing the location of a speaker in a 3 dimensional 10m cubic room to a resolution of 1cm yields 10^9 points to consider, for each analyzing frame. In the particle filtering terminology, those x's are called **particles** and their probability of occurrence derived from (1) is called their **weight** w.

In practice, the filter will be evaluating (1) 99% of the time on particles yielding near 0 probability. The speaker is in a narrow space and it is useless to continually evaluate all the positions where probability of presence is low; this fact has been observed [4] and **resampling** algorithms have been introduced to solve the problem. The idea behind resampling is to eliminate small probability x's and replace them in higher probabilistic regions of space (ie, to duplicate particles with higher weights).

Several resampling techniques exist, the most popular ones being systematic resampling, residual resampling and stratified resampling. In systematic and residual resampling [2], a *deterministic* approach is taken to resample the M particles such that a particle with normalized weight w_i gets duplicated $\lfloor w_i \cdot M \rfloor$ times on the next iteration. On the other hand, stratified resampling [5] generates M random trials with probability w_i of picking particle x_i; the number of times a particle is picked represents the number of its duplications for the next iteration. That being said, the higher the weight

of a particle, the more it will be duplicated, thus having the effect of concentrating the particles in a high probabilistic region of space. That can be detrimental to the filter's performance since particles become confined in a small space and the filter's view degenerates. This problem called *degeneracy* of the particle filter, is also refered by some other authors as *particle impoverishment*.

3. MOTIVATION

As stated above, the main drawback of conventional resampling is degeneracy, as illustrated in Fig. (1) where very few noisy observations y lead to successive duplication of particles around the noise source, degenerating the solution space, thus missing the real source when it appears later (much stronger) in other parts of space.

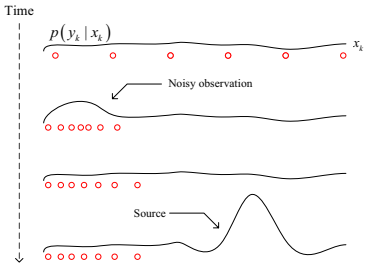

Fig. 1. Illustration of the degeneracy problem in a 1-D case

In fact, it has been established by the Kong-Liu-Wong theorem (Kong et al. 1994) that the variance of the weights can only increase over time; degeneracy is thus inevitable. Some improvements to classic resampling have then been suggested to correct this degeneracy problem. The *condensation* algorithm introduced by [6] tries to avoid degeneracy by injecting particles sampled uniformly from the solution space at each predict-update step; the contribution of the sensing is thus "boosted" compared to that of the predictive model. As an example, the algorithm evaluates $p(y_k|x_k)$ for a set of uniformly (and coarsely) distributed x_k to sense events outside of the current particles' region.

Another similar modification suggests systematically keeping a small fraction of the particles x_{k-1} to time k, independently of their weights, such that lower weight particles are not necessarily eliminated. Yet another technique called *kernel smoothing* (J. Liu and M. West, 2000) consists of adding small perturbations to the particles between each iteration to avoid eventual cluttering of the particles.

Though those fixes have proven effective in reducing degeneracy for some cases, their behavior is often hard to tune and predict; how many new particles should we inject? how closely should we sample $p(y_k|x_k)$ when injecting new particles? how many particles should we systematically keep between each iteration? how much perturbation should we introduce in kernel smoothing? What we suggest here is a different approach which requires less tuning, where the motion of the particles is based on attractive and repulsive forces; ***cohesive particle filtering*** is the term we use to describe our method.

4. METHOD

4.1. Resampling

To solve state degeneracy, we want to prevent the particles from collapsing to a single point when noise or simply no source is present, but at the same time we require them to concentrate around real sources when they are detected. This induces the ideas of *attraction* and *repulsion* between the particles. Attraction will ensure cohesion between the particles in regions of space maximizing $p(y_k|x_k)$, and repulsion will prevent them from collapsing to a single point (thus preventing excessive degeneracy). Mathematically, particle x_i will undergo a change in position at time k of $\overrightarrow{\Delta_i}$ given by (2) and (3).

$$x_{k+1,i} = x_{k,i} + \overrightarrow{\Delta_i} \tag{2}$$

$$\overrightarrow{\Delta_i} = \sum_{j,j\neq i}^{N} \left(f_a(i,j) - f_r(i,j)\right) \overrightarrow{u_{i,j}} \tag{3}$$

In (3), $f_a(i,j)$ and $f_r(i,j)$ are respectively scalar forces of attraction and repulsion between two particles x_i and x_j, and $\overrightarrow{u_{i,j}}$ is a unit vector pointing from x_i to x_j. The functions we propose as $f_a(i,j)$ and $f_r(i,j)$ are:

$$f_a(i,j) = K_a \frac{w_j^2}{\left|\overrightarrow{d_{i,j}}\right|}, \quad f_r(i,j) = K_r \frac{1}{\left|\overrightarrow{d_{i,j}}\right|^2} \tag{4}$$

where $\overrightarrow{d_{i,j}}$ is the geometric vector from particle x_i to x_j. Substituting (4) into (3) leads to:

$$\overrightarrow{\Delta_i} = \sum_{j,j\neq i}^{N} \left(K_a \frac{w_j^2}{\left|\overrightarrow{d_{i,j}}\right|} - K_r \frac{1}{\left|\overrightarrow{d_{i,j}}\right|^2} \right) \overrightarrow{u_{i,j}} \tag{5}$$

where the constants K_a and K_r can be adjusted to control the spread of the particles in the solution space. Specifically, increasing K_a and reducing K_r will result in particles staying closer to each other, and vice-versa; one can thus tradeoff bias for variance of the final position estimate. Two constants are used (instead of one) to control the *speed* of attraction and repulsion. As an example, increasing K_a will lead to faster reacting particles than if K_r were reduced (because of the w_j^2 term). The behavior of particles can thus be customized by changing those 2 constants. This resampling method differs somewhat from traditional sampling importance resampling techniques where particles are destroyed and replicated as a function of their current weight. Here, particles are rather moved as a function of their neighbor particles' weights. The final estimate of the true state is then taken as a weighted mean of the particles state. One notices in this case that the complexity of the algorithm is proportional to $O\left(N^2\right)$ where N is the total number of particles.

4.2. Prediction & update

The cohesive particle filter is implemented and applied to a microphone array single source localization problem based on the works of [3]. The far-field assumption is adopted and a steered beamformer is used to calculate $w_k = p(y_k|x_k)$ where the particles x_k represent **unit norm vectors** (ie, points on a sphere). For debugging and visualization purposes only, the sphere is tessellated and the energy w_k is also calculated on all joints of the tesselated sphere and used to color its faces (the higher the energy, the brighter the face); this is *not* a necessity of the algorithm, as it would beat the point of having particles in the first place. As explained in [3], the energy w of a point $\overrightarrow{v}$ (unit-norm vector) is calculated as:

$$w = \sum_{k=1}^{M} \sum_{l=k+1}^{M} R_{kl}(\tau_{kl,i}) \tag{6}$$

where M is the number of microphones, $R_{kl}(\cdot)$ is the cross-correlation between data from microphones k and l and τ_{kl} is pre-calculated as:

$$\tau_{kl} = \frac{f_s}{c}\left(\overrightarrow{v} \cdot \overrightarrow{d_{k,l}}\right) \tag{7}$$

where $\overrightarrow{d_{k,l}}$ is a geometric vector from microphone k to microphone l. The calculation of w at time k constitutes the **update step** as $w_k \triangleq p(y_k|x_k)$. The **prediction step** is trivial; since we assume that the speaker doesn't move considerably between sound frames, we will not modify the position of the particles (the resampling step is able to track the speaker by itself).

4.3. Estimation

Finally, after each step, a weighted mean of the particles is taken as an estimate of the true state (direction) of the source, as defined by:

$$\overrightarrow{estimate_k} = \frac{\sum_i w_{k,i}\overrightarrow{v_{k,i}}}{\sum_i w_{k,i}} \tag{8}$$

4.4. Summary

The algorithm thus operates in the following steps:

1. Acquire a new sound frame
2. For all particles:
   (a) Calculate the τ_{kl} delays using (7)
   (b) **Update**: Evaluate (6)
   (c) **Estimate**: Evaluate (8)
   (d) **Resample**: Evaluate (3) and (2)
3. Goto 1)

The evolution of the particles on the sphere of possible directions can be visualized with simulation tools to adjust the constants K_a and K_r as well as the number of particles thus controlling the spread of the *particle cloud*. Such visualizations, similar to Fig. 2, can be concatenated into videos that depicts the evolution of the algorithm in time. Samples videos illustrating the behavior of the algorithm can be downloaded from the author's web site [7].

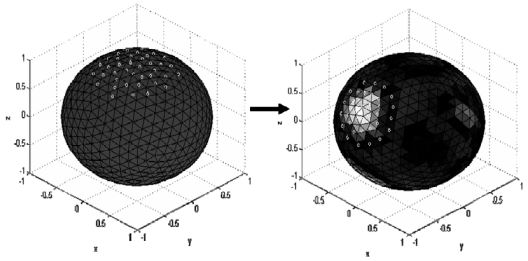

Fig. 2. Evolution of the particles in time

5. RESULTS

Using multi-microphone recordings of a stationary speaker, we will evaluate and compare the performances of 4 different systems, namely: our newly introduced cohesive particle filter (CPF), stratified resampling with kernel smoothing (KS), stratified resampling with condensation (CND) and without any filtering at all (NF). The recordings were taken with an array of 8 microphones arranged in a cube-like configuration; short French sentences are uttered by a female speaker and last about 4 seconds. In the case of the KS system, an additive gaussian noise of variance $\sigma^2 = 0.05$ was found to give optimal results, while injecting the 5 sphere vertices maximizing $p(x_k|y_k)$ for the CND system showed best localization performance. To ensure reproducibility of the following results, the experimental conditions are described in table 1 and the Matlab code as well as the speech samples used to implement and test the filter are readily available from the author's web page [7].

Sample information	
Sampling frequency	44.1kHz
Segment length	4s
Analysis window	Hamming
Frame length	50ms
Frame overlap	50%
Speed of sound (c)	340.29m/s
Filter parameters	
Number of particles	40
K_a	0.004
K_r	0.002

Table 1. Experimental conditions

As mentioned previously, a weighted mean of the particles is generated on each frame as the final estimate of the speaker's location. Since this represents a position in 3 dimensional space (a unit-norm vector), its azimuth and elevation angles (in degrees) are extracted for plotting purposes. Moreover, the sum of the particles' non-normalized weights is used as a *confidence* measure to control the color intensity of the plotted points; it is understood that in speech pauses, no useful localization information can be obtained and that the confidence can only be low. It is the filter's job to keep the output constant so that tracking can be done properly when the speaker is heard again.

The results in Fig. 3 show the ability of the 4 systems (CPF, KS, CND and NF) to locate a single speaker located at azimuth = -96^o and elevation = 38^o. Since the speaker location is initially unknown to the filter, the particles are placed randomly on the top of the sphere, over a radius of roughly 30 degrees. The systems are then started and are all seen to successfully locate the speaker when speech activity is present (after around 0.5 second here). The CPF is the stablest of all since particle attraction prevents the cloud of particles from degenerating and diverging in speech pauses. The KS correctly identifies the speaker's location but is seen to be less stable than the CPF. For the CND and NF systems, all joints of the tesselated sphere need to be evaluated by (1); the 5 most energetic directions get injected for resampling in the CND system, while *the* most energetic is taken as the final estimate in the NF system (since no filtering is applied). As can be seen, both CND and NF systems easily loose track of the speaker, especially during speech pauses.

To quantify the improvements of the systems, a measure of the MSE (Mean Squared Error) taken over 6 different recordings was computed and is depicted in Fig. 4. The error measured is simply the angular difference between the estimated direction and the actual one, taken over the last second (40 frames of 50ms overlapping by 50%) of each simulation; only the last second is considered to allow

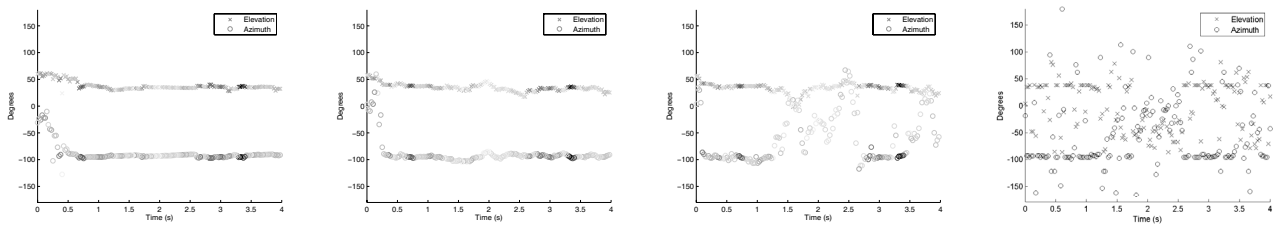

Fig. 3. Speaker localization with a) Cohesive particle filter (CPF) b) Kernel smoothing (KS) c) Condensation (CND) d) No particle filter (NF)

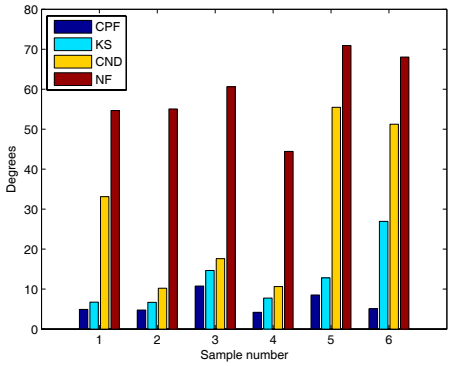

Fig. 4. MSE of the 3 systems for 6 different samples

filters to stabilize around the speaker's location (as seen on Fig. 3, stabilization times are on the order of 0.5 to 1 second). Our CPF method shows the best results and is seen to offer an MSE of less than 10 degrees, constantly lower than the KS system. The MSE is significantly higher when contribution from the sensing is given importance (CND), as small reflections or noise sources easily disturb the filter; the NF system is obviously the most sensitive to noise since it essentially operates on a per-frame basis.

On a complexity standpoint, we note that CND and KS are roughly $O(N \log N)$ because of the select with replacement operation. For a convenient number of particles in the given application $N < 100$, the greater complexity of our method ($O(N^2)$) was not found to be an issue for real-time operation. The performance difference depends on the exact form of f_a and f_r in (3), as well as optimization and several other implementation details.

6. CONCLUSION

To overcome degeneracy problems in particle filter implementations, we have introduced a new kind of resampling algorithm based on attractive and repulsive forces of particles. These forces are function of the distance between the particles and their weight; particles of higher weights *attract* their neighbors so that a source can be tracked while *repulsive* forces prevent the particles from collapsing onto a single point.

The newly introduced filter called a *cohesive particle filter* was implemented and tested on several 8-microphone recordings to locate a stationary speaker, and was shown to offer good accuracy and stable measurements, even during speech pauses. Simple tuning can control the spread of the particles such that it can be adapted to a wide range of filtering problems.

In its current implementation, the filter can only track a single source; it could be extended to several sources by segmenting the particles into different cohesive groups each behaving independently. Furthermore, attractive and repulsive forces other than the ones suggested here could be used to control differently the behavior of the filter's particle cloud. Strategies to reduce the overall complexity of the algorithm could be investigated as well. Finally, the theoretical consistency of our resampling step with the convergence of the weighted set of samples to the true MMSE state will be studied in a near future.

7. ACKNOWLEDGMENT

The author would like to thank Jean-Marc Valin for providing the datasets used in the experiments and simulations and to the reviewers for their constructive comments. Acknowledgement should also be given to the *Groupe de Recherche sur la Parole et l'Audio* of Sherbrooke University for providing the labs used for this research.

8. REFERENCES

[1] A. Doucet, S. Godsill, and Christophe Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," Tech. Rep. CUED/F-INFENG/TR. 310, Cambridge University Department of Engineering, 1998.

[2] Miodrag Bolic, Petar M. Djuric, and Sangjin Hong, "New resampling algorithms for particle filters," in *Proc. ICASSP 2003*, 2003, pp. 589–592.

[3] J.-M. Valin, F. Michaud, B. Hadjou, and J. Rouat, "Localization of simultaneous moving sound sources for mobile robot using a frequency-domain steered beamformer approach," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[4] Simon Maskell, "A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.

[5] Miodrag Bolic, Petar M. Djuric, and Sangjin Hong, "Resampling algorithms for particle filters: A computational complexity perspective," *EURASIP Journal on Applied Signal Processing*, vol. 15, pp. 2267–2277, 2004.

[6] P. Jensfelt, O. Wijk, D. Austin, and M. Andersson, "Experiments on augmenting condensation for mobile robot localization," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, Apr. 2000.

[7] J. Fillion-Deneault, "Cohesive particle filtering samples," http://www-edu.gel.usherbrooke.ca/filj2202/cpf/.